5-2010

Error Correcting Codes and the Human Genome.

Suzanne McLean Lyle
East Tennessee State University

Error Correcting Codes and the Human Genome

A thesis

presented to

the faculty of the Department of Mathematics

East Tennessee State University

In partial fulfillment

of the requirements for the degree

Master of Science in Mathematical Sciences

---

by

Suzanne McLean Lyle

May 2010

Debra Knisley, Ph.D., Chair

Jeff Knisley, Ph.D.

Michel Helfgott, Ed.D.

Keywords: coding theory, vectors, matrices, DNA, genome, genes.

ABSTRACT

Error Correcting Codes and the Human Genome

by

Suzanne McLean Lyle

In this work, we study error correcting codes and generalize the concepts with a view toward a novel application in the study of DNA sequences. The author investigates the possibility that an error correcting linear code could be included in the human genome through application and research. The author finds that while it is an accepted hypothesis that it is reasonable that some kind of error correcting code is used in DNA, no one has actually been able to identify one. The author uses the application to illustrate how the subject of coding theory can provide a teaching enrichment activity for undergraduate mathematics.

DEDICATION

I would like to dedicate this thesis first to God, who is good in all things, and who has given me the strength and clarity of mind to complete my masters degree program by holding me close during the storm. And second to my beloved daughter, Solana, who went home to be with the Lord November 2009. When I was not sure I could go on, I drew strength from the knowledge that she was proud of me for returning to school and that she would be disappointed in me if I did not finish because of her early homegoing.

ACKNOWLEDGMENTS

I would like to acknowledge my wonderful and patient committee chair, Dr. Debra Knisley, who helped me determine a thesis topic and stuck with me through some rough times to help me graduate on time. Also, to my committee members, Dr. Michel Helfgott and Dr. Jeff Knisley who gave me sound advice and helped shape my final thesis. And finally to all my family, friends, professors and coworkers who supported and believed in me. Thank you to all.

CONTENTS

LIST OF FIGURES

1 CODING THEORY

1.1 Introduction to Coding Theory

Error correcting codes are components of a field called coding theory which is of interest in engineering, computer science, and mathematics. Coding theory is the study of transmitting a message effectively and accurately. The process involves a starting place, called the **information source**, sending some kind of information, called the **message word**, across a distance, called the **communication channel**. For accuracy, the message word is encoded by the **transmitter** thus creating a **codeword**. This codeword is received and decoded by a **receiver** and the message word is then sent to its intended destination called the **information sink**. Coding theory is necessary because the communication channels these messages are transmitted across generally contain some kind of interferences, called **noise**, which may cause the message to be received incorrectly [7]. Figure 1 below illustrates this process:

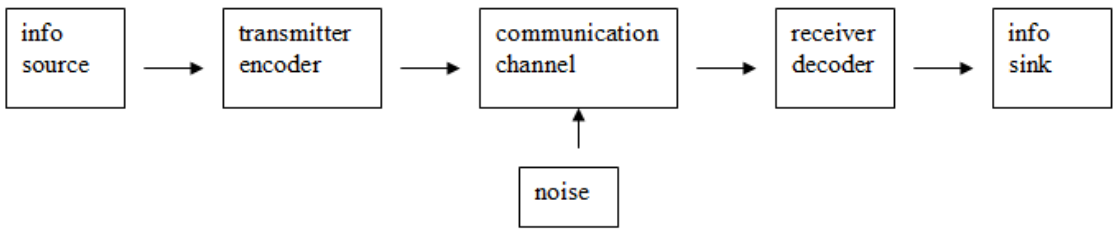

Figure 1: Coding Theory Process

To better understand Figure 1, consider a specific example such as receiving images from the Voyager spacecraft. The Voyager would be the information source and the

image would be the message word. The transmitter, a computer, would encode the message word and send the created codeword through a physical medium, the communication channel. In the case of the Voyager, this channel would be outer space. Undesirable circumstances in the channel, such as sunspots or meteor showers, are called noise and can cause the word received to be different from the codeword sent. The receiver, another computer, receives the word and uses the code to detect and correct any errors before sending the original message to the information sink, NASA.

Coding theory, as well as information theory, began with the ideas published in the Bell System Technical Journal by Claude Shannon in the 1948 paper, *The Mathematical Theory of Communication* [16]. Coding theory became necessary due to the invention of computers. Early computers were based on large banks of mechanical relays and their reliability was very low compared to today's computers. If a single relay failed to work, the entire calculation was wrong; engineers developed ways to detect faulty relays so they could be replaced. Richard Hamming, also at Bell Laboratories, added to coding theory the idea that if a machine was capable of knowing there was an error then perhaps it could also correct the error. The Hamming code is both an error detecting and error correcting code [3].

At the same time that Shannon and Hamming were developing coding theory in the United States, John Leech was inventing similar codes in Cambridge, England, in his work on Group Theory [13]. As stated in [4], group theory is the branch of mathematics that answers the question, "What is symmetry?" With the group theory codes, John Leech also helped develop the field of coding theory.

Coding theory is a very modern field of study. The transmission of data correctly

is very important to many different industries. A few specific examples of the uses of coding theory are the transmission of data and images from space, the transmission of financial data related to credit cards, and the minimization of noise from compact disc recordings.

1.2 Constructing a Code

To transmit a message, it must first be converted into a sequence of numbers. This work will deal strictly with binary codes to do this conversion. A **binary code** is a representation of information using a sequence of zeros and ones. Also, while there are many classes of codes, this work will only be considering linear codes.

Definition: A linear code C is a code that contains the zero word and is closed under addition of words [7].

Example 1: $C = \{000, 111\}$ is a linear code since $000 \in C$ and $000 + 000 = 000 \in C,\ 000 + 111 = 111 \in C,\ 111 + 000 = 111 \in C$, and $111 + 111 = 000 \in C$.

A linear code contains message words from the original message plus a sequence of **parity check digits** added onto the end of the message word to encode it. The **length** of a word is the number of digits, zeros or ones, in the word. The **length of the message word** is denoted k; the **length of the parity check** is denoted $n-k$. Together, the message word and the parity check digits make up the codeword. The **length of the codeword** is denoted n, and the **number of codewords in a code** is denoted $|C| = 2^k$.

Thus a linear code has the following properties:

- length of codeword: n
- length of message word (also called dimension): k
- length of parity check: $n - k$
- number of codewords: $|C| = 2^k$

In Example 1, $|C| = 2,\ n = 3,\ k = 1$, and $n - k = 2$.

An advantage to using linear codes is that the distance is easier to find. The distance, often referred to as the **Hamming distance**, is the number of positions in which two codewords disagree. In Example 1, the distance between the codewords $000, 111$ is $d(000, 111) = 3$. The **distance of a linear code** is the minimum distance among all pairs of codewords in the code.

Another important term when discussing linear codes is the **Hamming weight**. The weight of a codeword is the number of times the digit 1 occurs in the word. In Example 1, the weight of the codeword 111 is $wt(111) = 3$.

These two ideas of distance and weight can be used together to make finding the distance of a linear code easier to calculate. The distance of a linear code is equal to the minimum weight of any nonzero codeword [7]. The distance of $C = \{000, 111\}$ in Example 1 is the weight of the nonezero codeword, 111. Thus $d(C) = 3$.

One method to encode a set of messages, i.e. to construct a linear code C, is to determine a generator matrix G for C. A **generator matrix** transforms the message into a code.

Definition: If C is a linear code of length n and dimension k, then any matrix whose rows form a basis for C is called a generator matrix G for C [7].

A generator matrix is a $k \times n$ matrix of the form $[I_k, X]$ where I_k represents a $k \times k$ identity matrix augmented to a $k(n-k)$ matrix X. Example 2 illustrates a method for determining a generator matrix G for a given code C.

Example 2: Consider the linear code $C = \{00000, 10001, 01011, 00111, 11010, 01100, 10110, 11101\}$ with $n = 5,\ k = 3$, and $n - k = 2$. Construct a generator matrix G of C by considering $S = \{11101, 10110, 01011, 11010\} \subset C$. Note, $S \neq \{0\}$ and S must contain at least k codewords.

Let the codewords in S be the rows of a 4×5 matrix, A.

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix} \tag{1}$$

A reduces to reduced row echelon form, $RREF(A)$. Recall, **reduced row echelon form** is a matrix in row-echelon form with all pivots equal to 1 and with zeros above as well as below each pivot [6].

$$RREF(A) = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{2}$$

The nonzero rows of RREF(A) can be used to form the generating matrix G for the code C.

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \tag{3}$$

To generate the code C, multiply a message word by the generator matrix G. Since $k = 3$, the message words to be encoded are all the possible binary sequences of length 3. In other words, let $K = \{0, 1\}$ and let K^n be the set of all binary words of length n such that K^n satisfies the conditions of a vector space. Then for $k = 3$, $K^3 = \{000, 100, 010, 001, 110, 011, 101, 111\}$ is the set of message words to be encoded. Choose one of these message words, say $v = 110$, and multiply by G to create a codeword in the code C.

$$vG = \begin{pmatrix} 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \end{pmatrix} \tag{4}$$

The codeword for the message word $v = 110$ is 11010. When this procedure is repeated for all K^3, the code C is generated: $C = \{00000, 10001, 01011, 00111, 11010, 01100, 10110, 11101\}$.

To summarize, encoding a message in coding theory can be viewed mathematically as performing a vector transformation in a vector space as shown in Figure 2. The message word can be viewed as an input vector in $\mathbf{R}^k$. By multiplying the message word to the generator matrix, G, one is performing a transformation from $\mathbf{R}^k$ to $\mathbf{R}^n$. The resulting codeword is the output vector in $\mathbf{R}^n$.

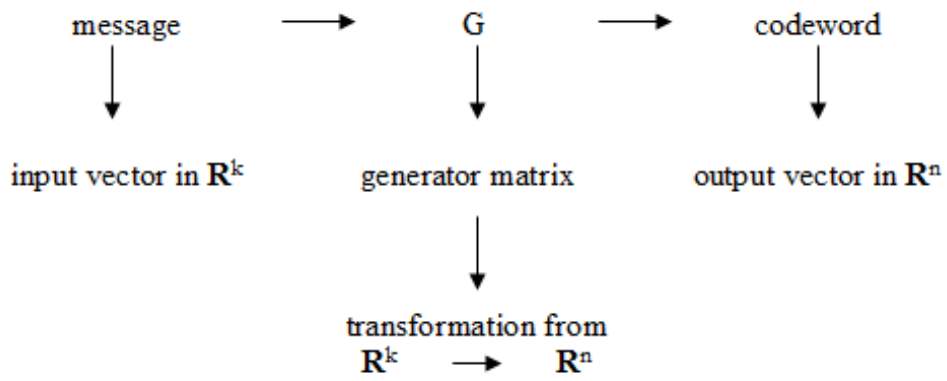

Figure 2: Correlation of Coding Theory to Vector Spaces

1.3 Linear Independence and Linear Codes

Consider $S = \{110, 011, 101, 111\}$ where S is a subset of $\mathbf{R}^3$. The **linear span** <S> is the set of all linear combinations of vectors in S and as such generates a linear code C.

Theorem 1: For any subset S of $\mathbf{R}^n$, the code $C =< S >$ generated by S consists precisely of the following words: the zero word, all words in S, and all sums of two or more words in S [7].

Example 3: Given $S = \{110, 011, 101, 111\}$. By applying Theorem 1, the code C can be generated. The code C woudl consist of the zero word, all the words in S and all the sums of two or more words in S. So $C = \{000, 110, 011, 101, 111, 100, 010, 001\}$.

If S is linearly independent, then it is a basis for code $C =< S >$. A set of vectors are a **basis** for C if it spans C and is linearly independent. If S is linearly dependent and $S \neq \{0\}$ then it contains a largest linearly independent subset B such that B is

a basis for code $C =< S >$.

A set of vectors is **linearly independent** if none of them can be written as a linear combination of finitely many other vectors in the collection.

Example 4: To test $S = \{110, 011, 101, 111\}$ for linear independence, consider

$$a(110) + b(011) + c(101) + d(111) = 000.$$

This equation yields the system of scalar equations:

$$a + c + d = 0$$

$$a + b + d = 0$$

$$b + c + d = 0$$

which yields the solutions $d = 0$ and $a = b = c = 1$ or 0. Therefore, S is a linearly dependent set.

Since $101 = 110 + 011$, we will discard 101 from S to create $S' = \{110, 011, 111\}$. To test $S' = \{110, 011, 111\}$ for linear independence, consider

$$a(110) + b(011) + c(111) = 000.$$

This equation yields the system of scalar equations:

$$a + c = 0$$

$$a + b + c = 0$$

$$b + c = 0$$

which yields the solutions $a = b = c = 0$. Thus, S' is a linearly independent set and a basis for $C = < S' >$.

It is important in coding theory to determine if a set is linearly independent or linearly dependent. It is true that any set of vectors containing the zero vector is linearly dependent. Also, in the previous section, the rows of the generator matrix G are a linearly independent set. All generator matrices are linearly independent.

1.4 Parity Check Matrices

A parity check matrix H is used to detect any errors once a codeword is received. Given a received word w, if $wH = 0$ then $w \in C$ and no errors have occured. If $wH \neq 0$, then an error has occurred.

Theorem 2: A matrix H is a parity check matrix for some linear code C if and only if the columns of H are linearly independent [7].

The relationship of H to the generator matrix G is given below:

$$\text{For } G_{k\times n} = \begin{pmatrix} I_k & X \end{pmatrix}, H_{n\times(n-k)} = \begin{pmatrix} X \\ I_{n-k} \end{pmatrix} \tag{5}$$

Thus the generator matrix G is a $k \times n$ matrix with the $k \times k$ identity matrix I_k augmented to a $k \times (n-k)$ matrix X. For the $n \times (n-k)$ parity check matrix H, the same $k \times (n-k)$ matrix X is put over a $(n-k) \times (n-k)$ identity matrix I_{n-k}.

Example 5: Consider $C = \{00000, 10001, 01011, 00111, 11010, 01100, 10110, 11101\}$ with $k = 3$, $n - k = 2$ and $n = 5$. The generator matrix G for C is as follows:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} I_k & X \end{pmatrix} \tag{6}$$

Then

$$H = \begin{pmatrix} X \\ I_{n-k} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{7}$$

Suppose the received word is $w = 11010$. Then

$$wH = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \end{pmatrix}, \text{ so w} \in \text{C}. \tag{8}$$

If $w = 11110$, then

$$wH = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix}, \text{ so w} \notin \text{C}. \tag{9}$$

Recall the distance d of a linear code C is the minimum weight of any nonzero codeword. The distance of the linear code C from Example 5 is $d = 2$.

One can also determine the distance of a linear code from the parity check matrix.

Theorem 3: Let H be a parity check matrix for a linear code C. Then C has distance d if and only if any set of $d - 1$ rows of H is linearly independent and at least one set of d rows of H is linearly dependent [7].

Consider the parity check matrix H for the linear code C with $d = 2$ from Example 5:

$$H = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{10}$$

By Theorem 3, any $d - 1 = 2 - 1 = 1$ rows of H must be linearly independent. Since a single vector is always linearly independent, this statement is true. Also, at least one set of $d = 2$ rows in H is linearly dependent. This is also true since $01 + 01 = 00$ and $11 + 11 = 00$. Thus, by parity matrix H, $d = 2$ for C.

1.5 Error Correcting Codes

In the previous section, it was shown that the parity check matrix H could detect an error in a received word. Therefore, C in Example 5 is an error detecting code. In error detecting codes, H can determine that a received word is not a codeword but cannot determine which codeword was sent. To determine which codeword was sent, a received word w must have a nearest neighbor in the code C. A **nearest neighbor codeword** would be a codeword v in C that is closer to w than any other codeword [7]. In terms of distance, $d(v, w)$ is less than the distance between w and any other codeword in C.

For the linear code C in Example 5, $d = 2$. Thus, each codeword in C differs in at least two positions. Figure 3 is a diagram of two codewords, v_1 and v_2, of code C in example 5 where $d = 2$:

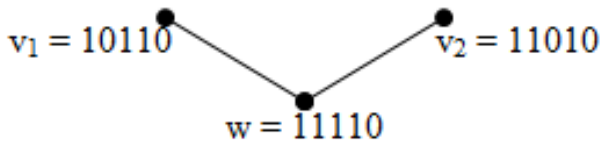

Figure 3: Node Graph for $d = 2$

For $w = 11110$ in Figure 3, there is not a nearest neighbor codeword in C. In such cases, the decoder would not correct w but would either arbitrarily choose one of the codewords closest to w or request a retransmission. However, in many occurrences, neither of these choices is reasonable. In our Voyager example from Section 1.1 of this work, precision would be desired and retransmission would be costly and likely impossible.

The parity check matrix H in Example 5 also indicated that C was a single error detecting code. For $w = 11110$, $wH = [11]$ indicating an error. But because both rows 2 and 3 of H are [11], it is impossible to determine if the error occurred in the second or third digit of w. If it was in the second digit then w would be $v_1 = 10110$. If the error was in the third digit then w would be $v_2 = 11010$.

For a linear code C to be a single error correcting code, d would need to be at least 3. Then a received word w_1 or w_2 would be closest to only one codeword as shown in Figure 4:

Figure 4: Node Graph for $d = 3$

One can build an error correcting code with the knowledge that $k = 3$ and $d = 3$. Recal thatl $d = 3$ means that each of the codewords differ in at least three places.

Example 6: Construct an error correcting code given $k = 3$ and $d = 3$. Starting with the zero word, build a code by adding digits as necessary to $K^3 = \{000, 100, 010, 001, 110, 011, 101, 111\}$ taking care that $d \geq 3$ between any two codewords and that the minimum weight of any codeword is 3.

$$\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{array} \tag{11}$$

Thus an error correcting code with $k = 3$ and $d = 3$ is $C = \{000000, 100110, 010011, 001101, 110101, 011110, 101011, 111000\}$. However, building a code in this manner is time consuming and tedious work. It is made easier with the use of parity check equations.

Definition: The columns of the parity check matrix, H, are the coefficients of a

system of linear equations whose solutions are precisely the codewords in C. These linear equations are called parity check equations [14].

Example 7: Consider the $[6, 3, 3]$ code from Example 6. A codeword could be written as $a_1, a_2, a_3, a_4, a_5, a_6$. The parity check digits, a_4, a_5, a_6, can be written as parity check equations of the message digits, a_1, a_2, a_3 in such a way that the equation can be set equal to zero using addition base 2. For the above example, $a_4 = a_1 + a_2, a_5 = a_1 + a_3$, and $a_6 = a_2 + a_3$. To verify that these equations can be set equal to zero consider

$$v = a_1, a_2, a_3, a_4, a_5, a_6 = 011110.$$

For a_4, the equation would be

$$1 = 0 + 1 \longrightarrow 0 + 1 + 1 = 0.$$

For a_5, the equation would be the same,

$$1 = 0 + 1 \longrightarrow 0 + 1 + 1 = 0.$$

And for a_6,

$$0 = 1 + 1 \longrightarrow 1 + 1 + 0 = 0.$$

Example 8: Consider the parity check matrix, H, from Example 5:

$$H = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{12}$$

Per the definition of parity check equations, the columns of H are the coefficients of the parity check equations. So $01110 = 0a_1+1a_2+1a_3+1a_4+0a_5 = 0 \longrightarrow a_4 = a_2+a_3$ and $11101 = 1a_1 + 1a_2 + 1a_3 + 0a_4 + 1a_5 = 0 \longrightarrow a_5 = a_1 + a_2 + a_3$.

1.6 Hamming Code

There are many types of error correcting codes. This work will discuss a special case of error correcting linear codes called the Hamming code. Hamming codes are considered to be perfect single error correcting codes [7].

Definition: Per [7], a code C of length n and odd distance $d = 2t + 1$ is called a perfect code if C attains the Hamming bound:

$$|C| = \frac{2^n}{\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}} \tag{13}$$

Definition: A Hamming code is defined as a binary code of length $n = 2^r - 1, r \geq 2$, where $r = n-k$, with parity check matrix H whose rows consist of all nonzero binary vectors of length r, each used once [11].

Since a Hamming code is a linear code it has the same properties with slight modifications to some of the notation. Recall that the length of a word is the number of digits it contains.

Properties of a Hamming Code:

- number of parity check digits: $n - k = r$
- length of message word: $k = 2^r - 1 - r, r \geq 2$

- length of codeword: $n = k + r = 2^r - 1, r \geq 2$
- number of codewords: $|C| = 2^k$
- distance of a code: $d = 2t + 1$
- number of errors corrected by a code: t

When $t = 1$, Hamming codes are single error correcting codes with $d = 3$. Another notation for linear codes such as the Hamming code is $[n, k, d]$ codes.

Example 9: To create a $[7, 4, 3]$ Hamming code C, first create the parity check matrix H which is a $n \times r$ matrix such that any $d - 1$ rows are linearly independent and at least one set of d rows is linearly dependent.

$$H_{7x3} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{14}$$

For H, any $d-1 = 2$ rows do not add to be zero but $d = 3$ rows $111+101+010 = 000$. The generating matrix G for the above the parity check matrix H is

$$G_{kxn} = G_{4x7} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \tag{15}$$

Using G, the codewords for C can be generated. Recall, since $k = 4$, the message words to be encoded are all the possible binary sequences of length 4, denoted by K^4. Let $v_1 = 1101 \in K^4$.

$$v_1 G = \begin{pmatrix} 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \quad (16)$$

Repeat this procedure for all elements in K^4 and the resulting code is $C = \{0000000, 1000111, 0100101, 0010011, 0001110, 1100010, 0110110, 0011101, 1010100, 0101011, 1001001, 1110001, 0111000, 1011010, 1101100, 1111111\}$.

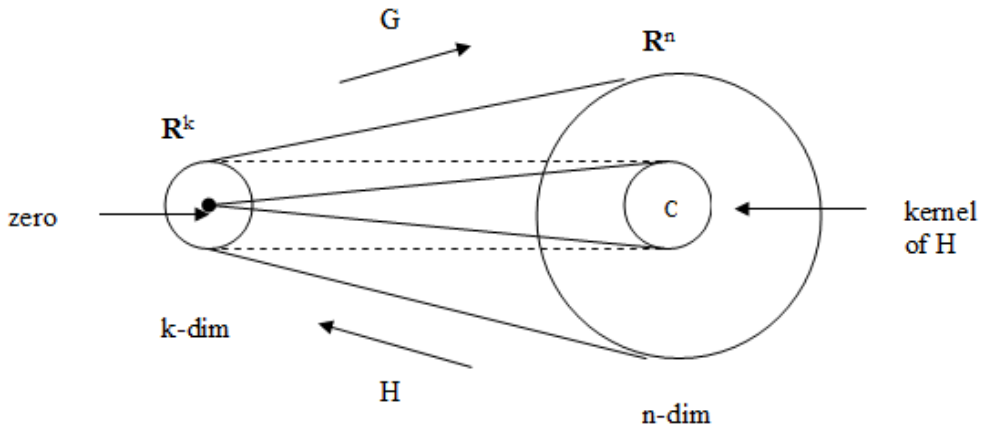

Figure 5: Summary of Coding Theory Process

To summarize, we return to the idea of vector spaces. Figure 5 is a diagram of the process of coding theory. The message word is an input vector in $\mathbf{R}^k$. By multiplying the message word to the generator matrix, G, one is performing a transformation

from $\mathbf{R}^k$ to $\mathbf{R}^n$. The resulting codeword is an output vector in $\mathbf{R}^n$. The parity check matrix H takes the output vectors in the $n - dimension$ vector space and translates them back to the $k - dimension$ vector space. The code is the kernel of H since it translates back to zero.

2 DNA, GENOME, AND GENES

2.1 Review of DNA, Genome, and Genes

One of the purposes of this work is to relate the study of coding theory to the human genome. We begin with a short review of DNA, genome, and genes.

DNA, or deoxyribonucleic acid, is a nucleic acid that provides long term storage of the genetic instructions used in the development and maintenance of all known living organisms and some viruses [1]. DNA has been compared to a code since the instructions in DNA construct other component of cells such as proteins and RNA molecules [1]. DNA exists as base pairs of molecules that are held tightly together. The four bases of DNA are adenine (A), cytosine (C), guanine (G), and thymine (T) [1]. These bases are classified into two types of base pairs - AG and CT [1]. So it is reasonable to think of these bases pairs in terms of a binary linear code where, without loss of generality, AG = 0 and CT = 1.

DNA is organized into long structures called chromosomes. The set of chromosomes in a cell make up its genome. The human genome has approximately three billion base pairs of DNA arranged into forty-six chromosomes [1]. The information carried by DNA is held in the sequence of pieces of DNA called genes. Humans have approximately 25,000 genes. Each gene has an average length of 3000 bases [1].

A gene contains both coding sequences that determine what the gene does and non-coding sequences that determine when the gene is active or expressed [1]. When a gene is active, the DNA of the gene undergoes two processes to become a protein transcription and translation [1]. To form RNA molecules only transcription is

needed. First, in transcription, the codons of the gene are copied into messenger RNA (mRNA). Codons are three-letter words formed from a sequence of three nucleotides to form the genetic code. Next, the RNA is decoded by a ribosome to translate the RNA into protein [1].

In humans, barely 2% of the genome consists of protein-coding DNA; much of the DNA in the genome is without an identified function [17]. However, scientists are now finding out that this extra DNA, once called *junk DNA* [17], plays important roles in the regulation of gene activity although no one knows to what extent. Some scientists have found that the sequence of the syllables in this DNA is not random at all but contains some kind of coded information [17]. But, again, the code and its function are as of yet unknown [17]. An interesting question then is: Could this *junk DNA* be used as parity check digits for the genetic code? Particularly, could you consider the gene as a message word (thus $k = 3000$) and determine how many parity checks you would need to create a $[n, k, d]$ linear code?

2.2 Coding Theory and the Human Genome

Mathematics and biology have had a long history, but recently gained in popularity with the sequencing of the human genome which was declared finished by the Human Genome Project in 2007 [5]. Mathematical biology studies the mathematical representation, treatment, and modeling of biological processes using a variety of applied mathematics techniques and tools [1]. In particular, genomic coding theory aims at applying concepts and techniques from the field of coding theory to problems from the field of molecular biology. Genomic coding theory is motivated by the redundant

structure of genetic code, the existence of large evolutionary conserved non-coding regions, and the existence of special sequences in coding regions [5].

Formalizing principles of genetic error correction into a coherent formal theory has been elusive for decades but several researchers have added contributions to the field. Yockey proposed one of the first models for gene expression using encoding and decoding concepts from communication theory in 1992 [5]. Liebovitch et al. developed the first efficient method to scan through DNA sequences to determine whether some linear block code was present [5]. Years later, Rosen built on Liebovitch's results and constructed a method for the detection of linear block codes that accounts for possible insertions and deletions in the DNA sequences [5]. Neither Rosen nor Liebovitch was able to support the existence of simple error correcting codes in DNA. However, as stated in by Liebovitch [8], if digital error correcting schemes are present in DNA, they may be more subtle than such simple linear block codes [8]. In 2006, Battail argues that due to the size of the human genome being far larger than needed there could exist nested error correcting codes in the DNA [5]. Other hypotheses have been a parity check code interpretation of nucleotide composition by MacDonaill and the use of block and convolutional codes to model the process of translation initiation by May et al [5].

Thus, can one prove the existence of some form of error correcting code in the structure of DNA? Many researchers are intrigued by the possibility of a connection between coding theory and DNA, but no one has managed to find it yet. The main challenge lies in the multilevel structure of the genetic error correction system and interactions not only among the different levels but also among other sub-systems in

the cell. Since evolution has had a lot of time to optimize its information transmission system, it might be a very complex code [5].

3 USING CODING THEORY AND IT'S REAL LIFE APPLICATIONS AS ENRICHMENT FOR STUDENTS OF MATHEMATICS

In this work, we have studied error correcting codes and how researchers are trying to determine if there is a connection between coding theory and the non-coding DNA in the genome. The field of genomic coding theory has emerged from these questions.

Coding theory, whose mathematical basis can be found in undergraduate linear algebra, is a very current field of study. Such a clear connections between abstract math ideas and real life applications can serve as powerful motivators for the gifted student to pursue a degree in mathematics. The author proposes using the subject of coding theory as a teaching enrichment activity for undergraduate mathematics to encourage these students. The author has included sample teaching modules for this enrichment activity in the Appendix.

Linear algebra is generally taught as a sophomore level course. Coding theory could then be offered as a mathematic elective in the junior year or as a enrichment activity in linear algebra or for mathematics honor students.

BIBLIOGRAPHY

[1] B. Alberts, D. Bray, A. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walters, *Essential Cell Biology: An Introduction to the Molecular Biology of the Cell*, First Edition, New York, Garland Publishing, Inc., c1998.

[2] G. Battail, *Information theory and error correcting codes in genetics and biological evolution*. Introduction to Biosemiotics, Springer, November 2006

[3] R. Calderbank, Interview with Neal Sloane, *IEEE Information Theory Society Newsletter*, **47** (1997), 3-4, 35-37

[4] N. Carter, *Visual Group Theory*, First Edition, USA, Mathematical Association of America, c2009.

[5] Z. Dawy, P. Hanus, J. Weindl, J. Dingel, F. Morcos, In genomic coding theory. *European Transactions on Telecommunications*, **18** (8)(2007):873-879.

[6] J. Fraleigh, R. Beauregard, *Linear Algebra*, 3rd Edition, USA, Addison Wesley Publishing Company, Inc., c1995.

[7] D. Hoffman, D. Leonard, C. Lindner, K. Phelps, C. Rodger, J. Wall, *Coding Theory: The Essentials*, First Edition, New York, Marcel Dekker, Inc., c1991.

[8] L. Liebovitch, Y. Tao, A. Todorov, L. Levine, Is There an Error Correcting Code in the Base Sequence in DNA?, *Biophysical Journal*, **71**(3)(September 1996):1539-1544.

[9] D. Mac Donnaill, Why nature chose A, C, G, and U/T: An error-coding perspective of nucleotide alphabet composition. *Origins of Life and Evolution of the Biosphere*, **33** (October 2003): 433-455.

[10] K. MacPherson, *Research Team Finds Important Role for Junk DNA*, News at Princeton University, viewed 10, March, 2010, http://www.princeton.edu/main/news/archive/S24/28/32C04/index.xml?section=topstories

[11] F. MacWilliams, N. Sloane, *The Theory of Error-Correcting Codes*, First Edition, New York, North-Holland, c1977

[12] E. May, M. Vouk, D. Nitzer, and D. Rosnick, An error-correcting code framework for genetic sequence analysis. *Journal of the Franklin Institute*, **34** (January-March 2004):89-109.

[13] R. Pinch, Coding theory: the first 50 years, *Plus Magazine*, **3** (1997), http://plus.maths.org/issue3/codes/index.html

[14] S. Roman, *Coding and Information Theory*, First Edition, Harrisonburg, R. R. Donnelley and Sons Co., c1992

[15] G. Rosen, Examining coding structures and redundancy in DNA. *IEEE Engineering in Medicine and Biology*, **25** (1)(January 2006): 62-68.

[16] C. Shannon, A Mathematical Theory of Communication, *The Bell System Technical Journal*, **27**(1948) 379-423, 623-656.

[17] USDA / Agricultural Research Service, *'Junk' DNA Proves to be Highly Valuable*, Science Daily, viewed 10, March, 2010, http://www.sciencedaily.com/releases/2009/06/090606105203.htm

[18] J. van Lint, *Introduction to Coding Theory*, First Edition, New York, Springer-Verlag New York Inc., c1982.

[19] H. Yockey, *Information theory and molecular biology*. Cambridge University Press, Cambridge, 1992.

APPENDIX

Module 1: Exploring the ASCII Code

Introduction

This activity will show students how something they use everyday, a computer keyboard, relates to coding theory.

Long Term Objectives

To familiarize students with ASCII code, the binary alphabet and binary (base 2) notation.

Short Term Objectives

1. Students will correlate several of the keys on the computer keyboard with its numerical representation in ASCII code.
2. Students will learn to encode several of the keys in ASCII code.

Materials

Access to the internet to research ASCII code, paper, pencil, calculator

Activity

Instructor:

A. Introduce the following terms:

1. ASCII code (American National Standard Code for Information Interchange) - a binary code used for coding the 256 characters customarily found on a computer keypad. These characters are correlated to the numbers 0 through 255.

2. Binary alphabet - $\mathbf{B} = \{0, 1\}$.

3. Binary (base 2) notation - The use of the binary alphabet to represent values; how computers represent numbers and perform arithmetic.

B. Give the following example:

The letter S on the keyboard is assigned the number 83 (decimal) which is 01010011 (binary). To encode S (or 83) consider the powers base 2.

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 = 16$

$2^5 = 32$

As you can see, for each term, n_k, in binary notation you can determine the next term by the equation $n_{k+1} = 2(n_k)$. A quick way to generate this set of powers base 2 is to use the table feature on a graphing calculator. In the equation window, type Y1 = 2∧X, then go to the table window. It will show the above list for all X $\in \mathbb{Z}$

Since $256 = 2^8$, each character in the ASCII code will be represented by a sequence of eight digits. Using the values of the powers base 2, one can determine that 83 =

$64 + 16 + 2 + 1$. So in binary notation, $83 = 0(2^7) + 1(2^6) + 0(2^5) + 1(2^4) + 0(2^3) + 0(2^2) + 1(2^1) + 1(2^0)$ or 01010011.

Student:

A. Using the internet, find a table listing the correlation of the 256 characters of ASCII code to the corresponding key on the computer keyboard. Determine what decimal number represents the following characters:

1. $\sim$
2. Q
3. backspace
4. \
5. space bar

B. Now, determine the binary notation for the same characters.

Assessment

Take up and grade part B of the student activity. The answers should be as follows:

Character	decimal	binary
1. $\sim$	126	01111110
2. Q	81	01010001
3. backspace	8	00001000
4. \	47	00101111
5. space bar	32	00100000

Bonus: (Critical Thinking)

When you looked up the ASCII tables you may have noticed that the tables are divided. The most common table has the characters for decimals 0 - 127 and then the next table has the characters for decimals 128 - 255. Give a possible explanation for this division.

Module 2: How Computers Use Parity Check Digits in ASCII Code

Introduction

This activity will show students how a computer uses a parity check digit to determine if there is an error in the ASCII character retrieved.

Long Term Objectives

To familiarize students with the ideas of parity check digits and error detecting codes.

Short Term Objectives

1. Students will learn how to encode a message using a parity check digit.
2. Students will learn to use a generator matrix to encode the characters in ASCII code.
3. Students will learn to use a parity check matric to check for errors in ASCII code.

Materials

Paper and pencil.

Activity

Instructor:

A. Introduce the lesson:

The main purpose of coding theory is to detect and correct errors in received messages. ASCII code uses a parity check digit. A parity check digit is a single 1 or 0 added at the end of the code to make the number of 1s in the code even. When a computer receives an encoded ASCII character, if the number of 1s is not even, the computer knows an error has occured. This is called an error detecting code. Error correcting codes will be discussed later.

B. Give the following example:

Recall the ASCII code for the letter S was 01010011. It has 4 1s, which is even, so to encode it you would add a 0 to the end making the encoded S = 010100110.

For a single parity digit code it is easy enough to do by hand, even if you were doing all 256 characters. But imagine you were doing thousands of characters or adding many parity check digits to the message. You would want to have a better way of encoding them.

Coding theory uses a generator matrix to generate a code. For ASCII code the generator matrix G would be:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{17}$$

To encde a word such as S = 01010011, multiply it by G using matrix multiplication.

$$S*G = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \tag{18}$$

which is the same result we obtained before.

C. Give the following example:

To check an encoded message, a parity check matrix can be used. The parity check matrix, H, for the ASCII code is

$$H = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \tag{19}$$

To check the received message s = [0 1 0 1 0 0 1 1 0], again use matrix multiplication. If you get zero, you know the received message is correct.

$$s * H = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \end{pmatrix} \tag{20}$$

Now let's see what happens when the message is not correct. Let's change s to $s' =$ [1 1 0 1 0 0 1 1 0] and check again.

$$s' * H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \end{pmatrix} \tag{21}$$

As you can see, you did not get a zero so you know an error occured. But you cannot determine where the error occured, so ASCII is only an error detecting code; not an error correcting code.

Student:

A. Determine the ASCII parity check code for the characters you determined in Module 1:

1. $\sim$
2. Q

3. backspace

4. \

5. space bar

B. Use the generator matrix, G, and the parity check matrix, H, discussed in class to verify the work you did in A.

Assessment

Quiz:

A. Encode the following ASCII codes using the generator matrix G.

1. 00010110

2. 00110011

B. Determine if the follwing words are encoded ASCII characters using the parity check matrix, H.

1. 01001011

2. 01110001

VITA

SUZANNE MCLEAN LYLE

Education:	B.S. Mathematics, Western Carolina University, Cullowhee, North Carolina 1988 M.S. Mathematics, East Tennessee State University Johnson City, Tennessee 2010
Professional Experience:	Adjunct Instructor (Mathematics), Northeast State Community College Blountville, Tennessee, 2005 - present Adjunct Instructor (Mathematics - RODP), Columbia State Community College Columbia, Tennessee, 2008 - present