INVESTIGATION OF CODING STRUCTURE IN DNA

Gail L. Rosen[*] *and Jeffrey D. Moore*
gailr@ece.gatech.edu, dntthink@bellsouth.net

Center for Signal and Image Processing, Georgia Institute of Technology

ABSTRACT

We have all heard the term "cracking the genomic code", but is DNA a code in the information theoretic sense? The coined term "genetic code" maps nucleotide triplets (codons) to amino acids. However, this is in computer coding sense because a codon instruction is performed to output an amino acid sequence. In this paper, we examine methods to detect redundant coding structure in DNA. First, a finite field framework for a nucleotide symbolic sequence is presented then approaches to finding sequence structure associated with error correcting codes are examined. We compare a previously proposed parity-check vector search method to a novel subspace partitioning algorithm. The subspace partitioning algorithm is a general approach to finding any linear coding redundancy. Our method provides an easy way of visualizing coding potential in DNA sequences as shown from the test data.

1. INTRODUCTION

Since the introduction of the Watson-Crick model of DNA, scientists have been trying to make sense of the long sequence, millions long for simple organisms and billions long for complex ones, composed of four bases. Since the introduction of Shannon's mathematical theory of communication, many scientists have tried to explain DNA within an information theoretic framework [4]. Claude Shannon's PhD thesis [9] was a mathematical theory of genetics. It is appropriate that his information theories are now applicable to his original interests.

1.1. The Channel

Communication channel models have been paralleled to DNA processes. In one doctrine, the channel is assumed to be the amino acid translation from nucleotide triplets [4]. In May et. al, the channel is the actual replication process [8]. The latter is good for mutation modeling since transcription and copying of DNA is a noisy process. "Proof-reading" mechanisms are observed during DNA replication, and when the activity of these polymerase mechanisms are blocked, error rates increase from 10e-6 to 10e-3 [3]. We use a model similar to May's since errors occur directly on the DNA strand in replication while errors in the translation process can also occur in the formation of amino acids and proteins. In our framework, DNA is the medium in which genetic information is transmitted from generation to generation.

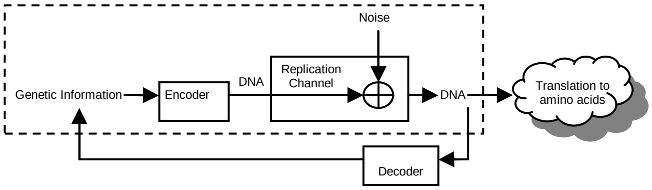


Figure 1: Our noisy channel model of genome replication with underlying coding assumption

1.2. Nucleotide Representation

Whenever one attempts to tie mathematical theory to the genome, the most important assumption is the representation of the nucleotides. There are several forms proposed and adapted to the type of analysis. Assessing the purine/pyrimidine structure, one can represent the purines(*A* and *G*) and the pyrimidines(*C* and *T*) with a binary representation. For four bases, one can choose a simple representation such as *A*=1, *C*=2, *G*=3, *T*=4 and use modulo operations, but this implies a structure on the nucleotides such that *T*>*A* and *C*>*G* [10]. For a model of the translation process, Anastassiou defined a complex representation: *A*= 1+j, *T*= 1-j, *C*= -1-j, *G*= -1+j [1]. The geometric interpretation of this representation still imposes constraints such that the Euclidean distance between *A* and *C* is greater than the distance between *A* and *T*, yet for the nucleotide quantization to amino acids, it was useful. Also, one can use indicator sequences (binary sequences representing the locations of each base in the nucleotide sequence) producing a four-dimensional representation yielding an efficient representation for spectral analysis [1]. When modeling processes in RNA, a fifth base, Inosine, can be taken into account [8].

In this paper, we will map nucleotides to a finite field of four, GF(4). This places on DNA the following Galois field properities: the elements are commutative under addition and both commutative and associative under multiplication as well as having an identity element and multiplicative inverses. Since GF(4) is an extension field of GF(2), we can create labels for the bases (Figure 2) using GF(2)'s primitive polynomial:

$$\alpha^2+\alpha+1=0$$

This abstraction of elements to integer labels makes finite field theory an attractive framework.

1.3. Problem Formulation

In Figure 1, we assume that the DNA is the sequenced genomic data available in GenBank [5], and our goal is to examine the dashed-line encompassed area and uncover the encoder scheme;

[*] This work was supported in part by a National Science Foundation Graduate Fellowship and an AT&T Research Laboratories Grant.

in other words, we wish to infer structure from the noisy output to retrieve the original genetic information. Also, if our assumption is correct and DNA is encoded in a linear redundant fashion, our analysis will uncover it. In this system, we know nothing about the encoder nor the original information, thus, system identification and deconvolution methods cannot be used. We will assume that the encoder is linear and try to characterize it given such output.

$$\alpha^0 = 1 \Leftrightarrow 1 \Leftrightarrow \mathrm{C}$$
$$\alpha^1 = \alpha \Leftrightarrow 2 \Leftrightarrow \mathrm{T}$$
$$\alpha^2 = \alpha + 1 \Leftrightarrow 3 \Leftrightarrow \mathrm{G}$$
$$0 = 0 \Leftrightarrow 0 \Leftrightarrow \mathrm{A}$$

Figure 2: Exponential root representation, polynomial representation, numerical label, and nucleotide label

1.4. (n,n-1) Code Search

Liebovitch et. al. introduced a single parity bit search [6]. A formal motivation for the approach can be derived from Theorem 4-8 from Wicker [11]: *"A vector **c** is a code word in* **C** *if and only if* $\boldsymbol{c}\mathbf{H}^{T}=0$*."* Thus, given a DNA frame vector $\boldsymbol{c} \in \mathbf{C}$, each row in **H**, the parity-check matrix, must span the nullspace of **C**, the full sequence. If $\boldsymbol{h}_n$ is a row vector of **H**, then $\boldsymbol{c} \bullet \boldsymbol{h}_n = 0$, ie: the parity-check matrix will be a set of orthogonal codes to $\boldsymbol{c}$. If there is one common $\boldsymbol{h}_n$ amongst all $\boldsymbol{c} \in \mathbf{C}$, then the DNA sequence is encoded in a single-bit parity-check fashion.

In the existence of a parity-check code, a sliding window will move frame by frame down a DNA sequence and will have one or more identical parity-check vectors orthogonal to each frame. See Figure 3 for an illustration of how the sequenced is windowed. If a nucleotide is inserted or deleted in the replication process, a frame shift error is introduced into the sequence and will cause the parity-check window to lose sync.

In order to exhaustively search for single bit parity-check codes with frame offset error possibilities, we decided to take the approach further and calculate the parity-check vector orthogonal to most frames including all frames offsets. Then each frame that contains this codeword is plotted and one can visualize from the graph if there is single bit parity-check vector common to each codeword over a region (See Figure 4a).

This approach reveals codewords that happen to be orthogonal to a parity-check vector by chance and should be compared against a sequence which has the same alphabet composition (see Section 2 for more on alphabet information content). A comparison of a random sequence is shown (Figure 4b). From the two plots, it is shown that an (n,n-1) code corresponding to a specific parity-check vector cannot be found. From simulations, we observed that the frequency of the most common parity-check vector directly relates to the entropy of the sequence (see Section 2 for explanation of sequence entropy).

This experiment provides context for the complexity of the problem. For this search, a type of code must be assumed. Thus, there is a need for a general approach such as Section 3 to discern an (n,k) coding structure from DNA sequence content.

2. GC CONTENT INTRODUCES REDUNDANCY

By investigating measures of entropy, we can look at basic measures of information content. The entropy of a sequence is maximized when all four nucleotides are equi-probable:

$$H = -\sum_i p_i \log_2(p_i) \qquad (1)$$
$$= -\sum_{i=1}^{4} \frac{1}{4} \log_2(\frac{1}{4}) = 2bits$$

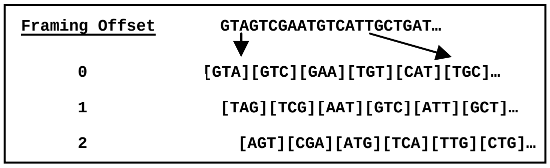
In many species, the bases are not equiprobable, but temperature dependent. Three bonds exist in *C* and *G* bases while only two exist in *A* and *T*. Thus, it takes more energy to make C and G, and it has been found that GC content is higher in warm-environment organisms than cold-environment. For example, Micrococcus Lysodeikticus[4] has the following base frequencies: Pr(C)=Pr(G)=.355 and Pr(A)=Pr(T)=.145. Therefore, the entropy for this organism utilizing the first part of (1), is 1.87 bits, which implies redundancy from this imbalance.

For our example data, a segment from the E. coli K-12 MG1655 coding region sequence has the following composition: *N(A)/N*=.262, *N(C)/N*=.281, *N(T)/N*=.206, *N(G)/N*=.25. Therefore, it is nearly at maximum entropy with 1.99 bits.

3. SUBSPACE PARTITIONING FOR (N,K) CODES

Our primary goal is to identify and characterize any linear constraints that might appear in regions of a sequence. Lacking the benefit or prior knowledge regarding the location, duration, or dimensionality of subspace partitioning in the sequence, we propose a method that generates a complete orthogonal basis set oriented to a local region of data. The basis set is used to decompose the sequence (equivalent to a coordinate transformation). The consistent presence of nulls in the transformed sequence indicates both the presence and the dimension of linear subspace partitioning in the data.

The first assumption is a fixed codeword length, *n*. The DNA elements are grouped into a matrix, $\mathbf{V} = [\boldsymbol{v}_1\ \boldsymbol{v}_2\ \cdots\ \boldsymbol{v}_N]$ where the length of the entire DNA sequence is $N * n$ and $\boldsymbol{v}_i$ is length *n*. The alignment of the frames relative to the starting point will be referred to as the framing offset. A choice of a particular framing offset will be referred to as the frameset, or open reading frames as called in the biological literature. Given the frame length *n*, there are *n* unique framesets.



```
Framing Offset      GTAGTCGAATGTCATTGCTGAT…

      0           [GTA][GTC][GAA][TGT][CAT][TGC]…
      1             [TAG][TCG][AAT][GTC][ATT][GCT]…
      2               [AGT][CGA][ATG][TCA][TTG][CTG]…
```

Figure 3: Illustration of vector framing for n=3.

We apply the Gram-Schmidt algorithm using finite field operations to the sequence of vectors to yield a complete set of orthogonal basis vectors, $\{\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots\boldsymbol{e}_n\}$. In the event that the entire sequence consists of vectors lying in a subspace of dimension less than *n*, we introduce random vectors and continue to iterate Gram-Schmidt until the basis set is complete. This yields a transform matrix **G** that is clearly full-rank, as it consists of *n* orthonormal vectors.

Once an orthogonal basis is formed from the first *j* frames of data, the $\boldsymbol{v}_i$'s for *i>j* are decomposed into components of each of the basis vectors. This is simply a coordinate transformation and can be described by:

$$\boldsymbol{t}_i = \mathbf{G}\,\boldsymbol{v}_i \qquad \textit{where} \quad \mathbf{G} = \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \vdots \\ \boldsymbol{e}_n \end{bmatrix} \tag{2}$$

Provided that the data has been framed correctly when applying the Gram-Schmidt algorithm, a linear coding redundancy can be detected by noting consistent null coordinates over a region in the transformed sequence of length-*n* vectors, $\{\boldsymbol{t}_1,\boldsymbol{t}_2,\ldots,\boldsymbol{t}_{N-j}\}$. This null detection would indicate a subspace of the actual n-dimensional space exists.

Knowing nothing of the dimension or alignment of the data, we must apply the algorithm for many codeword lengths. For a given codeword length and for a given locality in the sequence, we apply the algorithm *n* times to account for each framing offset. For each of the *n* iterations, the vector frameset is offset by one element of the sequence to guarantee that if length-*n* codewords are present in the sequence, one of the framesets will be properly aligned.

Algorithm Outline

1. *Obtain the orthonormal basis,* $\{\boldsymbol{e}_1,\boldsymbol{e}_2,\ldots,\boldsymbol{e}_n\}$*, by Gram-Schmidt orthogonalization of j number of* $\boldsymbol{v}_i$ *frames where* $j \geq n$*. Form the transform matrix,* $\mathbf{G}$*, from this set.*
2. *Decompose the sequence into its basis components,* $\{\boldsymbol{t}_1,\boldsymbol{t}_2,\ldots\boldsymbol{t}_{N-j}\}$*, across all possible framing offsets.*
3. *Note the persistence of nulls in* $\boldsymbol{t}_i$*'s. Calculate confidence by comparing against the probability of sequential sets of randomly chosen vectors having the same subspace partitioning.*

It should be noted that on finite fields, non-zero element vectors can have an inner product of zero (the additive identity element of the field), thus self-orthogonal vectors can exist. The situation sometimes arises in which a subspace is characterized entirely by self-orthogonal basis vectors. For this reason, the coordinates in the transformed vector sequence associated with these self-orthogonal basis vectors are always zero. In this case the decomposition cannot proceed and the algorithm must be terminated, reframed, and started anew.

Given the copious volume of data produced by iterating the algorithm over numerous frame shifts and codeword lengths, a visualization method is devised to aid in the search for consistent subspace partitioning. For each frameset, consistent nulls in the decomposed vectors are noted in an attempt to characterize the unoccupied subspace. A null-subspace indicator vector is used to mark the locations of nulls found consistently in the data. Each shift in sequence results in an update of the indicator vector. If the vector remains unchanged across iterations, a probabilistically-based value increases to indicate confidence in the presence of subspace partitioning (as the probability of randomly-chosen vectors possessing the observed subspace partitioning diminishes). We can then plot the confidence as a function of sequence index *i* across all possible framing offsets.

The algorithm is capable of detecting and characterizing linear subspace partitioning in any sequence provided that such structure is manifest in the data. For a given sequence, all such structure can be found provided that the algorithm is run for every possible framing offset and for every possible codeword length.

By way of illustration, a random test sequence is generated to occupy a five-dimensional subspace of an eight-dimensional vector space. This constitutes an (8,5) linear block code in GF(4). Running the algorithm on this sequence for *n*=8 yields the confidence image shown in Figure 5a.

Interstitial symbols are introduced throughout the sequence to illustrate the robustness of the algorithm to framing offsets. When framing offsets are introduced in the sequence, the region of high subspace partitioning confidence simply migrates to the corresponding row in the diagram.

These "confidence stripes" of themselves say nothing of the dimensional occupancy of the underlying sequence. Rather, they are used as search tools to simplify the analysis of large volumes of data. Their presence alerts us to the location of subspace partitioning in the sequence, at which point we can retrieve the local indicator vector to observe that, indeed, there are three dimensional nulls present throughout the duration of each of the confidence stripes.

4. RESULTS

The parity-check search method only searches for a common parity-check vector but could be expanded to a higher rank parity-check matrix, **H**. However, the subspace partitioning method is a more adaptable algorithm for general redundancy analysis.

The subspace partitioning algorithm described is able to identify and parameterize dimensional occupancy in a region independent of framing, provided that the structure is present from the outset. This algorithm can be more generally applied to any sequence for which it is suspected that coding properties are present. The algorithm could readily be adapted in a classification scheme for data of unknown origin or for cryptographic/cryptanalysis tasks in which the code or encryption scheme is unknown.

The algorithm needs improvement in two areas. Firstly, the algorithm uses nulls in a transform to indicate subspace partitioning. This requires that the coordinate system described by the transform be properly oriented. The transform matrix is guaranteed to be properly aligned for exactly one of the possible framesets, provided that the structure in question is present from the outset of the sequence. If there is an onset of structure in the data at a later point in the sequence, it may not be found. This stems from the primacy effect inherent to the Gram-Schmidt algorithm: the coordinate system (basis set) produced is oriented according to the order in which vectors are presented.

Secondly, the component decomposition algorithm is defeated by the case in which the Gram-Schmidt algorithm produces a fractional basis set. This is because finite field arithmetic allows for the existence of self-orthogonal vectors. The situation sometimes arises in which Gram-Schmidt produces a coordinate subspace whose complement contains entirely self-orthogonal vectors. While this situation is rare (7 out of 75 times in processing the E Coli DNA strand), it is impossible to perform the decomposition discussed here when it does occur. In this way, it creates "blind spots" for the algorithm: certain combinations of codeword length and framing offsets are self-orthogonal and cannot be analyzed using Gram-Schmidt.

A superior technique is presently being investigated to overcome both of these faults. The revised algorithm will use a more general solution to find all linear dependency among sets of vectors over a region. This is a "sliding window" solution that is insensitive to the starting point in the data but will yield the same information, the dimension of occupancy for a given codeword length. A method for sidestepping the problem of self-orthogonality is under investigation. Provided that a tractable solution exists, it guarantees the capability of detecting any and all subspace partitioning in a region of data.

5. CONCLUSIONS AND FUTURE WORK

From the investigations presented, preliminary results show no apparent error correcting code in E. Coli DNA. Other DNA sequences should be tested, and alternative forms of codes, such as convolutional codes, need to be considered. The following discusses biologically discovered regions in DNA and how they may affect this type of analysis.

Our methods were based on the hypothesis that there is an underlying coding structure in the DNA sequence used for mutation recovery in the replication process. We assumed this structure would occur in both “coding” (in the computer coding sense) and non-coding regions. (There has been great effort in distinguishing between these gene and “junk” regions [2].)

On the contrary, mutation rates vary from region to region in the genome, and these areas might need separate treatment. Nature relies on mutations and uses errors for diversity. It has been noted that non-coding regions (which compose over 97% of the entire genome) are more susceptible to mutation than coding regions. Also, frequency of mutation can vary from one gene to another; different genes in corn showed variation of mutation rates by 400-fold [3].

In addition, little is known about non-coding regions except that they possess signals that regulate transcription and translation processes. For example, the non-coding region upstream from a gene contains a ribosomal binding site which is the initiator for translation of amino acids. Information content of these areas is instrumental in gene finding [2,8].

Prokaryotes, cells without a nucleus, tend to have their genes encoded on DNA in one continuous nucleotide succession. Nuclear-celled organisms’ genes are interrupted by non-coding sequences called introns. Except for the fact that they are spliced out of the sequence before translation, not much is known about these regions. From a coding point of view, these otherwise useless bases would be perfect for containing error-control information such as parity bits. Intron sequences are prime candidates for information and coding analysis.

Mac Donaill suggests that nature prefers an alphabet of four due to the parity code structure in hydrogen donor-acceptor patterns of purine and pyrimidine molecules [7]. While our methods checked for coding structure on the nucleotide sequence itself, there is also the possibility of structure in the actual chemical bonds between complementary bases.

6. REFERENCES

[1] Anastassiou, D., “Genomic Signal Processing,” *IEEE Signal Processing Magazine*, July 2001.

[2] Borodovsky et. al. GeneMark: A Family of Gene Prediction Programs, http://opal.biology.gatech.edu/GeneMark/.

[3] Burdon, R.H., *Genes and the Environment,* Taylor and Francis Inc., Pennsylvania 1999.

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

[5] GenBank, National Center for Biotechnology Database, http://www.ncbi.nlm.nih.gov.

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

[7] Mac Donaill D.A. “A Parity Code Interpretation of Nucleotide Alphabet Composition,” *Chemical Communications, pp. 2062-2063, 2002.*

[8] May E.E., Vouk M.A., Bitzer D.L., and Rosnick D.I. “A Coding Theory Framework For Genetic Sequence Analysis,” *Genomic Signal Processing Workshop (GENSIPS), Oct. 2002.*

[9] Shannon, C.E., "Mathematical Theory of Genetics," Doctoral Dissertion, 1941.

[10] Wang W., Johnson D.H., “Linear Transforms of Symbolic Data,” *IEEE Transactions on Signal Processing*, Vol. 10, pp. 628-634, March 2002.

[11] Wicker, S.B., *Error Control Systems,* Prentice Hall, New Jersey, 1995.

7. FIGURES

(3,2) Codeword Search

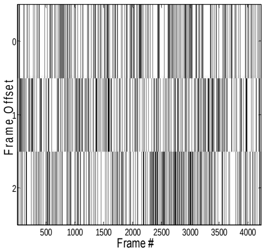


Fig. 4a: E. coli codewords which correspond to most common parity-check

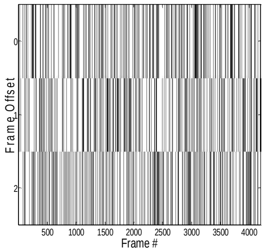


Fig. 4b: Random sequence codewords corresponding to most common parity-check

Linear Subspace Partitioning

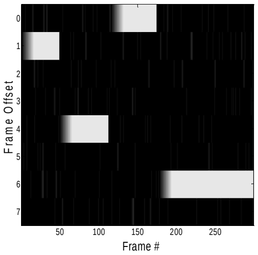


Fig. 5a: Analysis of an ideal (8,5) Coding Region, Test data

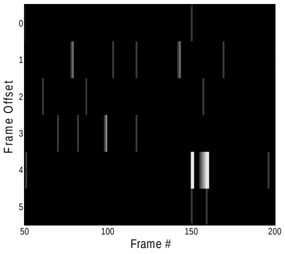


Fig. 5b: Analysis of a n=6 E. Coli MG1655 sequence