Invited Paper

Towards a Biological Coding Theory Discipline

Elebeoba E. May

Computational Biology Department, Sandia National Laboratories, Albuquerque, NM 87185 USA
e-mail: eemay@sandia.gov

General abstract: *How can information required for the proper functioning of a cell, an organism, or a species be transmitted in an error-introducing environment? Clearly, similar to engineering communication systems, biological systems must incorporate error control in their information transmission processes. If genetic information in the DNA sequence is encoded in a manner similar to error control encoding, the received sequence, the messenger RNA (mRNA) can be analysed using coding theory principles. This work explores potential parallels between engineering communication systems and the central dogma of genetics and presents a coding theory approach to modelling the process of protein translation initiation. The messenger RNA is viewed as a noisy encoded sequence and the ribosome as an error control decoder. Decoding models based on chemical and biological characteristics of the ribosome and the ribosome binding site of the mRNA are developed and results of applying the models to the* Escherichia coli *K-12 are presented.*

Keywords: Coding Theory, Translation Initiation, Information Theory, Biological Information Processing

Introduction

The translation of the messenger RNA (mRNA) sequence into chains of protein-forming amino acids can be compared to the decoding of a received information sequence in an engineering communication system. In a communication system, coding techniques are used to compensate for errors that occur during transmission of information (Sweeney, 1991). Error control is accomplished by introducing redundancy into the original information sequence. Therefore, there are more symbols in the transmitted sequence than in the original sequence (Sweeney, 1991). Redundancy naturally occurs within RNA and DNA sequences (Lewin, 1995). The existence of tandem repeats, and sequences such as the Shine-Dalgarno sequence, the Pribnow box, and the TATA box, leads us to believe that cellular communication systems use some method of coding to recognize valid information regions within a nucleotide sequence and correct for "transmission" errors such as mutations. Genetic mutations occur when replication errors are missed by genetic proofreading mechanisms. Since mutations corrupt the original message, in a communication sense, mutations can be viewed as transmission errors.

The goal of this work is to provide a basic introduction to ideas in coding theory and show how these ideas can be used to analyse genetic processes and sequences. The application of coding theory to genetic sequence analysis can provide new computational methods for modelling the regulatory aspects of translation. The first part of this paper provides an introduction and overview of coding theory, specifically the two main types of codes: block codes and convolutional codes. This is followed by a discussion of how genetic processes can be viewed from a coding theory perspective. References to text that provide a more in depth coverage of coding theory and other work on biological communication models are provided throughout.

Coding Theory: An Introduction

The need for coding theory and its techniques stems from the need for error control mechanisms in a communication system. Error control mechanisms can be categorized as forward error correction and retransmission error control. Forward error correction assumes that errors will occur and provides a mechanism that when applied to the received message, is able to correct the errors. Retransmission error control techniques detect the errors in the received message and requests retransmission of the message (Sweeney, 1991). The system in Figure 1 illustrates how error control coding (ECC) is incorporated into a typical engineering communication system (Sweeney, 1991). Digitised information is encoded by the channel encoder and prepared for transmission (modulation). It is then transmitted via a potentially noisy channel where the transmitted information may be corrupted in a random fashion. At the receiving end, the received message is prepared for decoding (demodulation) and then it is decoded by the channel decoder. The decoding process involves removal of errors introduced during transmission. The decoding mechanism can only cope with errors that do not exceed its error correction capability. The elements we will focus on in this system are the encoder, the channel, and the decoder.

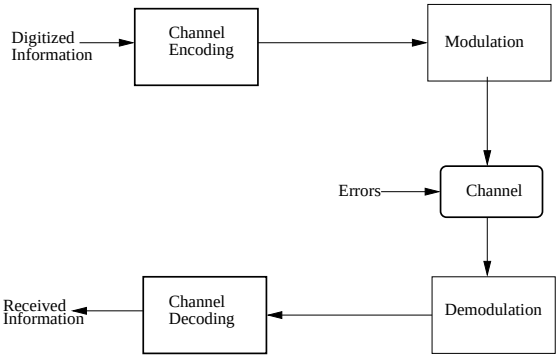

Figure 1: Communication system that incorporates coding.

a) Error Correcting Codes

The mathematics of coding is carried out in a finite field also referred to as a Galois field (Sweeney, 1991; Anderson & Mohan, 1991). A q-ary finite field GF(q) is a Galois field with q elements that consists of a finite set of symbols, a set of two operations, and the inverses of those operations. The operations and their inverses, when applied to the set of symbols, can only yield values within that set. The binary field GF(2) consists of:

- Finite set of symbols: 0,1
- Operations: modulo 2 addition (+) and modulo 2 multiplication (*)
- Corresponding inverse operations

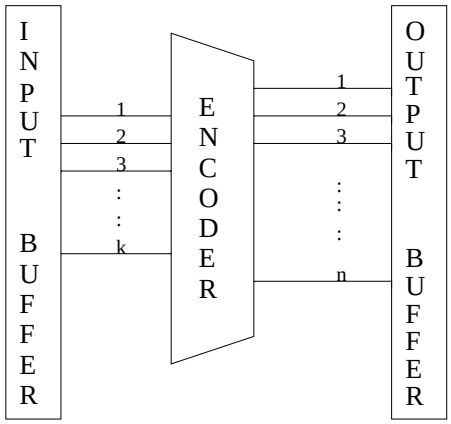

Figure 2: Encoder.

i) The Encoder

The encoder, represented in Figure 2, encodes the digitised information frame by frame. An input frame consists of a fixed number, k, of 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 (Sweeney, 1991). The coding rate,

$$R = k / n \qquad (1)$$

is 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 (Sweeney, 1991).

The encoder combines the k input symbols with $n - k$ symbols usually based on a deterministic algorithm, although random encoding methods, as illustrated by Shannon (Shannon & Weaver, 1949; Anderson & Mohan,

1991), can be used. Encoding results in a mapping of input frames into a set of output frames known as codewords. There must be a codeword for every possible information sequence. For a q-ary alphabet, the encoder will produce q^k codewords. As an example, for a binary code ($q = 2$) with $k = 2$, there are 2^2 or four possible information sequences therefore four codewords. The set of q^k codewords comprise the codebook. Because encoding adds redundant bits, hence n is greater than k, there are a number of n-bit sequences (exactly $q^n - q^k$ such sequences) which are not codewords. This allows error detection and correction. If a transmitted n-bit sequence does not map to a codeword, we assume one or more bits have been corrupted. The decoding task is to find the most likely changes in a transmitted n-bit sequence that will result in a valid codeword.

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 (Sweeney, 1991; Dholakia, 1994). Error control codes can be referred to as (n, k) codes or (n, k, m) codes in the case of convolutional codes where m is the memory length (a more detailed discussion of encoder memory is presented in (c)).

ii) Communication Channel

The communication channel is the medium through which information is transmitted to the receiver. The channel can corrupt the transmitted message through attenuation, distortion, interference, and addition of noise. The way in which transmitted binary symbols (0 or 1) are corrupted depends on various characteristics of the communication channel (Sweeney, 1991). If the channel is a:

- Memoryless Channel: The probability of binary symbol error is statistically independent of the error history of the preceding symbols.
- Symmetric Channel: For binary symbols, 0 and 1, the probability of 0 being received instead of 1, due to transmission errors, is the same as the probability of 1 being received instead of 0.
- Additive White Gaussian Noise (AWGN) Channel: This is a memoryless channel which adds wide-band, normally distributed noise to the amplitude modulated transmitted signal.
- Bursty Channel: There are periods of high symbol error rates separated by periods of low, or zero, symbol error rates.
- Compound Channel: The errors are a mix of bursty errors and random errors.

iii) Decoder

The method of decoding is dependent on the method of encoding. The aim of a coding system is to attempt to detect and correct the most likely errors. The decoder receives a series of frames that, given no errors in the 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. Different mechanisms are then used to decide what the original codeword was and thus correct the error. When the error rate exceeds the correction capacity of the code, two things can occur: 1) The decoder can detect the error but cannot find a unique solution and thus correct the error or, 2) The decoder cannot detect the error because the corruption has mapped one legal codeword into another legal codeword. Errors that exceed the error correcting capabilities of the code may not be handled correctly.

b) Basics of Linear Block Codes

A linear block code is a code defined such that the sum of any two codewords results in another valid codeword in the code book set. A more rigorous discussion on linear block codes can be found in most coding theory texts (Sweeney, 1991; Anderson & Mohan, 1991; Lin & Costello, 1983).

i) Encoding

There are several ways to produce codewords from a k bit information sequence (Lin & Costello, 1983). One

method, systematic encoding, produces codewords which contain the k information bits at the beginning of the codeword. The information bits are then followed by $n - k$ parity bits. All non-systematic linear block codes can be reduced to an equivalent systematic code (Anderson & Mohan, 1991). The value of these $n - k$ bits is determined by the encoding algorithm represented by the generator matrix G. The generator matrix is used to encode the k-bit information vector, u, and form the n-bit transmitted codeword vector, v. The relationship between u, v, and G is:

$$v = uG \tag{2}$$

The generator matrix, G, is k by n, u is 1 by k, and v is 1 by n; this yields the following matrix representation of the above equation:

$$v = \begin{bmatrix} u_1 & u_2 & \cdots & u_k \end{bmatrix} \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{bmatrix} \tag{3}$$

where,

$$v = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \tag{4}$$

The codeword, v, is produced by the modulo q (where $q = 2$ for binary sequences) addition of basis codewords (Sweeney, 1991). The basis codewords are the k linearly independent codewords which form the generator matrix. Linearly independent codewords are the set of k codewords that cannot be produced by linear combinations of two or more codewords in the codebook set.

When the generator matrix is in systematic form, G is of the form:

$$G = \begin{bmatrix} I_k ; P \end{bmatrix} \tag{5}$$

where I_k is the k by k identity matrix and P is a k by $n - k$ matrix (Anderson & Mohan, 1991). Equation 6 and Table 1 show the generator and corresponding data to parity mapping for a simple (3,2) linear block code.

$$G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{6}$$

From Table 1 we note that the codebook set is S_C = (000, 011, 101, 110).

u	$v = uG$
00	000
01	011
10	101
11	110

Table 1: Data to parity mapping for simple (3,2) linear block code.

ii) Decoding

Decoding involves two steps. First the decoder must check whether the sequence corresponds to a codeword. Second, if the decoder is an error correcting decoder, then it must identify the error pattern. There are various decoding methods. One method, minimum distance decoding, is a maximum likelihood approach based on comparative Hamming distance values between a received sequence and codewords in the codebook. The Hamming distance between two sequences, $d(a, b)$ is the number of differences between sequence a and sequence b (Sweeney, 1991). For a received sequence r, the minimum distance, $dmin$ of r is the minimum of $d(r, S_c)$, where S_c is the set of all codewords in the codebook. In minimum distance decoding, we decode r to the codeword for which $d(r, S_c)$ is the least. If the minimum distance computation results in the same distance value for more than one codeword, although an error is detected, it is not correctable because of the degeneracy of the

mapping. Minimizing the distance is the optimum decoding approach for a channel in which symbol errors are independent (memoryless channel) (Sweeney, 1991).

Another decoding technique, syndrome decoding, is based on the relationship between r the received sequence (a potentially noisy version of v) and the $(n - k)$ by n parity-check matrix H. The H matrix is the generator for the dual code space with respect to the code space generated by G (Anderson & Mohan, 1991). The parity-check matrix has the form:

$$H = \left[P^T ; I_{n-k}\right] \tag{7}$$

where P^T is the transpose of the P matrix of G (see Equation 5) and I_{n-k} is the $n - k$ by $n - k$ identity matrix (Anderson & Mohan, 1991). The relationship between H and G is:

$$GH^T = 0 \tag{8}$$

For every valid codeword, v, in the coding space of G:

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

If we represent the n-symbol received vector, r, as $r = v + e$, where e represents the error vector introduced by the channel, we can define the syndrome of r as:

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

The syndrome is the error pattern present in the received information sequence. In the absence of detectable errors $s = 0$. The syndrome pattern can be used to correct and decode the received information sequence. Using the simple (3,2) code in Equation 6, the corresponding H matrix is:

$$H = \begin{bmatrix}1 & 1 & 1\end{bmatrix} \tag{11}$$

Given two received messages r_1 = [011] and r_2 = [010] we can calculate the syndrome values for each, potentially noisy sequence:

$$s_1 = r_1 H^T = \begin{bmatrix}0 & 1 & 1\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}0\end{bmatrix} \tag{12}$$

$$s_2 = r_2 H^T = \begin{bmatrix}0 & 1 & 0\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}1\end{bmatrix} \tag{13}$$

From this simple illustration, we note that the non-zero s_2 syndrome value accurately indicates the presence of an error in r_2 while the zero s_1 value indicates the absence of errors in the received r_1 sequence. In later sections we theorize that this syndrome checking framework can be paralleled to the behaviour of various macromolecules, such as the ribosome, that operate on genetic messages.

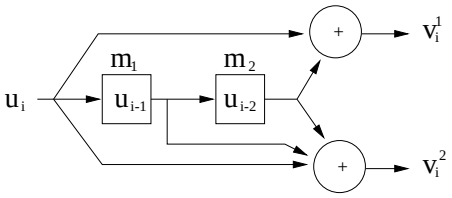

Figure 3: A (2,1,2) convolutional encoder.

c) Basics of Convolutional Codes

Block codes produce encoded blocks from the present information block at time i. In contrast convolutional coding produces encoded blocks based on present and past information bits or blocks. Convolutional coding, like block coding, is carried out over a finite field, using a set of discrete source symbols. For now, we consider the binary field, consisting of [0, 1] and the operations modulo two addition and modulo two multiplication. In convolutional encoding, an n-bit encoded block at time i depends on the k-bit information block at time i and on m previous information blocks (Dholakia, 1994). Hence, a convolutional encoder requires memory. Convolutional codes are referred to as (n, k, m) codes.

i) Encoding Methodology

A convolutional encoder is a mechanism with a *k*-bit input vector u_i, *n*-bit output vector v_i, and *m* memory elements. Figure 3 (Dholakia, 1994) illustrates a (2, 1, 2) convolutional encoder, where the blocks indicate memory. Figure 3 shows a $k = 1$, $n = 2$, or 1/2 rate encoding scheme where a block is equal to one bit. That is, for every input bit encoding produces two parity bits. The general encoding procedure is as follows (Sweeney, 1991; Dholakia, 1994):

- A *k*-bit input block at time *i*, u_i, is modulo two added to the previous *m* input bits to form the *n*-bit output vector v_i.
- The most recent *k* input bit is shifted into the memory register and the rest of the bits in the register are shifted to the right.
- The new input block is then modulo two added to the contents of the memory register to produce a new output vector.
- The process is repeated until all input data has been encoded.

A set of *n* generator vectors completely specify the encoder. The generators are *m* + 1 bits long and indicate which elements are modulo two added to produce each bit in the output vector. For the encoder illustrated in Figure 3, the generator vectors are as follows:

$$g_1 = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \quad (14)$$

$$g_2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \quad (15)$$

The generator vectors can also be represented as generator polynomials:

$$g_1(x) = 1 + x^2 \quad (16)$$

$$g_2(x) = 1 + x + x^2 \quad (17)$$

For x^D, *D* represents the number of delay units. Each generator vector or polynomial is associated with one of the *n* output bits in the output vector *v*. The encoding process depends not only on the present input but also on the previous *m* inputs. This forms an interdependence among the transmitted data bits. Given the following information stream:

$$u(t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad t = 0..5 \quad (18)$$

We can use the convolution code specified by Equation 14 and Equation 15 to produce the corresponding codeword sequence:

$$v(t) = \begin{bmatrix} 00 & 11 & 01 & 11 \end{bmatrix}, \quad t = 2..5 \quad (19)$$

In the above example, note that the first two valid outputs for *v* occur at time $t = 2$.

ii) Decoding Methodology

There are various approaches for decoding convolutionally encoded data. Similar to block decoding, the maximum likelihood decoding approach compares the received sequence with every possible code sequence the encoding system could have produced. Given a received sequence and the state diagram of the encoding system, maximum likelihood decoding produces the most likely estimate of the transmitted vector, *v*. The Viterbi decoding algorithm (Sweeney, 1991; Dholakia, 1994) is a maximum likelihood decoding algorithm which uses a code trellis to estimate the transmitted vector given a received vector.

Another decoding approach uses syndrome decoding methods and a decoding window which consists of $m+1$ frames (Sweeney, 1991; Dholakia et al., 1995; Bitzer et al., 1998). The received sequence is treated like a block code and a syndrome value is generated for each received block. As in block codes, the value of the syndrome indicates the presence or absence of an error in the received sequence. Although not a maximum likelihood method, syndrome-based decoding of convolutional codes is more computationally efficient. Table-based codes (discussed in detail in the section which follows) make use of syndrome decoding techniques

(Dholakia et al., 1995; Bitzer et al., 1998).

d) Table-Based Codes

This section describes a specific method for implementing convolutional coding: table-based encoding and decoding. All methods described for table-based encoding and decoding are based on concepts developed and presented by Bitzer et al. (in Bitzer & Vouk, 1991; Dholakia et al., 1995; Bitzer et al., 1998).

i) Table-Based Encoding

The existence of a one to one mapping between data bits and parity bits is the foundation for table-based encoding. A set of *w*-bit data block must correspond uniquely to a set of *w*-bit parity block. Parity bits are the bits generated by the encoder and they make up the output vector *v*. For an (*n, k, m*) convolutional code:

$$w = n\frac{L-k}{n-k} \tag{20}$$

where

$$L = m + 1 \tag{21}$$

Table-based encoding is implemented as follows: based on the knowledge of the encoder and the parameters *n, k, L* we can construct an encoding table that associates each *w*-bit data sequence with a unique parity sequence. For binary data there are 2^w possible data sequences. Depending on the value of *w*, the encoding table can become extremely large. We can construct a reduced encoding table with only *w* data elements and the corresponding *w* parity elements. For the reduced encoding table each data sequence is *w*-bits long and contains a single bit equal to one in the *i*th position, where *i* goes from position one to position *w*. These *w* data sequences are the basis vectors (the fundamental vectors that can be combined to form all other vectors or sequences) for the set of all possible *w*-bit data sequences. For instance, the data sequences, or basis vectors, for a reduced encoding table with *w* = 3 are [100 010 001].

The encoding masks, which are equivalent to the generator vector, are used to form the corresponding parity bits for each *w*-bit data sequence. In the following example a *w* = 4 bit parity sequence is generated for the encoder illustrated in Figure 3. For a given data sequence, parity bits are generated by ANDing (multiplication modulo two) the data bits with the encoding mask and XORing (addition modulo two) the results. For the data sequence *databits* = 1000 and encoding mask C_1 and C_2 (note C_1 and C_2 are equivalent to g_1 and g_2 in Equation 14 and Equation 15) defined as:

$$C_1 = [101] \tag{22}$$

$$C_2 = [111] \tag{23}$$

the parity bits $P_{1,1}$, $P_{2,1}$, $P_{1,2}$, $P_{2,2}$ are calculated as follows:

$$\begin{array}{cccccc} 1 & & 0 & & 0 & 0 \\ 1 & & 0 & & 1 & \\ \hline 1 & + & 0 & + & 0 & \end{array} = P_{1,1} = 1$$

$$\begin{array}{cccccc} 1 & & 0 & & 0 & 0 \\ 1 & & 1 & & 1 & \\ \hline 1 & + & 0 & + & 0 & \end{array} = P_{2,1} = 1$$

Shift C_1 by *k* = 1 to get the next parity bit:

$$\begin{array}{cccccc} 1 & & 0 & & 0 & 0 \\ & & 1 & & 0 & 1 \\ \hline & & 0 & + & 0 & + & 0 \end{array} = P_{1,2} = 0$$

Shift C_2 by *k* = 1 to get the next parity bit:

$$\begin{array}{cccccc} 1 & & 0 & & 0 & 0 \\ & & 1 & & 1 & 1 \\ \hline & & 0 & + & 0 & + & 0 \end{array} = P_{2,2} = 0$$

From the example above, 1100 is the corresponding parity bits for data sequence 1000. Following the same procedure we obtain Table 2 as the resulting reduced encoding table for the encoder in Figure 3.

Table-based encoding works as follows for an encoding window *w* data bits wide:

1. Using the reduced encoding table, process the present w data bits into w parity bits.
2. Shift into the encoding window k new data bits and process the data bits in the window to produce a new block of w parity bits. The new parity bits overlap the old parity block with the first $w - n$ bits of the new block. These overlapping bits are identical.
3. Repeat the encoding process until all data bits have been processed.

Data Bits	Parity Bits
1000	1100
0100	0111
0010	1101
0001	0011

Table 2: Reduced Encoding Table

For table-based encoding to work, the proper encoding mask (derived from the generator vector) must be selected. The encoding mask must be chosen such that there exists a one to one correspondence between the data and parity bits. For error correcting systems, the encoding mask must produce codes that have sufficient error correcting capabilities for a given correction algorithm.

ii) Table-Based Decoding

Decoding tables are used to perform table-based decoding on received sequences or parity bits. A decoding table can be constructed if there exists a unique one to one mapping between data blocks and parity blocks. Therefore, table-based codes are invertible codes. The size of a decoding table for binary data would be 2^w. As in table-based encoding, we can construct a reduced decoding table which contains w elements instead of 2^w elements. Each of the parity sequences in the reduced table are w bits wide and the ith parity sequence has a single bit equal to one in the ith position, where i goes from one to w. For a reduced decoding table with w = 2, the parity sequences are: [10 01]

Given a reduced encoding table, we can construct the corresponding reduced decoding table as follows:

1. Sum the x w-bit parity blocks in the reduced encoding table needed to form the parity block for the reduced decoding table.
2. Sum the x w-bit data blocks associated with the x parity blocks from the encoding table to produce the w-bit data block that corresponds to the needed parity block in step one.
3. Continue this process for all w parity block entries in the reduced decoding table.

The following is an illustration of the above method using the reduced encoding table in Table 2. To construct the corresponding reduced decoding table, we must find the corresponding data blocks for the following four-bit parity blocks:

[1000 0100 0010 0001]

For parity block = 1000:

1. 1000 = 1100 + 0111 + 0011
2. The corresponding four-bit data block is: 1000 + 0100 + 0001 = 1101
3. After repeating steps one and two for the other three parity blocks, we obtain the resulting reduced decoding table shown in Table 3.

Parity Bits	Data Bits
1000	1101
0100	0101
0010	1011
0001	1010

Table 3: Reduced Decoding Table

Given a decoding window w parity bits wide, we can decode a parity stream as follows:

- Using the encoding table, a block of w parity bits is mapped to w data bits, producing the associated w-bit data block.
- n new parity bits are shifted into the decoding window.
- From the w parity bits now in the decoding window, produce the next block of data. The $w - k$ bits at the beginning of the new data block will

overlap the $w - k$ bits at the end of the previous data block.
- The above process repeats until all parity bits are decoded.

If there are no errors in the parity stream, the overlapping $w - k$ data bits will match producing zero values when exclusive-ORed bit by bit. But, if there is an error in the parity stream, the exclusive-ORing of the overlapping bits will result in non-zero values.

The results from performing the exclusive-OR operation on the overlapping data bits are called syndrome values or syndromes. A syndrome vector consists of a series of syndrome values. The syndrome vector is zero if there are no detectable errors in the parity stream (i.e. exact match between overlapping bits); otherwise, for binary data, the syndrome value is one. The number of syndrome values in a syndrome vector is equivalent to the number of overlaps used to determine the vector.

iii) Formation of the Gmask

The syndrome vector, which is used to detect errors in the parity stream, can be generated by repeated application of the decoding table to the parity stream, but this can become computationally expensive. The gmask provides an efficient method for syndrome vector generation. A gmask is a vector or sequence that is applied to the received sequence to generate syndrome vectors. The values that comprise the gmask are based on the codewords of the encoding system. The gmask is $w + n$ bits long. The following procedure describes how to generate the gmask, given a decoding table.

- Consider parity streams with single bit errors in each position of the n bit parity block $[P_1\ P_2 \ldots P_n]$
- Find the syndrome vector S_i for parity stream with error in parity bit P_i for $i = 1, 2 \ldots, n$.
- The gmask is formed by interleaving the n syndrome vectors generated from parity streams containing single-bit errors. For instance, if $n = 3$ and $S_1 = 001$, $S_2 = 101$, and $S_3 = 110$ then the gmask is be defined as

$$gmask = [\ S_1(3)\ S_2(3)\ S_3(3)\ S_1(2)\ S_2(2)\ S_3(2)\ S_1(1)\ S_2(1)\ S_3(1)\ ]$$

where $S_i(j)$ is the jth bit in the ith syndrome vector. The resulting gmask is:

$$gmask = [1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1]$$

There will be $n - k$ gmasks for an (n, k, m) code. Once the gmask has been constructed, it can be used to calculate the syndrome vector for received parity streams. To calculate the syndrome vector using the gmask:

- The gmask is ANDed with the first $w + n$ parity bits.
- The result is exclusive-ORed to produce a syndrome value.
- The received parity stream is shifted by n bits.
- The process is repeated until all syndrome values for the syndrome vector are produced. Each shift results in one syndrome value.

Based on the value of the syndrome vector, the received parity sequences can be used to decode the transmitted sequence to data or used to detect errors in the transmission. The concept of a decoding mask, the gmask, is employed in the convolutional coding model for the translation-initiation system.

Biological Communication

Information theoretic principles have been used to develop effective coding theory and cryptographic algorithms to successfully transmit information from a source to a receiver in engineered systems (Shannon & Weaver, 1949; Lin & Jr., 1983). Living systems also successfully transmit their biological information through genetic processes such as replication, transcription, and translation, where the genome of an organism is the contents of the transmission. The study of the information processing capabilities of living systems was revived in the later part of the 1980s, due to the increase in genomic data, which spurred a renewed interest in the use of information theory in the study of genomics (Roman-Roldan

et al., 1996; Sarkar et al., 1978; Fowler, 1979; Eigen, 1993). Information measures, such as entropy, have been used in recognition of DNA patterns, classification of genetic sequences, and various other computational studies of genetic sequences (Roman-Roldan et al., 1996; Palaniappan & Jernigan, 1984; Almagor, 1985; Schneider, 1991a; Schneider, 1991b; Altschul, 1991; Salamon & Konopka, 1992; Oliver et al., 1993; DeLaVega et al., 1996; Schneider & Mastronarde, 1996; Strait & Dewey, 1996; Pavesi et al., 1997; Loewenstern & Yianilos, 1997; Schneider, 1997; Schneider, 1999). Schneider et al.'s information theoretical methods contributed significant statistical evidence used to identify key regions on the mRNA leader sequence (Schneider et al., 1986). Recently Schultzaberger et al. developed an information-based method that incorporates key factors that influence translation initiation: Shine-Dalgarno sequence (SD), initiator codon, spacing between SD and initiator (Shultzaberger et al., 2001). Applying techniques from coding theory, a sub-field of information theory, is a logical next step in the study of the information processing mechanisms of genetic systems. While information theoretic analysis of genetic sequences and processes provides insight into informational properties of the genetic system, coding theoretic techniques provide the method for analysing and constructing genetic messages that survive mutational, environmental, and evolutional noise.

Application of coding theory to genetic data dates back to the late 1950s (Hayes, 1998; Golomb, 1962) with the mapping of the genetic code (the codon to amino acid mapping). Since then coding theoretic methods have been used for frame determination, motif classification, oligo-nucleotide chip design, and DNA computing (Loewenstern & Yianilos, 1997; Sengupta & Tompa, 2002; Kari et al., 1999). Loewenstern applied source coding (compression) methods to genomic sequences for the purpose of motif identification. Kari and colleagues apply circular coding methods to the forward encoding problem for DNA computing applications. The forward problem being, how can one encode an algorithm using DNA such that one avoids undesirable folding. Sengupta and Tompa approach the problem of oligo array design from a combinatorial design framework and use ECC methods to increase the fidelity of oligo array construction.

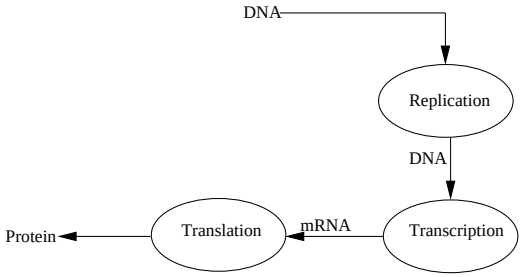

Figure 4: Central dogma of genetics.

a) Coding Theory and the Central Dogma of Genetics

The relationship between the error control coding process and protein translation may not be obvious. Figure 4 illustrates the central dogma of genetics. The central premise of genetics is that genes are perpetuated in the form of nucleic acid sequences but function once expressed as proteins (Lewin, 1995). 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 (Watson et al., 1987). There are three stages which transform genes from nucleic acid sequences to functional proteins.

- Stage 1: Replication - A DNA sequence replicates to form two identical DNA sequences.
- 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 (Watson et al., 1987).

- 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 (Lewin, 1995). The ribosome is an important part of the mechanism which translates mRNA information into proteins.

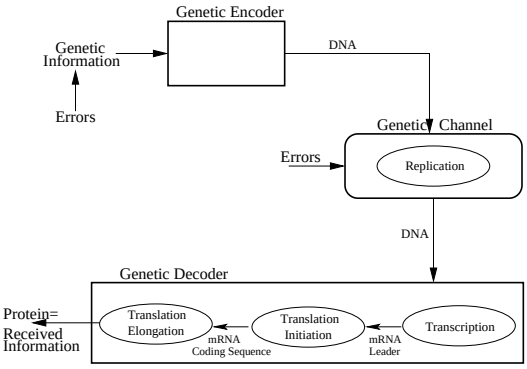

Figure 5: Central dogma of genetics as a coding system.

Researchers, such as Hubert Yockey who performed fundamental investigations of error correcting coding (ECC) properties of genetic systems, have explored the ECC properties of genetic sequences and systems (Yockey, 1992; May, 2002; Rosen & Moore, 2003; Liebovitch et al., 1996; MacDonaill, 2002). Several researchers have developed communication models for genetic processes (Gatlin, 1972; Yockey, 1992; Roman-Roldan et al., 1996; Battail, 1997; May, 2002). Our analogy of genetic information transmission to a communication system is illustrated in Figure 5. The un-replicated DNA sequence is the output of an ECC 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 (May, 2002). Incorporating the nested coding idea proposed by Battail (Battail, 1997), error-correcting decoding occurs in three phases represented by transcription, translation initiation, and translation elongation plus termination. We can see the parallel between this "genetic" communication system and a typical communication system (illustrated in Figure 1) by examining: the genetic encoder, the genetic channel, and the genetic decoder in our model.

i) The Genetic Encoder

It is not obvious what the biological mechanism is which behaves as a genetic encoder. But, it is fairly obvious that redundancy is present in the transmitted messages. For example, the number of DNA bases in an *E. coli* genome exceeds the number of bases needed to code for all the proteins produced by the *E. coli* genome. In general, there are sixty-four codons, three of which have specific control purposes. The sixty-one remaining codons code for twenty amino acids which are used to form proteins. The number of codons exceeds the number of amino acids represented, hence there exists redundant information in the genetic code. The coding rate and the type of code used to encode the DNA sequences are at the present unknown. In order to evaluate the validity of analysing translation from a coding theory perspective May et al. assume values for the coding rate.

ii) The Genetic Channel

If we assume that the DNA sequence is the output of a genetic encoder, the genetic channel is the medium or sequence of events that take the genomic DNA sequence and transmit it to the genetic decoder. Since channels do not change the alphabet of the message and only introduce errors, it is assumed that the replication process forms the genetic channel. As in an engineering communication channel, the genetic channel can introduce errors and noise into the transmitted message. However, it is also possible that the incoming message (DNA) is already "errored" due to some other biological event. During replication various types of errors can occur: deletion of DNA bases, insertion of incorrect DNA bases, and frame shifts. These errors or mutations in the genetic code can attenuate, corrupt, and distort the genetic signal which is vital to the survival of the organism. In this work the following assumptions about the genetic channel and the errors resulting from the channel are made:

- The probability of a nucleotide base error is independent from one nucleotide to the next.
- For all DNA bases (A,T,C,G), the probability of base *i* being replaced by base *j* is the same as the probability of base *j* being replaced by base *i*.

The above characteristics translate into a memoryless, symmetric channel. The replication channel can also exhibit characteristics of compound channels, resulting in DNA sequences with seemingly random base errors and sequences with regions of high mutation probability.

iii) The Genetic Decoder

Similar to nested decoders, transcription and translation are the two decoding phases. May et al.'s work focuses on the parallel between the translation process and the decoding step in an engineering communication system. Translation is the process by which genetic information stored in the messenger RNA is decoded into sequences of amino acids which form proteins. The ribosome is part of the cellular decoding mechanism. Once translation has been initiated, the ribosome (in conjunction with the tRNA and other protein factors) decodes the transmitted message by associating tri-nucleotide sequences (codons) in mRNA with corresponding tri-nucleotide sequences in charged tRNA (anti-codons). During the elongation phase, sequences of valid codewords will translate into viable proteins. By analysing the elongation phase of protein translation, we see that the ribosome, like a decoder in an engineering system, associates a fixed length received sequence or codeword with specific information.

During the initiation phase of translation, it is not clear what decoding method the ribosomal subunit employs. In initiation the small subunit along with initiation factor three (IF3) attach to the ribosomal binding site of messenger RNA. If we view the small subunit/IF3 complex as a decoding mechanism, we assert that error control coding theory could be used to classify the series of bases in that region as valid for translated sequences and invalid for untranslated sequences.

Analysis of several different *E. coli* mRNA ribosomal binding sites has revealed two common features in these sequences (Lewin, 1995):

- Presence of an initiation codon (AUG or less often GUG or UUG)
- Presence of the Shine-Dalgarno sequence: a short sequence complementary to the 3' end of the 16S rRNA (the rRNA hexamer: 3'...UCCUCC...5').

The initiation sequence protected by the bacterial ribosome is thirty-five to forty bases long (Lewin, 1995). Within this initiation sequence, not all six bases of the Shine-Dalgarno were present in each ribosomal binding site analysed; usually four to five bases in the sequence match the hexamer. Hence, the small subunit of the ribosome must have a mechanism for detecting the presence of valid codewords that are indicators for the initiation of protein translation. The decoding mechanism must be able to detect these valid codewords in the presence of noise introduced by the genetic channel. A mistake in decoding could result in the ribosome translating a protein sequence incorrectly, which is potentially detrimental to the organism.

As mentioned earlier, the encoding mechanism used in the genetic encoder is unknown. Therefore, we do not know the exact mechanism employed by the genetic decoder. By analysing key elements involved in initiating protein translation, it is hoped that we will gain insight into a possible decoding scheme used in the initiation phase of translation in *E. coli*. The key elements taken into consideration are: the 3' end of the rRNA, the common features of bacterial ribosomal binding sites, and base-pairing principles between the rRNA and the mRNA.

The coding alphabet must be derived from a finite field as in binary codes. Using base pairing, wobble pairing, and translation initiation information (Lewin, 1995) the RNA bases were mapped to the field of five as follows: Inosine(I) = 0, Adenine(A) = 1, Guanine(G) = 2, Cytosine(C) = 3, and

Thymine(T)/Uracil(U) = 4.
Multiplication and addition are modulo five. The RNA bases were defined so that in modulo five addition the sum of bases that form hydrogen pairs is zero. Hydrogen pairs or hydrogen bonds are weak, noncovalent bonds that hold macromolecules such as DNA and RNA together (Alberts et al., 1998). These definitions were used to construct the block code and convolutional code models for the protein translation initiation process.

In our work on biological coding theory, we illustrate how a coding theory framework can be used to analyse genetic processes and sequences (May, 1998; May et al., 1999; May et al., 2000; May, 2002). The next two sections describe some of our initial work.

b) Messenger RNA as a Block Code

If it is assumed that genetic information in DNA is encoded in a manner equivalent to block encoding, then the received message, the mRNA, can be viewed as a received parity sequence of a block encoded data stream. In the block code model, the genetic encoder is modelled as an (*n, k*) block code whose output is a systematic zero parity check code (Sweeney, 1991; May, 1998).

i) Genetic Encoder Model

Codewords of length $n = 5$ and $n = 8$ bases are developed based on the last thirteen bases of the 3' end of 16S ribosomal RNA (which contains the hexamer complementary to the Shine-Dalgarno sequence (Lewin, 1995)) and the proposed encoder model. The (5, 2) and (8, 2) models reflect the effect of two or more and three or more codons, respectfully. Specifically, the last thirteen bases of the 16S rRNA that interact with the Shine-Dalgarno domain and other sequences on the 5' untranslated mRNA leader, are (Lewin, 1995):

3'AUUCCUCCACUAG...5' (24)

Since our received sequence, the mRNA, contains the nucleotide sequence which base pairs with the 16S rRNA, we use the Watson-Crick complement of the thirteen base sequence in forming our codewords. The complement of the 3' end of the 16S rRNA is:

5'UAAGGAGGUGAUC...3' (25)

We select the $n - k$ parity symbols from all (n-k)-base sub-sequences of the thirteen base complement in Equation 25. For instance, if we desire a (5,2) code, we would select our parity symbols from all three-base sub-sequences of the thirteen base complement.

The three base parity sub-sequences are selected so that the following equation is satisfied:

$$\sum_{1}^{k} u_{genetic} + \sum_{1}^{n-k} ParityBases = 0 \quad (26)$$

where $u_{genetic}$ is the k-base information vector and *ParityBases* is the $n - k$ base parity vector. To illustrate, if we define the information sequence as $u_{genetic}$ = (C A), using the mapping from (a, iii, this part) the corresponding numerical representation is u = (3 1). We select a set of parity symbols such that $u_1 + u_2 + \sum_{1}^{3} ParityBases = 0$.
Hence (U A A) is selected as our parity bases. The resulting codeword is: *Codeword* = (3 1 4 1 1). The equivalent genetic codeword is: $Codeword_{genetic}$ = (C A U A A). We generate codewords for all possible k-base genetic information vectors. For a (5,2) code our information vectors would be drawn from every possible two-base RNA sequence; there are sixteen such sequences. A codeword is produced, as previously illustrated, for each possible two-base RNA sequence. If the resulting codeword satisfies Equation 26, then it is included in the codeword list (the codebook) otherwise it is excluded.

ii) Decoder for Model Verification

A minimum Hamming distance decoder, based on the systematic, zero-parity check encoding methodology, was designed to analyse the proposed block coding model. The analysis sequence is composed of: the thirty bases of the

mRNA leader sequence preceding the initiation (start) signal, the initiation signal (usually AUG), and twenty-seven bases from the coding region immediately following the initiation signal. The received sequence is an *n*-base subset of the analysis sequence.

The decoding process normally corrects the received sequence to the codeword with the lowest minimum distance value and recovers the transmitted information sequence, *u*. Since our objective is to analyse the coding model, the minimum distance is recorded for each received genetic sequence in the analysis stream. This distance is used to evaluate how well the proposed block coding model captures the biological aspects of the initiation process.

iii) Results

The *E. coli* K-12 strain MG1655 sequence data accession number U000096 (downloaded from the NIH site: ncbi.nlm.nih.gov) was used to test the model. Using the information in the GenBank data file, known and possible reading frames were divided into three sequence groups: translated, hypothetical, and non-translated. The translated sequence group contained open reading frames classified as protein producing regions. Hypothetical sequences were open reading frames that GenBank classified as hypothetical proteins. Open reading frames that did not appear in the GenBank annotation file as translated or hypothetical are included in the non-translated sequence group. Figure 6 shows the resulting mean minimum distance by position for the (5,2) block code model. The smaller the value on the vertical axis, the stronger the bond formed between the ribosome and the mRNA. Zero on the horizontal axis corresponds to the alignment of the first base of a codeword with the first base of the initiation codon.

As Figure 6 illustrates there is a significant difference among the translated, hypothetical and the non-translated sequence groups. For the translated and hypothetically translated sequence groups, a minimum distance trough occurs between the -15 and -10 regions. The -15 to 0 region contains large synchronization signals which can be used to determine valid protein coding sequences or frames. There are also smaller synchronization signals outside the -15 to 0 region which exhibit a weak oscillatory behaviour.

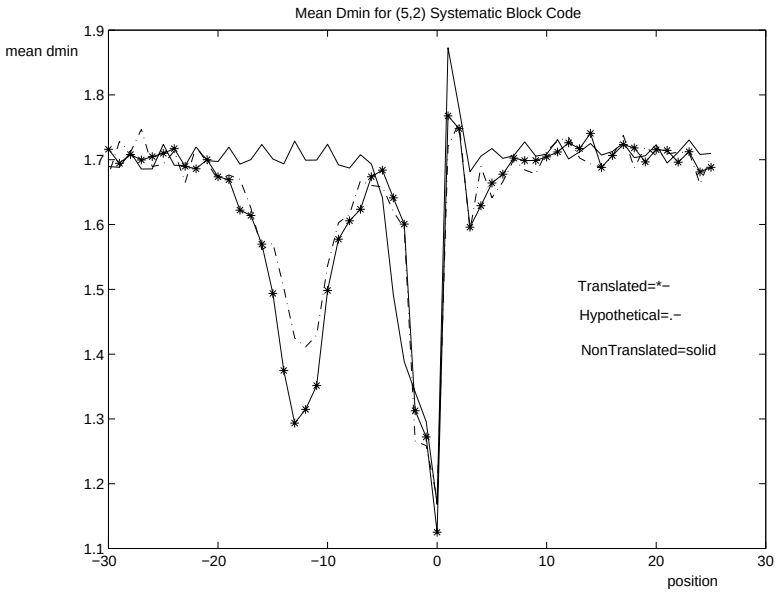

Figure 6: Average minimum Hamming distance values of mRNA leader sequences using the (5, 2) block decoding model.

c) Messenger RNA as a Convolutional Code

The second error-correcting coding model investigated is based on the principle hypothesis that the messenger RNA (mRNA) sequence can be viewed as a noisy, convolutionally encoded signal. The ribosome is functionally paralleled to a table-based convolutional decoder. The 16S ribosomal RNA (rRNA) sequence is used to form decoding masks for table-based decoding.

i) Genetic Encoding Framework

Convolutional coding produces encoded blocks based on present and past information bits or blocks. The modelling assumption is that genetic operations such as initiation and translation may involve "decisions" which are based on immediate past and immediate future information. This would allow error correction and other related functions. The convolutional code model views the ribosome as a mechanism with memory. Evaluating the messenger RNA as convolutionally encoded data allows the model to capture the inter-relatedness between the bases in a mRNA sequence. The formation of bonds between the mRNA and the 16S rRNA significantly influence the initiation of protein translation. When a base on the mRNA pairs with a base on the 16S rRNA, hydrogen bonds are formed. The greater the number of consecutive pairings formed between these two RNA molecules, the greater the probability of translation initiation. Every time the 16S ribosomal subunit attaches to the mRNA, a bonding pattern is formed. The bonding pattern that results in a positive signal is the bonding pattern with high numbers of consecutive hydrogen bonds. This process of locating regions on the mRNA which form high numbers of consecutive hydrogen bonds can be paralleled to locating parity blocks which produce zero syndrome vectors for a received parity stream.

The messenger RNA is modelled as a received parity sequence of a convolutional encoded data stream (Bitzer et al., 1992). We use the syndrome concept developed for table-driven decoding to check whether a mRNA translation initiation region can be interpreted using a covolutional coding model. In order to use the table-driven decoding model, we must define biological coding constructs which are analogous to the following coding concepts: the decoding mask, syndrome, and interpretation of syndrome values.

ii) Genetic Gmask Based on 16S rRNA

The gmask selects which bits are included in the exclusive-OR operation. For binary data, the bits in the decoding window associated with the gmask are the bits used to determine the syndrome vector. For the genetic model, the genetic gmask is derived from the 16S rRNA sequence:

3'AUUCCUCCACUAG...5'

The equivalent GF(5) mapping is:

3'...1 4 4 3 3 4 3 3 1 3 4 1 2...5'

The genetic gmask is formed from subsets of contiguous bases of the 16S rRNA. The subsets indicate which (n+w)-base region is being included in the exclusive-OR operation of the ribosome. Selecting subsets of the 16S rRNA corresponds to base pairing between regions of the 3' end of the 16S rRNA and regions within the mRNA sequence. Assuming a coding model with n=2, k = 1, L = 3 (m = 2), and w = 4, the length of the genetic gmask is $w + n$ or six. On average, five nucleotides on the mRNA leader complement pair with the exposed part of the 16S rRNA (Gold & Stormo, 1987). The coding parameters result in a genetic gmask length that reflects this. A gmask for the translation initiation system can be selected from a table of eight possible six-base genetic gmask values derived from the 16S rRNA (May, 1998). The eight gmasks are all the possible six-base sub-sequences of the 3' end of the 16S rRNA as indicated above.

For the chosen gmask, the syndrome values of a stream of mRNA codons can be calculated. The received mRNA parity (or codon) sequence includes the last thirty bases of the leader region, the initiation codon, and the first nine codons of the translated region:

$$mRNA = \left[b_{-30} \quad b_{-29} \quad \cdots \quad b_{-1} \quad A \quad U \quad G \quad b_{+3} \quad \cdots \quad b_{+29} \right] \tag{27}$$

with,

$$b_i = \left[A, G, C, U \right] \tag{28}$$

iii) Syndrome Calculation

The syndrome value for a given mRNA is calculated by ANDing the received codon bases with the genetic gmask and exclusive-ORing the result. Multiplication (AND) and addition (XOR) are modulo-five. For example, given the following mRNA sequence:

mRNA = [A U G U G A U C U C]

and the following six-base gmask (which is in essence an element of the Shine-Dalgarno sequence)

gmask = [C A C U A G]

the first three syndrome values are calculated as follows:

A U G U G A U C U C ⇐ mRNA
C A C U A G ⇐ gmask

The numerical equivalences are:

1 4 2 4 2 1 4 3 4 3 ⇐ mRNA
3 1 3 4 1 2 ⇐ gmask

3+4+1+1+2+2 = s_1 = 3

Shift by n=2:

G U G A U C U C ⇐ mRNA
C A C U A G ⇐ gmask

The numerical equivalences are:

2 4 2 1 4 3 4 3
3 1 3 4 1 2

1+4+1+4+4+1 = s_2 = 0

Note: this is an exact pairing match between the mRNA sub-sequence and the gmask

Shift by n=2 again:

G A U C U C ⇐ mRNA
C A C U A G ⇐ gmask

The numerical equivalences are:

2 1 4 3 4 3
3 1 3 4 1 2

1+1+2+2+4+1 = s_3 = 1

This work looks for a correlation between syndrome values and the position of the genetic gmask relative to the translation initiation codon.

iv) Distance Value Derivations

In binary table-driven decoding, a syndrome value of zero indicates that there are no detectable errors within the parity stream. For the translation initiation system, it would be ideal if syndrome values could be used to determine the presence or absence of valid ribosome binding sites. The presence of a valid ribosome binding site would indicate a valid translation initiation site.

In the example in the preceding section our syndrome vector S was [3, 0, 1]. The zero syndrome value occurred when an exact complement of the six-base genetic gmask appeared in the decoding window. Theoretically, a zero syndrome value should occur when an exact complement to the genetic gmask is present in the decoding window. But experiments indicate that the genetic gmask match value (the syndrome value resulting from the presence of an exact complement to the genetic gmask in the decoding window) does not always result in a zero syndrome value (May, 1998).

Since various gmasks yield different mask match values, syndrome values are normalized by transforming each syndrome value to represent the distance of the syndrome value from the genetic gmask match value. For example, if the genetic gmask match value is three and the resulting syndrome value is four then the normalized syndrome value or distance representation is one because 3 + 1 = 4.

Hence the normalization equation for a syndrome value *s*, given a genetic gmask match value *mm*, is as follows:

$$distance = [(s + 5) - mm] \bmod 5 \qquad (29)$$

The algorithm and table for conversion from syndrome value to distance value is presented in (May, 1998) for different values of *mm*. These normalized distance values are used to evaluate the convolutional coding model of the translation-initiation system.

v) Results

The *Escherichia coli* K-12 strain MG1655 sequence data accession number U000096 (downloaded from the NIH ftp site: ncbi.nlm.nih.gov) was used to test the model for two genetic gmask lengths. The data was divided into three sequence groups (translated, hypothetical, and non-translated) as previously described in (b, iii, this part). Figure 7 shows the frequency of the most frequent distance pattern among all possible two-symbol distance patterns $d_i d_j$, where distance values range from zero to four for a six-base gmask. The horizontal axis indicates position, with zero corresponding to the alignment of the coding mask with the first base of the initiation codon. The vertical axis indicates frequency (0.04 corresponds to four percent, the expected frequency of occurrence for a random, two-symbol distance pattern). As shown in Figure 7, the convolutional code model was able to distinguish between translated and non-translated sequence groups. The distinction among hypothetical and translated groups is also evident. The convolutional code model indicated relative occurrence of significant activity in the -15 area and following the -10 region. The Shine- Dalgarno sequence is located within this region (Lewin, 1995).

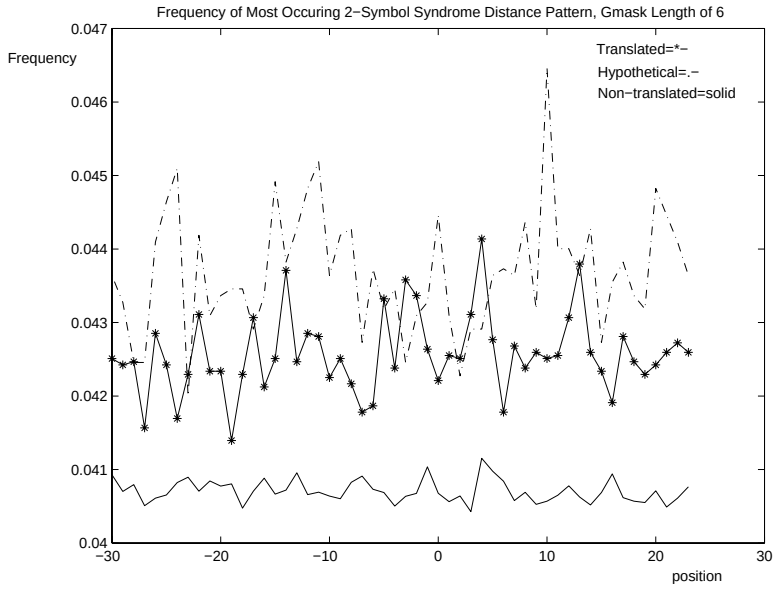

Figure 7: Frequency of two-pattern syndrome distance values of mRNA leader regions using the (2,1,2) convolutional code model.

d) Analysis of Coding-Based Models

Three issues were critical to analysing the effectiveness of each error-control model for translation initiation: (1) Recognition of regions within the mRNA leader sequence; (2) Distinction between translated and non-translated sequence groups; (3) Indication and recognition of the open reading frame construct. Both models distinguished translated sequence groups from the non-translated sequence group. They both also indicated the existence of key regions within the mRNA leader sequence. The block code model seemed to recognize the ribosomal binding site (the location of the Shine-Dalgarno sequence) more readily than the convolutional code model. The block code model also indicated the existence of a reading frame synchronization construct more so than the convolutional code model. Additional results for longer block codes and results for the longer gmask (twelve-base masks) are presented in May (1998). More detailed investigations of convolutional code models for translation initiation have been conducted (May, 2002).

Conclusion

The block code model is a sliding block code. Therefore a convolutional code more accurately depicts the behaviour of the ribosome as a decoder that incorporates memory in its translation (or decoding) decisions. The results of the error-control coding models suggest that it is possible to design a convolutional coding based heuristic for distinguishing between protein coding and non-protein coding genomic sequences by "decoding" the mRNA leader region. Results also imply that genetic systems may use methods that functionally parallel channel coding techniques to protect and detect genetic signals. The successful development and implementation of a channel coding model for the translation initiation system can lead to the development of powerful methods for identifying and manipulating protein coding sequences within a genome as well as further our understanding of translation regulatory mechanisms.

Acknowledgments

The work described is a result of research performed in close collaboration with Mladen A. Vouk, Donald L. Bitzer, and David I. Rosnick of North Carolina State University's Computer Science Department. The author performed a majority of the research while a graduate researcher under the direction of M. A. Vouk and D. L. Bitzer. The author would like to thank the reviewers for their constructive comments and suggestions.

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-AC04-94AL85000.

References

Alberts, B., Bray, D., Johnson, A., Lewis, J., Raff, M., Roberts, K. & Walter, P. (1998) *Essential Cell Biology: An Introduction to the Molecular Biology of the Cell*. Garland Publishing, Inc., New York.

Almagor, H. (1985) Nucleotide distribution and the recognition of coding regions in DNA sequences: an information theory approach. *Journal of Theoretical Biology*, 117, 127–136.

Altschul, S. F. (1991) Amino Acid substitution matrices from an information theoretic perspective . *Journal of Molecular Biology*, 219, 555–565.

Anderson, J. B. & Mohan, S. (1991) *Source and Channel Coding An Algorithmic Approach*. Kluwer Academic Publishers, Boston, MA.

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

Bitzer, D. L., Dholakia, A., Koorapaty, H. & Vouk, M. A. (1998) On Locally Invertible Rate-1/n Convolutional Encoders. *IEEE Trans. on Information Theory*, 44, 420–422.

Bitzer, D. L. & Vouk, M. A. (1991) A Table-Driven (Feedback) Decoder. In *Tenth Annual International Phoenix*

Conference on Computers and Communications, pp. 385–392.

Bitzer, D. L., Vouk, M. A. & Dholakia, *A. Genetic Coding Considered as a Convolutional Code*. North Carolina State University, Raleigh.

DeLaVega, F. M., Cerpa, C. & Guarneros, G. (1996) A mutual information analysis of tRNA sequence and modification patterns distinctive of species and phylogenetic domain. In *Pacific Symposium on Biocomputing*, pp. 710–711.

Dholakia, A. (1994) *Introduction to Convolutional Codes with Applications*. Kluwer Academic Publishers, Norwell, Massachusetts.

Dholakia, A., Bitzer, D. L. & Vouk, M. A. (1995) Table based decoding of rate one-half convolutional codes. *IEEE Trans. on Communications*, 43, 681–686.

Eigen, M. (1993) The origin of genetic information: viruses as models . *Gene*, 135, 37–47.

Fowler, T. B. (1979) Computation as a thermodynamic process applied to biological systems . *International Journal of Bio-Medical Computing*, 10 (6), 477–489.

Gatlin, L. L. (1972) *Information Theory and the Living System*. Columbia University Press, New York, NY.

Gold, L. & Stormo, G. (1987) Translational initiation. In *Escherichia coli* and *Salmonella typhimurium*, *Cellular and Molecular Biology*, pp. 1302–1307.

Golomb, S. W. (1962) Efficient coding for the desoxyribonucleic channel. *Proc. of Symposia in Applied Mathematics*, 14, 87–100.

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

Kari, L., Kari, J. & Landweber, L. F. (1999) Reversible molecular computation in ciliates. In *Jewels are Forever* pp. 353–363.

Lewin, B. (1995) *Genes V*. Oxford University Press, New York, NY.

Liebovitch, L. S., Tao, Y., Todorov, A. & Levine, L. (1996) Is there an Error Correcting Code in DNA? *Biophysical Journal*, 71, 1539–1544.

Lin, S. & , D. J. Costello Jr. (1983) *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, Inc., Englewood Cliffs, NJ.

Loewenstern, D. & Yianilos, P. N. (1997) Significantly lower entropy estimates for natural DNA sequences. In *Proceedings of the Data Compression Conference*.

MacDonaill, D. (2002) A Parity Code Interpretation of Nucleotide Alphabet Composition. *Chem. Communic.*, , 2062–2063.

May, E. E. (1998). *Comparative Analysis of Information Based Models for Initiating Protein Translation in* Escherichia coli *K-12*. Master's thesis NCSU.

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

May, E. E., Vouk, M. A., Bitzer, D. L. & Rosnick, D. I. (1999) Coding model for translation in E. *coli* K-12 . In *First Joint Conference of EMBS-BMES*.

May, E. E., Vouk, M. A., Bitzer, D. L. & Rosnick, D. I. (2000) The ribosome as a table-driven convolutional decoder for the *Escherichia coli* K-12 translation initiation system . In *World Congress on Medical Physics and Biomedical Engineering Conference*.

Oliver, J. L., Bernaola-Galvan, P., Guerrero-Garcia, J. & Roman-Roldan, R. (1993) Entropic profiles of DNA sequences through chaos-game-derived images . *Journal of Theoretical Biology*, 160, 457–470.

Palaniappan, K. & Jernigan, M. E. (1984) Pattern analysis of biological sequences. In *Proceedings of the 1984 IEEE International Conference on Systems, Man, and Cybernetics*.

Pavesi, A., Iaco, B. D., Granero, M. I. & Porati, A. (1997) On the informational content of overlapping genes in prokaryotic and eukaryotic viruses. *Journal of Molecular Evolution*, 44 (6), 625–631.

Roman-Roldan, R., Bernaola-Galvan, P. & Oliver, J. L. (1996) Application of information theory to DNA sequence analysis: a review. *Pattern Recognition*, 29 (7), 1187–1194.

Rosen, G. & Moore, J. (2003) Investigation of coding structure in DNA. In *ICASSP 2003*.

Salamon, P. & Konopka, A. K. (1992) A maximum entropy principle for the distribution of local complexity in naturally occuring nucleotide sequences. *Computers and Chemistry*, 16 (2), 117–124.

Sarkar, R., Roy, A. B. & Sarkar, P. K. (1978) Topological information content of genetic molecules – I . *Mathematical Biosciences*, 39, 299–312.

Schneider, T. D. (1991a) Theory of molecular machines. I. Channel capacity of molecular machines. *Journal of Theoretical Biology*, 148, 83–123.

Schneider, T. D. (1991b) Theory of molecular machines. II. Energy dissipation from molecular machines. *Journal of Theoretical Biology*, 148, 125–137.

Schneider, T. D. (1997) Information content of individual genetic sequences . *Journal of Theoretical Biology*, 189, 427–441.

Schneider, T. D. (1999) Measuring molecular information. *Journal of Theoretical Biology*, 201, 87–92.

Schneider, T. D. & Mastronarde, D. N. (1996) Fast multiple alignment of ungapped DNA sequences using information theory and a relaxation method. *Discrete Applied Mathematics*, 71, 259–268.

Schneider, T. D., Stormo, G. D., Gold, L. & Dhrenfeucht, A. (1986) Information Content of Binding Sites on Nucelotide Sequences. *Journal of Molecular Biology*, 188, 415–431.

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

Shannon, C. E. & Weaver, W. (1949) *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, IL.

Shultzaberger, R. K., Bucheimer, R. E., Rudd, K. E. & Schneider, T. D. (2001) Anatomy of *Escherichia coli* ribosome binding sites. *J. Mol. Biol.*, 313 (1), 215–228.

Strait, B. J. & Dewey, T. G. (1996) The Shannon information entropy of protein sequences . *Biophysical Journal*, 71, 148–155.

Sweeney, P. (1991) *Error Control Coding an Introduction*. Prentice Hall, New York, NY.

Watson, J., Hopkins, N., Roberts, J., Steitz, J. & Weiner, A. (1987) *Molecular Biology of the Gene*. The Benjamin Cummings Publishing Company, Inc., Menlo Park, CA.

Yockey, H. (1992) *Information Theory and Molecular Biology*. Cambridge University Press, NY, NY.

 Manuscript received: 09/09/03; revised:31/12/03 ; accepted: 14/01/04; number: 04-200401.