Classification of *Escherichia coli* K-12 Ribosome Binding Sites

An Error-Control Coding Model

BY ELEBEOBA E. MAY, MLADEN A. VOUK, AND DONALD L. BITZER

Advances in genomic sequencing have provided large amounts of data and have spurred computational tools for recognition and modeling of protein coding regions and accurate identification of exact translation start sites [1]–[10]. For example, probabilistic methods, such as Suzek et al.'s RBSFinder and Yada et al.'s GeneHacker Plus [11] return the location of the initiation codon for prokaryotic genes. Besemer and Borodovsky's GeneMarkS uses iterative hidden Markov models (HMMs) to locate translation start sites with relatively high accuracy [12]. Walker, et al. couple statistical methods with comparative genomics methods to identify start sites. Hannenhalli et al. incorporate several biological factors into their quantitative description of translational start sites, including the binding energy at the ribosome binding site (RBS), distance between RBS and initiator, and the initiator codon. They use a mixed integer linear program to determine parameters for their discriminatory model. GeneLook identifies protein-coding sequences using a two-stage, ab initio process [10]. Classification is based on structural characteristics of the sequence such as properties of the ribosome binding site (RBS), operon structure, and codon and nucleotide frequency. Other computational techniques including support vector machine, machine learning, combinatorial approaches, free energy calculations, Bayesian methods, and information theory have also been used in quantifying and classifying translational start sites [7], [13]–[15], [8], [16].

Though current computational methods have provided tools for locating start sites and increased the overall accuracy of gene locater systems such as GLIMMER and GeneMark, they usually require larger sequence windows for classification of initiation sites. Several initiation site classification tools such as RBSfinder use prior gene classification knowledge to aid in start site identification, hence functioning more like a post-processor. The ribosome, the protein translation machine, makes initiation decisions based on "real-time" processing of a single messenger RNA leader region. To construct a classification system that can make sufficiently correct real-time classification decisions and use a relatively small classification window that is relatively independent of other environmental factors, we propose an approach based on information theory. Drawing on parallels between genetic information processing in living organisms and the processing of communications data, we develop an error-control coding-based translation initiation classification system that uses an eleven base classification window.

In the sections that follow, we begin with an overview of channel codes and a summary of the translation initiation process. We draw parallels between the two and briefly review a channel code model for translation initiation. We present our block-code Bayesian classifier and discuss the results of applying our system to the translation start site location problem for *Escherichia coli* K-12.

Channel Codes and Translation Initiation

Overview of Channel Codes

In data communication, the need for coding theory and its techniques stems from the need for error control mechanisms. In an engineering communication system, a k-symbol block (bits for a binary alphabet) of digitized information is encoded by a (n, k) encoder that combines the input symbols with $(n\text{-}k)$ additional symbols based on a deterministic algorithm. In the biological domain a "symbol" can be the designator for an amino acid or nucleic acid base. The algorithm produces an (n, k) code, and the encoder is referred to as the channel encoder or the error-control encoder. The set of all valid n-symbol sequences (each sequence is called a *codeword*) produced by the (n, k) code make up the codebook [17], [18]. There are Q^k codewords for a Q-ary code. (In the context of genetics, the term *code* usually refers to the mapping of symbols used to identify nucleic acid bases to symbols used to identify amino acids that form proteins. In the information theory domain, a code is the result of algorithmic manipulation of basic symbols used to describe information. The purpose of this type of code is to provide robustness in data communication processes. We believe that the latter principle may in fact be used in the analysis of genetic sequences.)

The encoded information is transmitted through a potentially noisy channel where the transmitted bits can be corrupted in a random fashion. At the receiving end, the received message is decoded by a complementary channel decoder [17], [18]. The decoding process involves the removal and possibly correction of errors introduced during

 0739-5175/06/$20.00©2006IEEE

transmission and removal of the n-k excess symbols in order to recover the original k-symbols of transmitted information. The decoding mechanism can only cope with errors that do not exceed the code's error-correction capability. Figure 1 shows an example of a (3, 1) binary repetition code that uses a majority-logic decoding algorithm.

Channel codes can be broadly described as pattern recognition systems [19]. The codewords produced by the code are patterns the system wants to recognize. A "good" code will separate valid patterns in such a way that they can be recognized and will reject all other patterns. This work defines a "good" code based on how well the code recognizes the "patterns" or RNA bases that form the leader region, that is, the RNA bases upstream of (preceding) the location of translation initiation site (e.g., AUG).

Translation Initiation in Prokaryotes

There are three main steps involved in converting information contained in DNA sequences into functioning polypeptide chains: replication, transcription, and translation [20]. During replication, DNA doubles, forming an identical copy of itself. In transcription, information contained in DNA is converted to its RNA equivalent. The result, for gene-specifying DNA, is a messenger RNA (mRNA). In the final process, translation, the ribosome (a compact macromolecule made up of two subunits; in prokaryotes this is the 30S and 50S subunits) locates a valid start site (initiation process) and converts the mRNA sequence to a sequence of amino acids, which specifies a protein (elongation process). Each three-base mRNA sequence (a codon) corresponds to an amino acid.

Initiation, the longest phase in translation, involves two principal steps. First, the 30S ribosomal subunit combines with a stabilizing protein, initiation factor three (IF3). The 30S/IF3 molecule then recognizes the ribosome binding site of the bacterial mRNA. A special hexamer called the Shine-Dalgarno [21] sequence is contained in the ribosome binding site. Once the 30S/IF3 molecule attaches to the mRNA, IF3 is relased, leaving the mRNA/30S complex, which is called the initiation complex. The 30S associates with the mRNA sequence by forming hydrogen bonds between the 16S ribosomal RNA (rRNA) in the 30S subunit and the bases of the mRNA (Figure 2).

We now draw a parallel between the information communication processes and the genetic processes [22]. Assume that the unreplicated DNA is the output of a concatenated encoding process, the genetic replication process then represents the error-introducing channel. The genetic decoding process is then: transcription, translation initiation, and translation elongation plus termination [23] ,[24], [22]. We can now view the ribosome's interaction with the mRNA similar to the interaction of a channel decoder with an error-control encoded received data stream. Ideally, to determine the decoding model for the ribosome we would simply invert the encoding model that produced the DNA. We have yet to satisfactorily identify the genetic process that parallels the error-control encoding process. Instead we analyze key elements involved in initiating protein translation and constructed a plausible

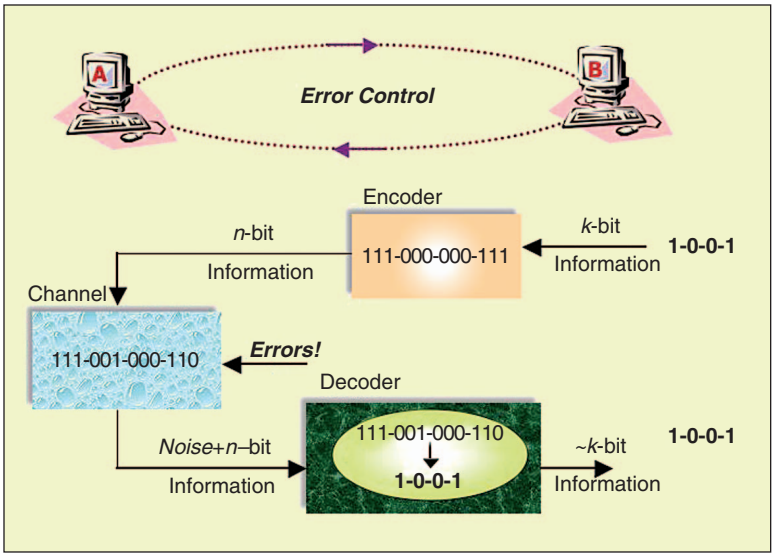

Fig. 1. An example (n=3, k=1) binary repetition error-control coding system. The encoder combines the one information bit with two additional bits, which are simple repetitions of the information bit. As the encoded bit stream is transmitted through the channel, some bits can become corrupted. The decoder uses simple majority logic to determine the original transmitted message. In majority logic decoding, we estimate our original information bit to be a 0 if the majority of the received bits are zeros and a one otherwise.

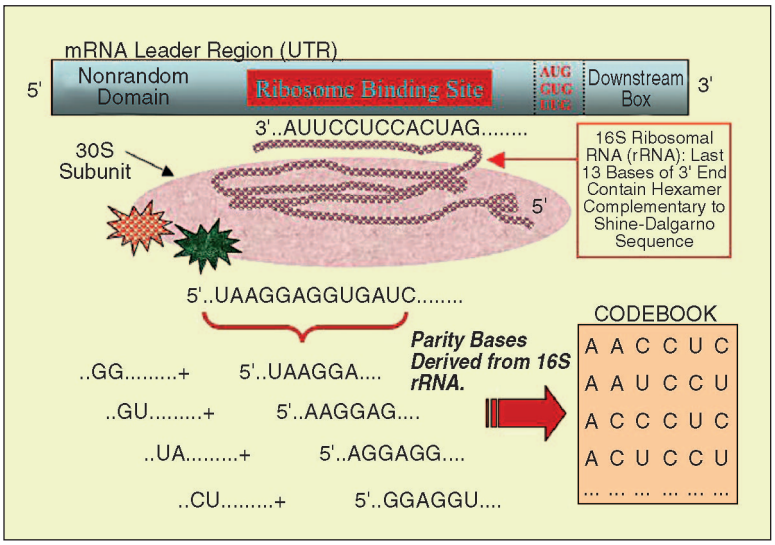

Fig. 2. An illustration of the translation initiation process. Genetic information is processed 5′ (five prime) to 3′ (three prime). The region before (upstream) the initiation codon (designated by AUG/GUG/UUG) is known as the 5′ untranslated leader region (UTR). The region following (downstream) the initiation codon is the protein coding portion of the mRNA. The interaction between the 16S rRNA and the leader region of the mRNA is key in successful initiation. Therefore we use the 3′ end of the 16S rRNA as a template for forming the codewords for our block coding model.

encoding and corresponding decoding scheme for describing the initiation process in prokaryotic organisms. The key biological elements considered in forming the coding model are: the 3′ (three prime) end of the 16S ribosomal RNA, the common features of bacterial ribosomal binding sites (such as the existence and location of the Shine-Dalgarno sequence), and RNA/DNA base-pairing principles. An encoding method was assumed and the corresponding decoding algorithm was developed using the 16S ribosomal RNA [25].

Block Code Model of Translation Initiation

In the information theory block code model, a genetic encoding is modeled as an (n, k) block code whose output is a systematic, zero parity check code [17], [25]. A systematic zero parity code contains the k information symbols at the beginning of the codeword, followed by n-k parity symbols selected such that the "sum" of the codeword symbols is zero. In the binary space (zeros and ones) the "sum" operator is the exclusiveOR operation, in higher-order field mathematics, special operator tables need to be constructed. Codewords of length $n = 5$ are constructed using the last 13 bases of the 3′ end of 16S ribosomal RNA, which contains the hexamer complementary to the Shine-Dalgarno sequence [20]. We use minimum Hamming distance decoding to test the block code model of translation initiation. (The Hamming distance between two sequences is the number of positions where they differ when aligned. See [25] for a detailed description of the model.)

Sequence data from the *E. coli* K-12 [26] strain MG1655 genome (downloaded from the National Institutes of Health ftp site: www.ncbi.nlm.nih.gov) is used to construct and test the model. Figure 3 shows the resulting mean minimum Hamming distance values by position for the (5, 2) block code model. The horizontal axis is the position of the RNA base relative to the first base of the start (initiation) codon. The vertical axis shows the ensample mean of the minimum Hamming distance values aligned for each of the three sequence groups (translated sequences, hypothetical translated sequences, and nontranslated sequences; categorized based on GenBank annotations). In general, the smaller the value on the vertical axis, the stronger the hydrogen 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.

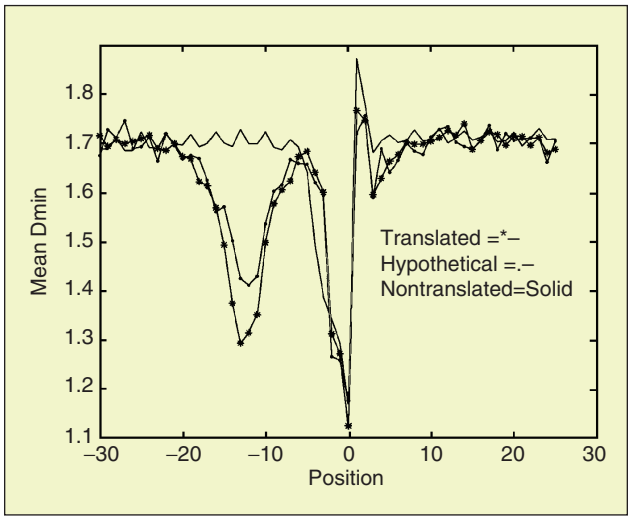

Fig. 3. Results of minimum distance block decoding model for the (5, 2) code.

As Figure 3 illustrates, there is a significant difference among the translated, hypothetical, and the nontranslated sequence groups. For the translated and hypothetically translated sequence groups, a minimum distance trough occurs in the –15 to –10 regions. These key regions contain the nonrandom domain and the Shine-Dalgarno domain [27]. All the sequence groups in the (5, 2) block code model achieve a global minimum mean distance value in the –5 to 0 region. This is most likely a result of their shared initiation codons.

Block-Code-Based Bayesian Classifier

Using the results of the block code model, we designed four Bayesian classification systems. The systems classify individual mRNA sequences as translation initiation sites or noninitiation sites based on the average minimum Hamming distance values in the -15 to -11 alignment window (this includes mRNA bases from position -15 to -7). The -15 to -11 window appears to provide the greatest distinction between the mean minimum Hamming distance values of leader (contains valid initiation site) and nonleader (contains invalid initiation site) sequences in *E. coli* K-12 (Figure 3).

The components of a Bayesian classifier are s, a measurable classification variable; $P(s/w_i)$, the conditional probability of measuring a value of s given classification class w; $P(w_i)$, the probability of the occurrence of each classification class.

The discrimination function for the classifier is

$$P(w_i\,|s) = \frac{P(s\,|w_i) * P(w_i)}{P(s)}, \tag{1}$$

where i designates the classification classes and

$$i = (\text{Translated, Nontranslated}) \tag{2}$$

and

$$P(s) = \sum_{i=1}^{N_{\text{class}}} P(s\,|\,w_i)^* P(w_i). \tag{3}$$

The constant N_{class} is the number of classification classes; N_{class} is two for the current work. Since $P(s)$ is the same for all classification classes, we can (for the purpose of classification) simplify the discrimination function, (1), to

$$P(w_i\,|\,s) \approx P(s\,|\,w_i)^* P(w_i). \tag{4}$$

Measurable Classification Variable

In the discrimination function, the value of the classification variable s is derived from the sum of the positional Hamming distance values within the -15 to -11 alignment window:

$$s = \sum_{p=-15}^{-11} D_{\text{avg}_p}, \tag{5}$$

where

$$D_{\text{avg}_p} = \frac{1}{N}\sum_{j=1}^{N} d_{\text{min}_{p,j}}. \tag{6}$$

The positional Hamming distance value D_{avg_p} is the average of the N lowest Hamming distance values $d_{\min_{p,j}}$ for the subsequence that begins at position p. In this work, N is set to a value that equals 10% of the total number of codewords in the codebook. Below are the steps for calculating s for a given mRNA subsequence, $R_{-15..-7}$ (alignment positions are from -15 to -11).

1) Set s to zero and position counter, p, to position -15.
2) For position p, find the Hamming distance between subsequence $R_{p..(p+n-1)}$ and all codewords in the codebook [25].
3) Find D_{avg_p} using the Hamming distance values calculated for subsequence $R_{p..(p+n-1)}$.
4) Increment s: $s = s + D_{\text{avg}_p}$.
5) Increment p: $p = p + 1$.
6) If $p > -11$ then exit, else goto Step 2.

To illustrate, let the $(n = 5,\ k = 2)$ block code model produce the following set of ten valid codewords (a codebook):

AAGGU AAGUG AAAUC GCAGG GCGGA
GCGAG CUGUG CUAUC UGAAG UCGGU

for

$$N = 0.20^{*}10 = 2, \tag{7}$$

and as a simple example, assume the mRNA subsequence (the received parity stream) from position -15 to position -7 is a repetitive sequence of cytosine bases:

$$R_{-15..-7} = \text{CCCCCCCCC}. \tag{8}$$

The s value for $R_{-15..-7}$ is calculated as follows:

- Set $s = 0$ and $p = -15$
- For $p = -15$,

$$R_{-15..(-15+5-1)} = R_{-15..-11} = \text{CCCCC}.$$

Table 1 lists the Hamming distance between subsequence $R_{-15..-11}$ and all codewords in the example codebook.

- The $N = 2$ lowest Hamming distance values from Table 1 are three (corresponding to codeword CUAUC) and four (corresponding to codeword AAAUC). Using (6), $D_{\text{avg}_{-15}}$ is:

$$D_{\text{avg}_{-15}} = \frac{1}{2}(4 + 3) = 3.5.$$

- Increment s:

Table 1. Hamming distance between example codeword set and *CCCCC*.

Codeword	$D_{Hamming}$	Codeword	$D_{Hamming}$
AAGGU	5	AAGUG	5
AAAUC	4	GCAGG	4
GCGGA	4	GCGAG	4
CUGUG	4	CUAUC	3
UGAAG	5	UCGGU	4

$$s = 0 + D_{\text{avg}_{-15}} = 3.5.$$

- Increment p:

$$p = -15 + 1 = -14.$$

- Continuing as illustrated in the previous steps, we find that for alignment positions $p = -15.. - 11$, $D_{\text{avg}_{-15..-11}}$ is

$$D_{\text{avg}_{-15..-11}} = (3.5 \quad 3.5 \quad 3.5 \quad 3.5 \quad 3.5),$$

and the value for the classification variable, s, is:

$$s = 3.5 + 3.5 + 3.5 + 3.5 + 3.5 = 17.5. \tag{9}$$

This process is used to calculate the s statistic for every mRNA subsequence in the training and test sets. We compiled our data set using GenBank sequences and annotations. All open reading frames (with AUG on the start codon) on the noncomplement strands that were not listed as a gene were categorized as nontranslating genes. All genes on the noncomplement strand were categorized as translating genes. The data set (translating and nontranslating) was divided in half to form the training and testing data sets.

Defining the Statistical Model: $P(w_i \mid s)$

The distribution for s of the training set of *E. coli* leader and nonleader sequences are shown in Figure 4 and Figure 5, are formed. In the probability distribution function (PDF), Figure 2, and in the cumulative distribution function (CDF), Figure 3, the horizontal axes are the s values, and the vertical axes are the probability of the s value occurring for the translated (valid leader) and nontranslated (invalid leader) training set models. A Wilcoxon Rank-Sum test applied to the two training sets verified that their corresponding

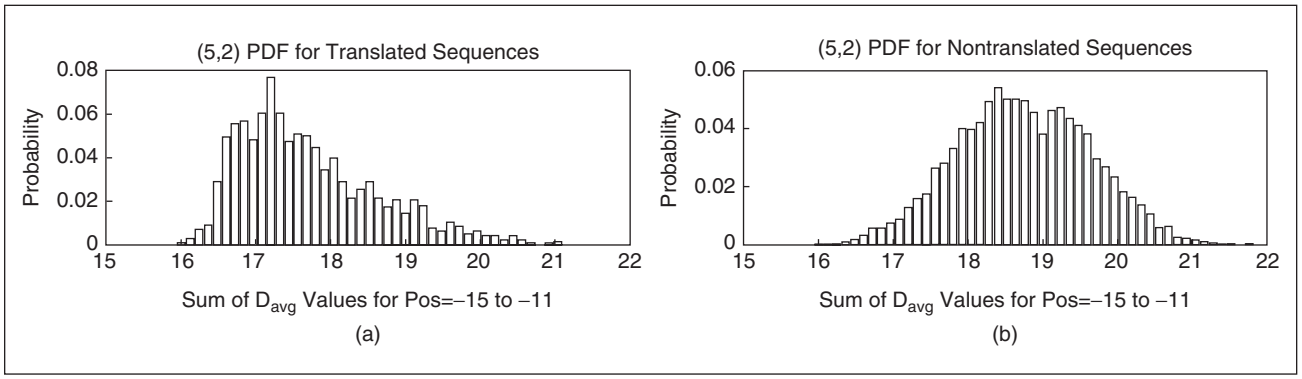

Fig. 4. Probability distribution of s values for the (5, 2) block code model.

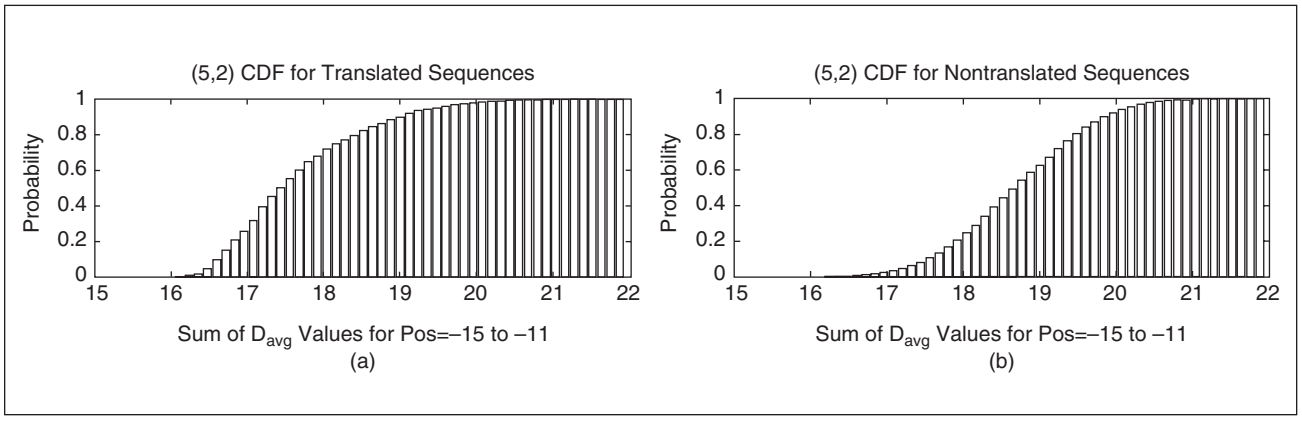

Fig. 5. Cumulative distribution of s values for the (5, 2) block code model.

probability distributions were nonidentical. Both the PDF and CDF are used to model the probability that s occurs, given that we are in class w_i.

The PDF model is the probability of a single s value occurring in a given classification group, $P(S = s \mid w_i)$; the CDF model is the probability of a range of s values occuring, $P(S \leq s \mid w_i)$.

Incorporating Prior Knowledge, $P(w_i)$

Two approaches for defining $P(w_i)$, the probability that class w_i occurs, are investigated. One approach defines prior probabilities by assessing the number of valid leader regions and the number of invalid leader regions in the *E. coli* genome. Valid leader regions are leaders that are associated with an annotated gene, as designated in the *E. coli* K-12 MG1655 genome [26] (available through GenBank). Invalid leaders are sequences upstream from an open reading frame that is not designated as a valid gene in GenBank. The open reading frames associated with invalid leaders must contain at least 33 codons. We estimate the prions by taking the ratio of valid leaders to the total number of potential leaders and the ratio of nonvalid leaders to the total number of potential leaders:

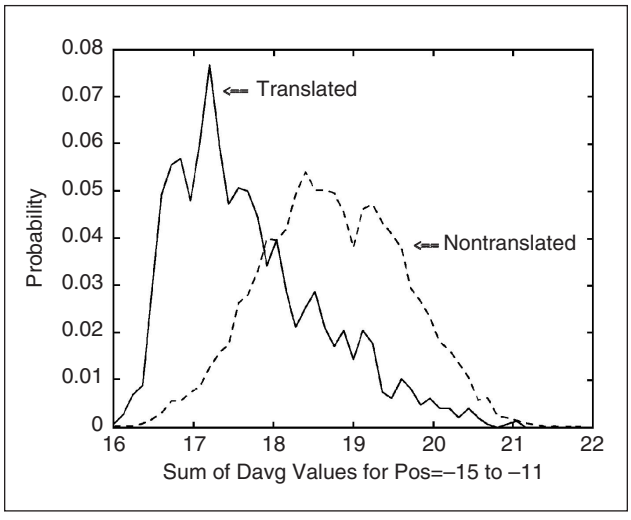

Fig. 6. Probability distributions used for the (5, 2) Bayesian classifiers.

$$P(w_{\text{Translated}}) = 9.39\%, \tag{10}$$

$$P(w_{\text{Nontranslated}}) = 90.61\%. \tag{11}$$

The second approach for defining $P(w_i)$ uses the coding theory framework on which our model is constructed. From a coding theory view, the decoder has no prior knowledge regarding the probability that a received parity sequence (the mRNA leader) being a valid or an invalid leader. Therefore, each class can be viewed as equally probable. This results in the following prior probability values:

$$P(w_{\text{Translated}}) = 50\%, \tag{12}$$

$$P(w_{\text{Nontranslated}}) = 50\%. \tag{13}$$

Both approaches are used in our classification system.

Bayesian Classification Systems

We form four discriminate functions using all combinations of equal and unequal priors.

- Classification System 1: Uses PDF and unequal prior probabilities.
- Classification System 2: Uses PDF and equal prior probabilities.
- Classification System 3: Uses CDF and unequal prior probabilities.
- Classification System 4: Uses CDF and equal prior probabilities.

Given the discrimination function, the rule for deciding to which class a received parity sequence (the mRNA test sequence) belongs to is:

IF $P(w_{\text{Translated}} \mid s) > P(w_{\text{Nontranslated}} \mid s)$ THEN

Select $w_i = w_{\text{Translated}}$

ELSEIF $P(w_{\text{Nontranslated}} \mid s) > P(w_{\text{Translated}} \mid s)$ THEN

Select $w_{\text{Nontranslated}}$

ELSE

Indicate a tie occurred.

To illustrate, using the statistic calculated in (9), we can classify our example mRNA "received" sequence, $R_{-15..-11}$. For $R_{-15..-11}$, the classification variable s equals 17.5. The discrimination functions for Classification System 1 and $P(w_{\text{Translated}} \mid s)$ and $P(w_{\text{Nontranslated}} \mid s)$ are calculated as follows:

- Discrimination function for translated class:

$$P(w_{\text{Translated}} \mid s = 17.5) = P(17.5 \mid w_{\text{Translated}})^* P(w_{\text{Translated}}) = 0.05^*0.0939 = 0.0047. \quad (14)$$

- Discrimination function for nontranslated class:

$$P(w_{\text{Nontranslated}} \mid s = 17.5) = P(s \mid w_{\text{Nontranslated}})^* P(w_{\text{Nontranslated}}) = 0.015^* 0.9061 = 0.0136. \quad (15)$$

Since the value of the discrimination function for the nontranslated class is greater than that of the translated class, the example sequence is classified as a nonvalid leader region.

Results

The four classification systems are implemented using the codebook generated from the (5, 2) block code model for *E. coli* K-12 genomic sequences [25]. Table 2 gives the number of sequences in the training and testing data groups.

The training set is used to construct the statistical models for the four classification systems and the classifiers are applied to the sequences in the testing set. Table 3 shows the classification results for the four Bayesian classifiers.

The true positive and false positive rates for each classifier in Table 3 are calculated as follows [27]:

$$\text{True Positive} = \frac{\#\text{ of leaders correctly classified}}{\text{total }\#\text{ of leader sequences}} \quad (16)$$

$$\text{False Positive} = \frac{\#\text{ of nonleaders in correctly classified}}{\text{total }\#\text{ of nonleader sequences}} \quad (17)$$

True negative and false negative rates are calculated in a similar manner to (16) and (17), respectively.

Of the four classification systems, Classifier 2 seems to perform the best, classifying leader and nonleader sequences equally well while maintaining a relatively low rate of incorrect classifications. Table 4 shows the correct versus incorrect classification rates for all four classification systems.

The incorrect classification is calculated as follows [29]:

$$\%\text{Incorrect} = 1.0 - \frac{\text{TruePos} + \text{TrueNeg}}{\text{TruePos} + \text{FalsePos} + \text{FalseNeg} + \text{TrueNeg}}. \quad (18)$$

The correct classification value flows naturally from (18).

Table 2. Size of training set and test set.

	Number in Training Set	Number in Test Set
Leader	1,459	1,458
Non-eader	10,520	10,519

Table 3. Results of Bayesian Classifiers for (5, 2) Block Code (values are in %).

	True Positive	False Positive	False Negative	True Negative
Classifier 1	19.91	1.78	80.09	98.22
Classifier 2	67.90	20.28	32.10	79.72
Classifier 3	26.73	2.52	73.27	97.48
Classifier 4	100	100	0.00	0.00

Table 4. Classification rates (in %) for (5, 2) Bayesian classification systems.

	Correct Classification	Incorrect Classification
Classifier 1	59.065	40.935
Classifier 2	73.81	26.19
Classifier 3	62.105	37.895
Classifier 4	50	50

Discussion

Classifiers 1 and 3 have correct classification percentages above 50%, as shown in Table 4. This is elevated rates of accuracy are a result of very high specificity or true negative values. Classifiers 1 and 3 are able to effectively detect "errors" (sequences that are not part of the codebook set). Their high error detection rate (ability to accurately classify nonleader sequences) is heavily biased by the use of unequal prior probabilities. Since the prior probability for nontranslated sequences is large, only sequences with very few errors or deviations from the coding model can be detected. From a coding theory perspective, the classifiers detect errors extremely well. From an engineering perspective, a decoding system like Classifiers 1 and 3 wastes resources because it causes the transmitter to resend information multiple times. Multiple retransmission would be necessary since the decoding system fails to recognize slightly errored transmissions. The biological parallel to Classifiers 1 and 3 is a system where only "perfect" sequences are recognized as translation initiation sites. Such a system may not be evolutionally viable.

Classifier 4 has the inverse problem. It fails from a coding theory standpoint. Classifier 4 fails to detect any errors, since it classifies all received sequences as valid. The biological system represented by Classifier 4, which indiscriminately initiates translation at all potential initiation sites, would exist below Eigen's error threshold for viable mutants [24]. Such a system would also be evolutionally inviable.

As an error-control decoding system, Classifier 2 outperforms the other classification systems. It detects received sequences with slight variations from the codeword set

67.9% of the time and detects nonsystem sequences at an even higher rate. For a system where a false positive classification (interpreting an invalid sequence as valid) is costly or detrimental to the system, Classifier 2 is not as desirable as Classifiers 1 and 3. Classifier 2 has a false positive rate of 20.28% while Classifiers 1 and 3 have false positive rates of 1.78% and 2.52%, respectively. For some communication systems, it is better to retransmit than to decode the information incorrectly. When compared to the other three classification systems, Classifier 2 represents the most biologically feasible system. It is able to detect a varied set of correct initiation sites, while rejecting sequences with errors beyond the error threshold. The biological equivalent of Classifier 2 has the greatest prospect for evolutionary viability.

To improve the classification systems that use unequal prior probabilities, the block code model would have to produce codewords that have a greater minimum Hamming distance separation than the present code. Figure 4 shows the translated and nontranslated PDFs for the current (5, 2) block code. Reducing the region of overlap between the coding-based probability distribution of leader and nonleader sequence sets would increase the sensitivity of the classifier and reduce incorrect classification rates. To accomplish this, a more powerful error-control code must be designed. Such a code would contain codewords with larger Hamming distances between the sequences in the codebook set, thus increasing the minimum distance of the code. The larger the minimum distance of a code, the more errors it can detect and correct.

Conclusion

The classification system presented uses an eleven base classification window to identify translation initiation sites. This is a relatively small decision window compared to other classification methods. The 74% correct classification rate of System 2 appears to be comparable to that of GeneMarkS (tested on a set of 195 experimentally validated *E. coli* genes [12]) after its intermediary Step 2 (67% accuracy following an initial coding region identification step); but GeneMarkS exceeds System 2 after intermediary Step 4 (85% following prediction using GeneMark.hmm). Upon completing all model iterations, the accuracy of GeneMarkS increases to almost 95%. We use genomic data from GenBank to test our classification system, which differs from the dataset used by GeneMarkS, Glimmer (71% correct classification rate), and ORPHEUS (76% correct classification rate). Their classification rates also reflect correct classification of both the 5′ and 3′ ends of the gene, which corresponds to the transcribed mRNA. A better comparison system for our work is Nishi et al.'s GeneLook system [10]. Using annotated gene sequences, GeneLook was able to accurately identify 76% of selected *E. coli* genes. Our 74% accurate classification rate is comparable to Nishi et al.'s.

Our classification system reflects the genetic initiation process in several aspects. In practice, the small ribosomal subunit does not "analyze" the entire open reading frame before determining whether a three-base nucleic acid sequence is an initiation site. Likewise, the classification systems presented use a relatively small window to detect potential ribosome binding sites. Similar to the biological model, the error-control-based classifiers use the redundancy, or extra information, present in the mRNA leader sequence to locate valid translation initiation sites.

The results thus far are encouraging. They suggest that it is highly possible to implement an error-control coding-based scoring system that can be combined with Bayesian classification for detecting and possibly designing prokaryotic translation initiation sites. Elucidating how genetic systems incorporate and use redundancy, which is at the core of information-based error correction, and understanding the functional significance of genetic errors from a coding theory perspective will help provide insight into the fundamental rules that govern genetic regulatory systems.

Acknowledgments

The authors would like to acknowledge the contributions of Dr. David I. Rosnick to this work and thank Dr. Anne-Marie Stomp for her extensive editorial comments. We also thank Dr. Jeffrey Thorne and Dr. Winser Alexander for their input into this work. Elebeoba May performed this work while at North Carolina State University. This work was supported in part by a National Science Foundation Minority Graduate Research Fellowship (Grant DGE-9616159) and a Ford Foundation Dissertation Fellowship.

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.

Elebeoba E. May received her Ph.D. in computer engineering from North Carolina State University and joined Sandia National Laboratories' computational biology department in May 2002. Her research interests include the use and application of information theory, coding theory, and signal processing to the analysis of genetic regulatory mechanisms, the design and development of intelligent biosensors, and large-scale simulation and analysis of biological pathways and systems. She has served as an associate editor and reviewer for *IEEE Transactions on Information Technology in Biomedicine*, on the IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS) 2004 organizing committee, and as chair and organizer of a special session on Communication and Coding Theory in Biology for the IEEE Engineering in Medicine and Biology Society (EMBS) 2003 annual meeting. Organizational memberships include: the IEEE EMBS, the IEEE Signal Processing Society, and the IEEE Information Theory Society. She is a recipient of the 2003 Women of Color Research Sciences and Technology Award for Outstanding Young Scientist or Engineer.

Mladen A. Vouk received a Ph.D. from King's College, University of London, the United Kingdom. He is the interim department head and professor of computer science and the associate vice provost for information technology at North Carolina State University, Raleigh. Vouk has extensive experience in both commercial software production and academic computing. He is the author/coauthor of over 180 publications. His research and development interests include software engineering, scientific computing (including application of engineering methods to

genetics, bioinformatics, and biophysics) information technology, assisted education, and high-performance networks. He is a member, former chairman, and former secretary of the IFIP Working Group 2.5 on Numerical Software, and a recipient of the IFIP Silver Core award. He is an IEEE Fellow, and a Member of IEEE Reliability, Communications, Computer, and Education Societies, and of the IEEE Technical Committee on Software Engineering. He is a member of ACM, ASQ, and Sigma Xi. He is an associate editor of *IEEE Transactions on Reliability*, a member of the editorial board for the *Journal of Computing and Information Technology*, and a member of the editorial board for the *Journal of Parallel and Distributed Computing Practices*.

Donald L. Bitzer received his Ph.D. in electrical engineering from the University of Illinois in 1960. He was a professor of electrical and computer engineering at the University of Illinois from 1960–1989. He retired from the University of Illinois to become a distinguished university research professor in the Computer Science Department at North Carolina State University. Bitzer's work has involved applying signal processing and coding theory to a variety of areas from radar signals and speech processing to the development of software and hardware required for large computer networks. The large educational computer systems PLATO and NovaNet are results of this research. His research led to the intelligent modems for telephone lines and cable systems as well as the flat plasma display panel now being used for television. More recently, his research has been directed toward using signal processing and coding theory to look for genomic information that controls the translation process in protein production. He has been granted numerous patents in the computer and electronic areas.

Dr. Bitzer has been a member of the National Academy of Engineering since 1974. He is a member of the American Society for Engineering Education (since 1974), a fellow in the American Association for Advancement of Science (since 1983), a Fellow of the IEEE (since 1976), a fellow in the Association for Development of Computer Based Instructional Systems (since 1986), and a fellow of the International Engineering Consortium (since 1984). He has received numerous awards. In 1967, he received the Industrial Research 100 Award, and in 1973, he received the prestigious Vladimir K. Zworkin Award of the National Academy of Engineering for "outstanding achievement in the field of electronics applied in the service of mankind." Other awards include the Chester F. Carlson Award (1981) from the American Society for Engineering Education for "Innovation in Engineering Education," the Computer Science Man of the Year (1975) from the Data Processing Management Association, and the Education Award (1989) from the American Federation of Information Processing Societies. In 1982 he was named Laureate of the Lincoln Academy by the State of Illinois for contributions made "for the betterment of human endeavor." In 2002 he received the National Academy of Television Arts and Sciences Emmy Award for Scientific Development and Technical Achievement for his invention and development of plasma displays. The College of Engineering at the University of Illinois awarded him with the Alumni Distinguished Service Award in 2004.

Address for Correspondence: Elebeoba E. May, Sandia National Laboratories, P.O. Box 5800 MS 0310 Albuquerque, NM 87185 USA. Phone: +1 505 844 9933. Fax: +1 505 844 5670. E-mail: eemay@sandia.gov.

References

[1] T.D. Schneider, G.D. Stormo, L. Gold, and A. Dhrenfeucht, "Information content of binding sites on nucelotide sequences," *J. Mol. Biol.*, vol. 188 pp. 415–431, 1986.
[2] D. Frishman, A. Mironov, and M. Gelfand, "Starts of bacterial genes: Estimating the reliability of computer predictions," *Gene*, vol. 234, no. 2, pp. 257–265, 1999.
[3] M. Tompa, "An exact method for finding short motifs in sequences, with application to the ribosome binding site problem," in *Proc. ISMB,* 1999.
[4] S.S. Hannenhalli, W.S. Hayes, A.G. Hatzigeorgiou, and J.W. Fickett. "Bacterial start site prediction," *Nucleic Acids Res.*, vol. 27, no. 17, pp. 3577–3582, 1999.
[5] B.E. Suzek, M.D. Ermolaeva, M. Schreiber, and S.L. Salzberg, "A probabilistic method for identifying start codons in bacterial genomes," *Bioinformatics*, vol. 17, no. 12, pp. 1123–1130, 2001.
[6] M. Walker, V. Pavlovic, and S. Kasif, "A comparative genomic method for computational identification of prokaryotic translation initiation sites," *Nucleic Acids Res.*, vol. 30, no. 14, pp. 3181–3191, 2002.
[7] A. Zien, G. Ratsch, S. Mika, B. Scholkopf, T. Lengauer, and K.R. Muller, "Engineering support vector machine kernels that recognize translation initiation sites," *Bioinformatics*, vol. 16, no. 9, pp. 799–807, 2000.
[8] E. Crowley, "A Bayesian method for finding regulatory segments in DNA," *Biopolymers*, vol. 58, pp. 165–174, 2001.
[9] H.-Y. Ou, F.-B. Guo, and C.-T. Zhang, "GS-Finder: A program to find bacterial gene start sites with a self-training method," *Int. J. Biochem. Cell Biol.*, vol. 36, pp. 535–544, 2004.
[10] T. Nishi, T. Ikemura, and S. Kanaya, "GeneLook: A novel ab initio gene identification system suitable for automated annotation of prokaryotic sequences," *Gene*, vol. 346, pp. 115–125, 2005.
[11] T. Yada, Y. Totoki, T. Takagi, and K. Nakai, "A novel bacterial gene-finding system with improved accuracy in locating start codons," *DNA Res.*, vol. 8, no. 3, pp. 97–106, 2001.
[12] J. Besemer, A. Lomsadze, and M. Borodovsky, "GeneMarkS: A self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions," *Nucleic Acids Res.*, vol. 29, no. 12, pp. 2607–2618, 2001.
[13] W.S. Hayes and M. Borodovsky, "How to interpret an anonymous bacterial genome: Machine learning approach to gene identification," *Genome Res.*, vol. 8, no. 11.
[14] A.G. Pedersen and H. Nielsen, "Neural network prediction of translation initiation sites in eukaryotes: Perspectives for EST and genome analysis," in *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 1997.
[15] Y. Osada, R. Saito, and M. Tomita. "Analysis of base-pairing potentials between 16S rRNA and 5′ UTR for translation initiation in various prokaryotes," *Bioinformatics*, vol. 15 pp. 578–581, 1999.
[16] T.D. Schneider, "Measuring molecular information," *J. Theor. Biol.*, vol. 201 pp. 87–92, 1999.
[17] P. Sweeney, *Error Control Coding an Introduction.* New York: Prentice Hall, 1991.
[18] S. Lin and D.J. Costello Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
[19] Y. G. Savchenko and A.A. Svistel'nik, "An approach to pattern recognition systems," *Engineer. Cybernetics*, vol. 2, pp. 144–146, 1968.
[20] B. Lewin, *Genes V*. New York: Oxford Univ. Press, 1995.
[21] J. Shine and L. Dalgarno, "The 3'-terminal sequence of Escherichia coli 16S ribosomal RNA: Complementarity to nonsense triplets and ribosome binding sites," *Proc. Nat. Acadamy Sci.,* vol. 71, no. 4, .pp. 1342–1346, Apr. 1974,.
[22] E.E. May, M.A. Vouk, D.L. Bitzer, and D.I. Rosnick, "An error-correcting code framework for genetic sequence analysis," *J. Franklin Instit.*, vol. 341, pp. 89–109, 2004.
[23] G. Battail. "Does information theory explain biological evolution?," *Europhysics Lett.*, vol. 40, no. 3 pp. 343–348, Nov. 1997.
[24] M. Eigen, "The origin of genetic information: Viruses as models," *Gene*, vol. 135, pp. 37–47, 1993.
[25] E. May, M. Vouk, D. Bitzer, and D. Rosnick, "Coding theory based models for protein translation initiation in prokaryotic organisms," *BioSyst.*, vol. 76, pp. 249–260, 2004.
[26] F.R. Blattner, Plunkett, G. III, C.A. Bloch, N.T. Perna, V. Burland, M. Riley, J. Collado-Vides, J.D. Glasner, C.K. Rode, G.F. Mayhew, J. Gregor, N.W. Davis, H.A. Kirkpatrick, M.A. Goeden, D.J. Rose, B. Mau, and Y. Shao, "The complete genome sequence of Escherichia coli K-12," *Sci.*, vol. 277 no. 5331 pp. 1453–1474, 1997.
[27] L. Gold and G. Stormo, "Translational initiation," in *Escherichia coli and Salmonella typhimurium*, *Cellular and Molecular Biology*, pp. 1302–1307, 1987.
[28] F. Provost and T. Fawcett, "Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions," in *Proc. 3ird Int. Conf. Knowledge Discovery Data Mining*, 1997.
[29] G.M. Weiss and F. Provost, "The effect of class distribution on classifier learning," Rutgers Univ., New Brunswick, NJ, Tech. Rep. ML-TR-43, 2001.