Applications of Coding Theory in Molecular Biology: An Overview

Amir Hesam Salavati
E-mail: hesam.salavati@epfl.ch
Algorithmics Laboratory (ALGO)
Ecole Polytechnique Federale de Lausanne (EPFL)
Supervisor: Prof. Amin Shokrollahi
E-mail: amin.shokrollahi@epfl.ch

I. Introduction

In this report, we will briefly review current works on the applications of coding theory in molecular biology. We consider various applications including source coding and channel coding.

There are two approaches toward investigating the relationship between coding theory and molecular biology. The first is to use coding and information theory to model different phenomena in the realm of molecular biology. This approach is becoming more and more popular in recent years as there is an ever growing amount of evidence suggesting the existence of error correction and compression codes in DNA which makes applications of coding theory in bioinformatics very promising.

The second approach is what we call bio-inspired coding and that is to identify coding mechanisms employed in cells and adapt them to our uses in communication systems. This method becomes very interesting when one notes that if there are any coding structures in molecular biology, it must be optimal because of millions of years of evolution and the force of natural selection.

Nevertheless, one should first prove the existence of such error control or compression methods in DNA and then identify their exact mechanism. These are the topics of interests addressed in this report. We review both arguments in favor and against the existence of coding structures in molecular biology and discuss current research on identifying such mechanisms, if any.

Coding and information theory for biological systems are still in their infancy. At this point, there are more questions than answers in this area. According to [15], we can divide the field into four groups, according to their research subject:

1) Applications of information theory to biology
2) Existence of error correction in biological information processing
3) Applications of coding theory to biomolecular computing (e.g., DNA computing)
4) Applications of coding theory to computational molecular biology and bioinformatics.

The rest of this report is organized as follows: In section II we briefly review the principles of molecular biology. We then discuss different methods to model gene expression process as a communication channel in section III. In section IV, we consider the applications of Error Control Coding (ECC) in molecular biology. We first explain arguments both in favor and against existence of error correction mechanisms in molecular biology. Then, we will review current methods to search for and investigate such structures in DNA. Section V discusses applications of coding theory in building better models for gene regulatory networks. We mention some applications of coding theory in constructing the phylogenetic trees of life in section VI. In section VII we address the applications of source coding in bioinformatics. Finally, section VIII concludes the report.

II. Introduction to Biology

All living things are made up cells. In other words, cell is the smallest component of life. Therefore, in order to examine biological principles governing the life, we focus on cells.

Each cell needs to perform certain activities to remain alive. These activities could be categorized into two classes:

- Activities required for living such as ”eating” (to supply the cell with energetic materials) and ”breathing” (to burn materials and get energy out of them).
- Passing information about above activities to next generations.

Cells employ proteins to perform the first type of tasks and DNA for the second types. In fact, DNA stores the recipe for making proteins and is passed to future generations. In addition to these two families of molecules, i.e. proteins and DNA, there is a third kind of molecule called RNA which is the interface between DNA and protein. In other words, RNA reads DNA, decode it and constructs proteins. In what follows, we become more familiar with each of these molecules.

A. Deoxyribose Nucleic Acid

Deoxyribose Nucleic Acid (DNA) is a long molecule responsible for passing information on how to build vital proteins to cells of next generations. DNA is composed of two parallel chains that form the famous helix shape.

Each chain is a long sequence of smaller building blocks called nucleotides. Nucleotides are some times called bases as well. All nucleotides are composed of a phosphor part (p), a saccharide (sugar) part (deoxyribose in DNA and ribose in

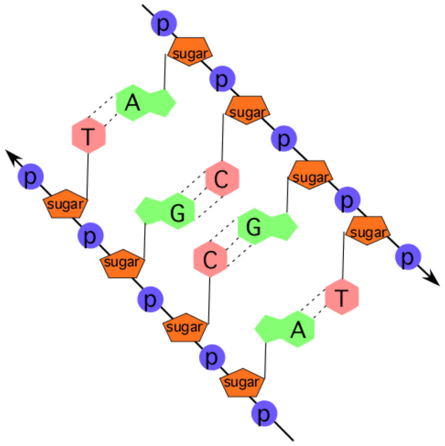


Fig. 1. Nucleotides structure.

RNA) and a base, the nucleic acid, that determines the type of nucleotide. Figure 1 illustrates the structure of nucleotides and DNA.

There are four types of nucleotides: adenine (A), thymine (T), guanine (G) and cytosine (C). Of these, (A, T) and (C, G) are complements of each other meaning that they form a bound if they are close enough. As a result, the second chain in DNA is the complement of the first chain: wherever we have A in the first chain we will have T in the complementary chain and so on. This introduces redundancy in DNA which as we see later, has an important role in reducing error rate of protein construction process.

Since DNA is composed of nucleotides, we can view it as a long sequence of letters that are drawn from the quaternary alphabet $\{A, T, C, G\}$. From now on, when we talk about DNA, we mean its equivalent sequence which we facilitates handling them in different computational and coding algorithms.

The length of DNA varies in different species from a couple of hundred thousand nucleotides in viruses and bacteria to a few hundred billion bases in mammals. Note that the length of the DNA does not necessarily reflect the evolutionary levels of the species. For example, the length of human genome is about 3 billion while that of hippopotamus is about 100 billion.

As mentioned before, DNA is responsible for storing the recipe for building proteins. In order to see how DNA manages to perform this task, we must first become familiar with genes.

1) Genes: A Gene is a part of the DNA that encodes the proteins regarding certain characteristics of the species. Different genes have different length, some are short and some others are longer.

In primary species (prokaryotes), there is gene for each characteristic of the species and genes are continuous. In other words, if a bacteri has 1000 characteristics, it has 1000 genes for each of them.

However, in more developed species (or eukaryotes), such as vertebrates and plants, genes are not continuous. To be more specific, genes are composed of various smaller parts called *exons* that are separated by sequences of different lengths called *introns*. Exons of a gene could be located far from each other: one of them could be at the beginning of the genome while the other one is located 1 billion nucleotides away.

While this non-continuity increases the complexity of protein construction, it helps the DNA to encode characteristics of the living being much more efficiently. It is necessary to have a single gene for each specification of the species any more. In contrast, since a combination of exons are used to represent a characteristic, DNA can reuse some of these exons as well as others to encode for a different property of the species. For instance, there are about 50000 genes in human genome coding for all the characteristics (internal and external) of humans.

2) Coding and Non-coding DNA: Not all of the DNA are composed of genes. The parts of DNA that make genes are called *coding region* since it contains codes for protein construction. The rest of the DNA is called the non-coding region. Note that since genes are spread all over DNA, these two regions are not continuous.

In prokaryotes, the coding region constitutes most of the DNA, up to 98% in some cases. However, and in contrast to expectations, the situation is reversed in eukaryotes as it is the non-coding region that constitutes most of the DNA. For example, in human genome, only 3% to 7% of the DNA is coding region and the rest is non-coding. In other words, more than 90% of human DNA does not encode proteins and does not contain genes.

For a long time it was believed that the non-coding region is useless. As a matter of fact, it used to be called ”junk DNA”. However, recent findings shows that non-coding region is even more structured than the coding region and contains controlling sequences that are necessary for gene expression. Nevertheless, the exact functionality of this region is still unknown.

B. Proteins

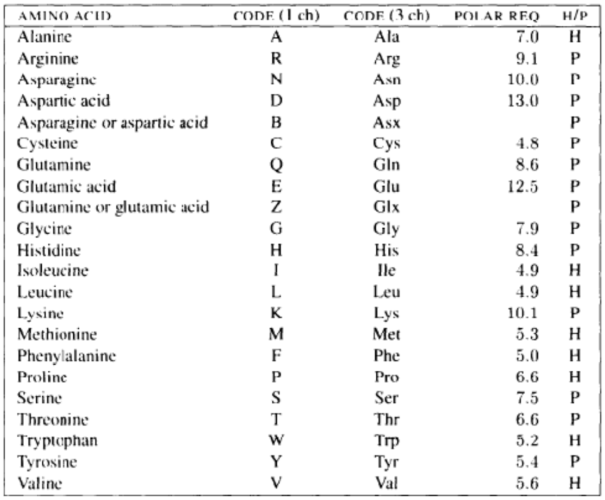
Proteins are responsible for all vital activities of the cell. They keep the cell, and hence the living being, alive. The list of activities performed by proteins include: transmission of various signals inward and outward of the cell, transporting small molecules, building various structures inside the cell, making enzymes and controlling intra-cell activities.

Like DNA, proteins are also a log sequence of smaller building blocks called amino acids. However, in contrast to DNA whose 3D structure is always a helix, proteins appear in various 3D shapes. In fact, that is the 3 dimensional structure of the protein that determines its functionality.

Moreover, instead of having a quaternary alphabet, amino acids form a 20-letter alphabet. Each amino acid is composed of two parts: the one which is similar in all amino acids and the one that determines the properties of each amino acid. This later part is called the $Rgroup$.

Depending on their properties, amino acids are divided into four categories [5]:

- : Amino acids with positive charge.
- : Amino acids with negative charge.



AMINO ACID	CODE (1 ch)	CODE (3 ch)	POLAR REQ	H/P
Alanine	A	Ala	7.0	H
Arginine	R	Arg	9.1	P
Asparagine	N	Asn	10.0	P
Aspartic acid	D	Asp	13.0	P
Asparagine or aspartic acid	B	Asx		P
Cysteine	C	Cys	4.8	P
Glutamine	Q	Gln	8.6	P
Glutamic acid	E	Glu	12.5	P
Glutamine or glutamic acid	Z	Glx		P
Glycine	G	Gly	7.9	P
Histidine	H	His	8.4	P
Isoleucine	I	Ile	4.9	H
Leucine	L	Leu	4.9	H
Lysine	K	Lys	10.1	P
Methionine	M	Met	5.3	H
Phenylalanine	F	Phe	5.0	H
Proline	P	Pro	6.6	H
Serine	S	Ser	7.5	P
Threonine	T	Thr	6.6	P
Tryptophan	W	Trp	5.2	H
Tyrosine	Y	Tyr	5.4	P
Valine	V	Val	5.6	H

Fig. 2. List of all amino acids as well as their polarity and hydrophobicity/hydrophilicity [5].

- : Hydrophilic amino acids: these are the amino acids that are electrically neutral. However, since they have an asymmetric charge distribution, they tend to conform a hydrogen bound with water.
- Hydrophobic amino acids: these amino acids are both neutral and have a symmetric charge distribution. Hence, they do not attract to water.

Figure 2 lists all the amino acids together with their symbols and their characteristics.

1) Codons: As mentioned before, DNA encodes proteins. DNA stores a representation of amino acids called the ”genetic code”.[1] Since there are 20 amino acids, we need at least 3 nucleotides to be able to represent them ($4^3 > 20$). These nucleotide triplets are called codons.

Having 64 possible sequences and only 20 objects to encoded allows a lot of flexibility. However, instead of leaving the remaining 44 sequences useless, the ”genetic code” assigns them to amino acids as well. In other words, certain amino acids are represented by more than one codon. 61 out of 64 codons are dedicated to coding amino acids and the remaining three are reserved for stop codons, which signal the end of a protein to RNA. Figure 3 indicates the amino acids as well as their corresponding codons [5].

Each codon represent an amino acid and a protein is a sequence of amino acids. Therefore, each protein could be represented by a sequence of codons in DNA. RNA reads DNA codon by codon and ”decodes” them to construct the corresponding protein. We become more familiar with RNA in the next section.

C. RiboNucleic Acid

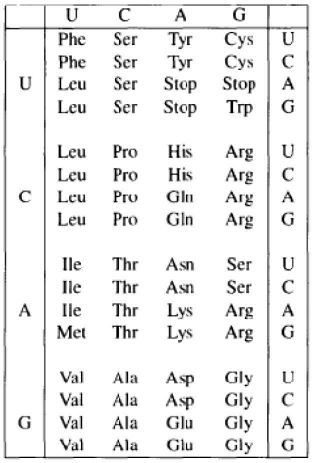
RiboNucleic Acid or RNA is another important molecule in the realm of molecular biology. RNA is chemically similar to DNA with three major differences:

1) It contains ribose saccharide instead of deoxyribose in DNA.
2) It is composed of one strand instead of two.
3) Instead of thymine, it has another type of nucleotide called uracil (U) in its structure.

In fact, RNA could be considered as the ancestor of DNA. Several primary species including viruses still use RNA to carry out their genome.

1) Protein Synthesis Process: There are many types of RNAs. However, the one we are interested in here are the messenger RNA (mRNA) and transfer RNA (tRNA), which are the interface between DNA and proteins [5]. In protein synthesis from DNA, a particular enzyme, called RNA polymerase or transcription factors, first unzips the DNA into two single strands and then makes a copy of the part of DNA that represents the target protein. RNA polymerase detects this region by binding to a particular sequence, called *promoter sequence*, in the DNA which is usually located before the target gene in DNA. RNA polymerase then starts reading codons consecutively and making a copy until the stop codon is reached. The resulting copy is called mRNA. This step is known as *transcription*. The total number of transcription factors (TFs) encoded by a genome increases with the number of genes in the genome [20].

Messenger RNA is then ”decoded” to construct proteins. Information about the target protein is carried out by mRNA to ribosomes. Ribosomes are the components of cells that build proteins from corresponding amino acids. The output of ribosomes are then handed to tRNA that are short RNA sequence (of about 80 nucleotides) that transfers a specific amino acid to a growing polypeptide chain which leads to protein formation in the end. This step is known as translation.



	U	C	A	G	
U	Phe	Ser	Tyr	Cys	U
	Phe	Ser	Tyr	Cys	C
	Leu	Ser	Stop	Stop	A
	Leu	Ser	Stop	Trp	G
C	Leu	Pro	His	Arg	U
	Leu	Pro	His	Arg	C
	Leu	Pro	Gln	Arg	A
	Leu	Pro	Gln	Arg	G
A	Ile	Thr	Asn	Ser	U
	Ile	Thr	Asn	Ser	C
	Ile	Thr	Lys	Arg	A
	Met	Thr	Lys	Arg	G
G	Val	Ala	Asp	Gly	U
	Val	Ala	Asp	Gly	C
	Val	Ala	Glu	Gly	A
	Val	Ala	Glu	Gly	G

Fig. 3. Amino acids and the ”genetic code” which determines the codon to amino acids assignment [5].

[1] We have put the term ”genetic code” inside quotations because it is not a code according to communications systems terminology. It is just a mapping between nucleotides and amino acids. We have kept this convention everywhere in the report so that there is no confusion between this mapping and actual coding algorithms. Nevertheless, as we will see later, this mapping contains some interesting properties from coding-theoretical point of view.

Note that this is just a simple explanation of what actually happens in the cell during protein synthesis process. In particular, it is much more complicated in eukaryotes where genes are no-continuous, i.e. introns separate exons and need to be *spliced* out before building the protei. This is known as gene splicing.

D. Gene Regulatory Networks

Genes are not independent of each other. The expression of certain genes affect other genes and vice versa. A gene is said to be expressed if its mRNA or protein can be found in the cell cytoplasm. Moreover, the activity of a gene could be measured via the amount of its corresponding mRNA or protein.

Whenever a gene is expressed, it could cause the expression or suppressions of other genes. In other words, it could switch other genes on or off. These iter-gene interactions could be modeled by a network called Gene Regulatory Network (GRN).

GRNs are an important field in system biology [18] because complex interaction patterns among genes in a genome has a big impact on our understanding of several biological procedures including disease development (particularly cancer).

In the most general model of GRNs, the influence of genes $1, \ldots, n$ on gene i is captured by the f ollowing equation:

$$\frac{dv_i}{dt} = f_i(v_1, \ldots, v_n) \tag{1}$$

where f_i is an arbitrary function depending on the model and v_j represent the expression level of gene j.

There are various mathematical models for modeling GRNs. However, the most widely used model is the so called NK-model [19], [12]. Here is how NK-model works: we have N genes and each gene is affected by K other genes. As already mentioned, these genes act as binary switches: they are either on or off. Their state depends on the activities of other genes. This control action is known as epistasis and is described in terms of a *genetic graph*. The nodes of this graph represent genes and a directed edge exists between two interacting genes. The average *incoming degree* of the nodes is K.

For the purpose of analysis, one can describe the genetic graph by a bipartite factor graph, where the left hand side nodes represent genes, and the nodes on the right hand side represent control nodes for each gene. Genes connected to control node C_j exhibit an influence on the operation of gene G [28]. For example, if there is an edge between the vertices L_i and R_j on the left and right side, respectively, then gene V_i affects gene V_j, i.e. if V_i is switched on, V_j is switched on or off. This transformation is shown in figure 4. In the figure, the variables Y_i represent the measured expression levels of the genes and one can think of $P(Y_i|V_i)$ as describing the unknown and noisy measurement channel.

III. Modeling Gene Expression and DNA Replication as a Communication Channel

There are different mechanisms of information exchange at various levels of the genetic systems: from nucleotides to codons, from genes to proteins and from an ancestor to its descendants. There are many approaches to model these processes from information theoretical point of view. Three of these models have become more popular. The first of is suggested by Gatlin [14]. In Gatlin's model, DNA is a coded sequence which encodes vital information. The decoded messages are the amino acids which build proteins in the end.

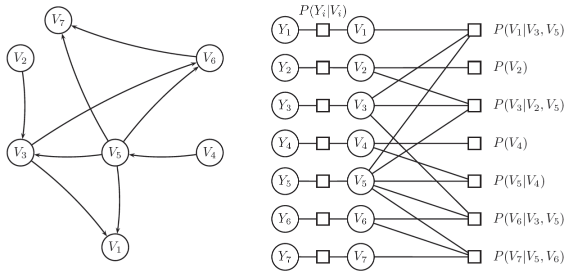


Fig. 4. Gene network and its corresponding gene factor graph [7]

Gatlin has also modeled DNA from a computer programming point of view. In her opinion, the coding part of the genome is the information source for a program which is controlled by functions lying in the non-coding parts of the DNA [14].

The second model is due to Yockey [35]. His model is based on data storage systems and Turing machine. DNA is assumed to be the input tape where sequence of bits are stored. The tape is feeded into the Turing machine equivalents, RNA molecules. The output is the amino acids, just like the Gatlin's model [14].

In Yockey's model, the process DNA→mRNA→protein is modeled as a communication channel. Here, DNA represent the source bits. The transcription process (which transforms DNA into mRNA) is equivalent of encoding source bits. The encoded sequence (mRNA) is then transmitted through the channel, where noise is added. There are two types of noise here: genetic noise, which is responsible for mutations, insertions and deletions of nucleotides, and a general type of noise which models the effects of medium, such as thermal noise, radiation and cell environment, on the mRNA molecules. The noisy message is then decoded via the translation process during which proteins are constructed from mRNA. Figure 5 illustrates the Yockey's model.

Since DNA represents stored source bits in Yockey's model, it is probable that similar to man-made storage systems, error correction codes are used in DNA to protect transmitted data over the noisy channel. In fact, Yockey himself suggests that the redundancy in the codon-to-amino-acid mapping is used as part of the error protection mechanism [35].

The third proposed model is the one suggested by May et. al. [25]. In their suggested framework, DNA is the output of and encoder which encodes biological information and adds error control capabilities to this sequence. The DNA replication process is the communication channel. The decoding process is done by RNA molecules after which amino acids are given

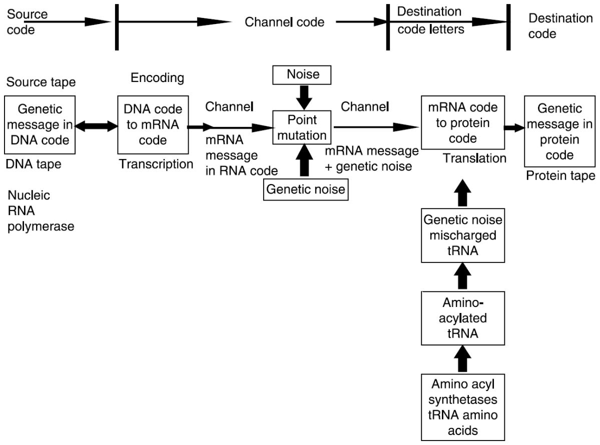


Fig. 5. Yockey's model of gene translation as a communication system [27].

as the output messages. May et al.'s model is shown in figure 6.

Another important point in analyzing coding properties of DNA is identifying system characteristics such as the channel capacity [27]. To calculate the capacity, we must have the error probability of the channel. This translates into mutation rate in May et al.'s model in which the DNA replication process is modeled as a communication channel. Some results on the rate of mutation in different species could be found in [9] and [10]. Based on these numerical values on mutation rate, channel capacities of different organisms are illustrated in figures 7 and 8 [27].

IV. Error Control Coding and Molecular Biology

In this section, we will discuss various applications of error control coding in genetics. We will first discuss the issue of existence of error correction mechanisms in DNA. Both arguments in favor of and against existence of such mechanisms are addressed. We then consider various approaches to identify error correction mechanisms, if any, in DNA. Finally, we will mention the concept of soft codes and see why it is so important for investigating error correction mechanisms in genome.

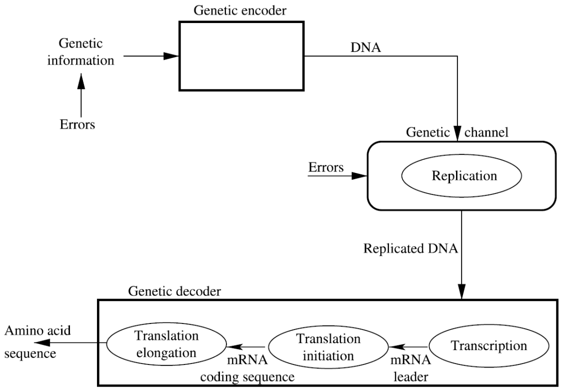


Fig. 6. May et al.'s model of DNA replication as a communication system [27].

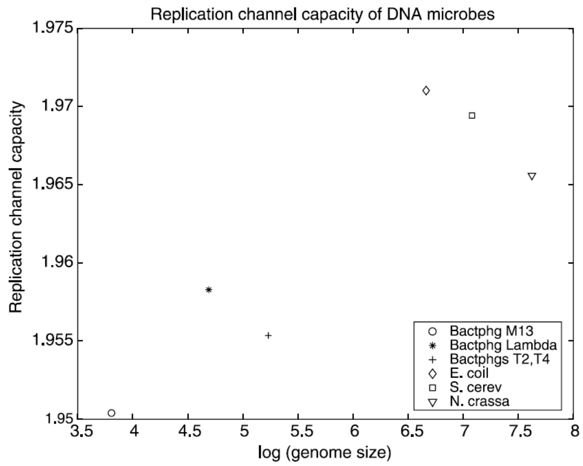


Fig. 7. Capacity of prokaryotic replication channels [27].

A. Evidence in Favor of Existence Error Correction Mechanisms in Genome

The redundancy in genome and DNA is an accepted fact. However, whether this redundancy is because of an error correction code is still not known [27]. Nevertheless, we have an ever increasing amount of evidence that suggest error control codes must be present in genetic systems.

First of all, we know that mutations in the genome replication due to chemical agents or radiations, are responsible for aging and certain diseases like cancers. Noting that genetic data is replicated several million times in evolutionary time

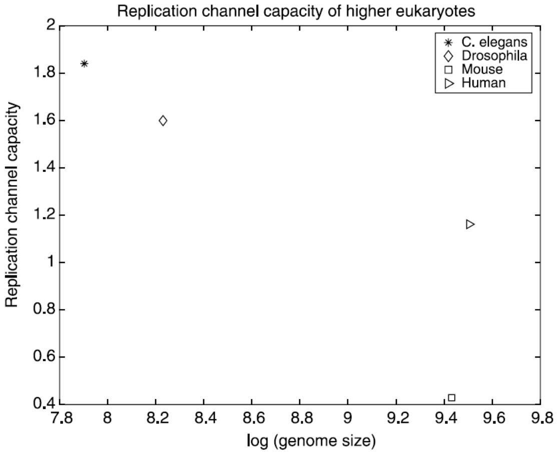


Fig. 8. Capacity of eukaryotic replication channels [27].

scale, if there were no error correction mechanism, the accumulation of errors during periods million times longer would simply make genetic communication, and hence life, impossible [4]. In other words, the number of errors in a k-symbol message that has been replicated t times is approximately equal to the number of errors in an unreplicated message with $t \times k$ symbols. Thus, in order to have a reliable message during the life cycle of an organism (let alone during the evolutionary time scale) the message must have good methods of error correction.

Moreover, it is quite surprising that while the process of DNA replication occurs in a noisy environment, the cell, the replication error rate is as low as 10^{-9} mutations/nucleotide. This value is noticeable as DNA replication procedure alone has an error rate of 10^{-3} to 10^{-5} [31]. One might argue that the final low error rate is a result of DNA's internal proofreading mechanism: when copied, the helical structure unzips into two separate strands. RNA uses one strand to read DNA and then check the read sequence with the other strand. If they matched, it proceeds. Otherwise, it waits until the correct nucleotide is restored. This simple proofreading reduces the error rate to approximately 10^{-10}. Moreover, there are other proofreading mechanisms as well.

Proof-reading mechanism of DNA replication process is another phenomena suggesting existence of error correction methods in DNA. Proof-reading mechanisms are observed during DNA replication, and when the activity of these polymerase mechanisms are blocked, error rates increase from 10^-6 to 10^{-3} [31]. However, the proofreading mechanisms can at best ensure that the copy is faithful to the original. In other words, they can correct the errors which occur within the replication process but not those that may affect the original itself [4].

In addition to above *qualitative* arguments, several researchers have tried to address this issue in a *quantitative* manner. MacDonaill has discovered a parity check code interpretation of nucleotide composition [23]. In his model, nucleotides form a 4-bit even parity code. We will discuss his work in more details later.

At a different level, there are several works suggesting that ”genetic code”, i.e. codon assignments to amino acids, are optimized in a way to admit error minimization during translation process. As an example, in [11] authors claim that natural selection has chosen amino acid to codon assignment such that translational errors are minimized.

In order to prove their claim, the authors have first defined a measure to evaluate the codewords of a code quantitatively, just like the hamming weight in coding literature (in this case, the authors have used polarity of the corresponding amino acid as the measure). Then, given a codon, they do a single mutation in the codon and ”decode” the corresponding amino acid of the resulted codon. They then measure the distance between the new and original amino acids to assess the strength of the coding method. The ideal case is that a single mutation does not change the resulting amino acid. By repeating this process for all of the three nucleotides in a codon, and for all codons, and then averaging the results according to a weight function, they obtain a measure of how good a code is. The lower this measure is, the stronger the code will be.

The authors have then build many random ”codes” by arbitrarily assigning codons to amino acids and evaluated their strength based on the aforementioned measure. Their result show that only 1 of codes (from a pool of 1 million random codes) performed better than the standard genetic code. Therefore, their work clearly suggest that the codon assignment is optimized by evolution in a way to minimize translation errors.

Furthermore, as mention in [6], the codons which ”code” for one amino acid are more closely related to one another (in sequence) than they are related to codons that code for other amino acids. In other words, codons that code for one amino acid differ in several cases by just one nucleotide. Thus, single nucleotide mutations (especially in the third location) will often not change the resulting amino acid.

Nevertheless, note that there are a few drawbacks in the approach of [11]. First of all, the strength of code is their method is completely dependent on the feature used to measure the codewords. In other words, and as the authors have mentioned themselves in their paper, while some properties may hint for support of existence of error correcting codes, it may not happen for other properties. Therefore, it is of outmost importance to choose the measure properly.

A very simple yet influential coding theoretical argument is used in [16] in order to explain a mismatch between theory and biological findings regarding the number of transcription factors (TFs) in a genome. As mentioned earlier, transcription factors regulate gene expressions by binding to the DNA at the promoters of the target genes.

Transcription factors are categorized into several families. The categorization is usually based on the length of the binding sites of TFs. Now consider the TF family whose binding site's length is equal to n. This length depends on the genes that the binding site should represent. A binding site with length n can represent $4^n/2$ genes (we have divided by two because half of the sequences are complementary). Therefore, a family of TFs that bind to sites with length n should have 2^{2n-1} members [20]. However, this number is much less in reality. As an example, for $n = 6$ the number of TFs is 300 instead of 2048.

A coding-theoretic explanation for this mismatch is mentioned in [16]. Suppose we would like to design a codeword with quaternary alphabet and code length of n. This code lies in the 4^n space. Now, if we would like to add one bit error correction capability to this code, the total number of codewords will reduce to $N = \sum_{i=0}^{1} \binom{n}{i}(4-i)^i$. For $n = 6$, this number will equal to 108 which is quite close to the quantity found in reality, i.e. 300.

A possible explanation for this difference is that in our code design, we assumed that the spheres around different codewords lie completely sperate of each other. This allows for complete correction of bit of error. However, if we relax this constraint by admitting error correction for part of the codewords, then these spheres could share common sequences. In this way, we can have more codewords as we need more spheres to cover up the space [16]. The problem of one bit error correction could also be solved by the help of synonymous codewords. As mentioned before, for certain amino acids, there are more than one codons to encode them. Therefore, if the codewords whose spheres are not completely separate differ only in these synonymous codons, then whichever is selected as the codeword, the corresponding protein has the same functionality as it must have since both codons encode the same amino acid.

A further argument in favor of the existence of ECC in genome comes from the fact that this hypothesis in genome helps us explain some puzzling phenomena very easily, which otherwise could not be explained simply [4]:

- The fact that the species are discrete and the fact that evolution proceeds by jumps: a puzzling fact in biology is the discreteness of species. Why don't we have a *spectrum* of living things instead of having different species and families? Error correction codes could explain this phenomena quite easily and in a neat matter: small number of mutations (errors in close distance of current codewords) are corrected while the ones with larger distances are left uncorrected. Hence, they result in new species! In other words, genomes located in a distance less than that of the minimum distance of the code can not exist in outside world!
- The trend of evolution towards increased complexity: longer and more complex genomes means better error correction (as the length of the genome, i.e. codeword, goes to infinity, coding becomes better, which is a well known fact in coding literature.) In fact error rates are higher in simple species such as viruses and bacteria compared to that of highly developed ones such as mammals. This may suggest the role of natural selection in developing error correction mechanisms in DNA during evolutionary time scales.

And the last but not the least is the applications of spin glasses model in both coding theory and molecular biology. The applications of spin glasses and statistical physics are quite well known in coding theory. On the other hand, there exists interesting relationships between spin glasses and origin of life [15] which encourages further research from this point of view on applications of coding theory in bioinformatics.

B. Arguments Against the Existence Error Correction Mechanisms in Genome

Although there is an every growing amount of evidence about the existence of error control methods in DNA, there are also several arguments against such mechanisms.

First of all, neither of the works mentioned in the previous section is a *proof* of the existence of error correction in DNA. They all *suggest* that such mechanisms must exist. Hence, there could be different explanations for different phenomena leading us to existence of error correction hypothesis.

Furthermore, some of the preliminary works on finding error control coding in DNA were unsuccessful [21], [31]. However, as they have already mentioned, their approaches were too simplistic and limited. Both approaches considered block codes and proposed algorithms to search for signs of block-coding in DNA. Therefore, neither of them addressed the existence of convolutional codes. In fact, Liebovitch et al. [21] suggest that a more comprehensive examination would be required.

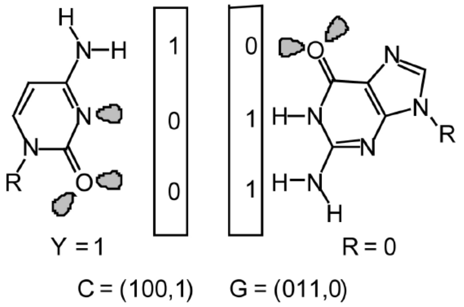


Fig. 9. Binary representation of nucleotides [23].

In brief, while there are arguments both in favor of and against existence of error correction methods in DNA, the issue remains an open question at this point and demands further research and development of more sophisticated algorithms for identifying coding structures in genome. In the next section, we will review some of the current approaches toward finding such structures.

C. Investigating the Error Correction Mechanisms in Genome

Before reviewing the literature on possible error correction methods in molecular biology, note that such mechanisms could have been implemented in different levels. We have nucleotide's level, where MacDonail has already discovered an even parity code [23]. We also could have error correction at ”genetic code” level, i.e. the mapping between codons and amino acids, as already suggested in [11] and [6]. A higher level is the genes level where genes form regulatory networks. It is suggested that the formation of genes in such networks admits error correction [28]. In other words, when a gene is mistranslated, other genes act in a way that the effects of mistranslation is neutralized. Hence, one must keep in mind that error correction could present in various levels of the DNA at the same time. We discuss the concept of nested codes later in this section which addresses this issue.

As already mentioned, MacDonail has discovered an even parity code in nucleotides [23]. The key part of his work is the way nucleotides are represented in the binary format. In his suggested scheme, nucleotides are represented by four bits: the first tree bits determine if different bounds of a nucleotide accept (0) or donate (1) of hydrogen. The last bit identifies nucleotides based on purine/ pyrimidine (R/Y) size motif ($R = 0$ and $Y = 1$). Figure 9 shows an example of the proposed binary representation.

Constructing this binary representation for all nucleotides, we get 1001 for C, 0110 for G, 1010 for A and 0101 for U.[2]

[2] U is equivalent of T in RNA. Since RNA is addressed in this work, the author has considered U instead of T.

As one can readily observe, all these binary sequences have an even number of ones, i.e. they form a $(4, 2)$ even parity code.

Whether this parity code structure is just a coincidence or shaped by natural selection through evolution is still an open question. Nevertheless, if it is formed via natural selection, it suggests that the probability of transforming one nucleotide into another one is reduced using an even parity code, a hypothesis to be verified in future.

The coding model of MacDonail [23] has a superiority over its counterparts in that it can explain a mystery in molecular biology: why nature has used these four nucleotides while some other combinations with similar chemical functioning were also possible? While some researchers suggest that nature has simply failed to discover these nucleotides [29], MacDonail explains this phenomena by noting that the Hamming weight of the binary representation of these nucleotides is odd, which violates the even parity constraint.

There is one point though that could affect MacDonail's analysis. Here, the Hamming distance may be not homogenous for all nucleotides. For instance, one nucleotide may differ from the other based on their D/A pattern (Donator/Acceptor) but different from the other one based on the purine/pyramid type. The chemical bound of these two may not be as strong as each other. Therefore, while the Hamming distance between both pairs is two, one of them may require less energy to transform into the other one, meaning that the distance is not equal for these two pairs though their Hamming distance is two.

In [30], the authors employ an interesting approach in using finite fields to represent the four nucleotides and then use finite field arithmetic to find the parity check matrix of the genetic coding scheme. They propose an approach to investigate the existence of a linear $(n, n-1)$ code. Their method is based on dividing the whole DNA sequence into frames of n nucleotides, n being the codeword length. Using Gram-Schmidt algorithm, they identify the basis of the subspace formed by these vectors of length n. Having built the basis, the single vector of the parity check matrix could be found by looking for a base vector whose corresponding coordinate is zero when all vectors are expressed as a linear combination of basis.

The main idea behind the suggested approach is to check the dimensionality of n-tuples in genome. If there is a coding method there, then there must be redundancy among the n-tuples and the dimensionality must be less than n. While the approach mentioned in [30] focuses on identifying linear $(n, n-1)$ block codes, in [31] Rosen extends this framework to look for any sign of redundancy in genome.

Generally speaking, any method for finding the dimensionality of a subspace in a space defined by n-tuples would be useful in this area. More specifically and as mentioned in the paper, here is a need for a general approach to find k-parity bits placed in *any order* in *any n-size code* to identify an (n, k) linear block coding structure in DNA.

While the idea considered in [31] is very interesting and promising, their model is quite unrealistic (That's probably the reason they have not yet found any sign of linear coding in genome of primal species such as E. Coli bacteri). For one, the author has assumed that the whole genome is a coded sequence and divided it into N vectors (codewords) with length n [31]. She has then proposed a novel approach to compute the dimensionality of the resulting $N \times n$ matrix, obtained by putting the codewords and the rows of the matrix. However, the assumption that the whole genome is a coded sequence is rather simplistic and it is probable that some parts are coded while the other parts are left uncoded for evolutionary reasons.

Schmidt and May [33] exploited graph-theoretic methods to analyze error correction and detection properties of Escherichia coli K-12 translation initiation sequences. They first prove that in contrast to binary random sequences, binary block codes form distinctive cluster graphs. Then, they have applied their method to Escherichia coli K-12 translation initiation sequences. Their results show that non-initiation sites fail to cluster into distinct groups. However, cluster formations in valid initiation sequences are clearly observed, suggesting the possibility of existence of an error control coding mechanism in E. coli's translation initiation sites.

May et al. [24] have modeled mRNA as a noisy, systematic zero-parity encoded signal and the ribosome as an (n, k) minimum-distance block decoder. Their idea is to build the parity check matrix of the (n, k) code by finding the set of vectors that are orthogonal to codewords. Then choose the *best* possible parity check matrix. Here, the best matrix is the one with the highest fitness, which is a function of the number of zeros in H and its syndrome S. Assuming H to be in systematic forms, i.e. $H = [P^T|I]$, then the fitness is defined according to equation (2).

$$Fitness(H) = \alpha \frac{|\text{zeros in S}|}{|S|} + \beta \frac{|\text{non-zeros in P}|}{|P|} \qquad (2)$$

Where α and β are two constants such that $\alpha + \beta = 1$. Moreover, $|X|$ denotes the number of elements in X.

In [26] a similar approach is used to search for $(5, 2)$ and $(8, 2)$ block codes. First of all, each nucleotide is represented with an integer between 1 to 4. Then, all *valid* codewords are constructed, i.e. those that the sum of all n bits equals to zero (modulo 4). Next, a particular section of DNA is divided into blocks of 5 and then 8 nucleotides. Finally, each block of length of n in DNA is compared to the set of valid codewords and the one which has the lowest Hamming distance with it is selected as the corresponding codeword. This process is repeated for different reading frames and for different codes (different parity check matrices as discussed above). In the end, the code with least distance to the corresponding section of DNA is selected. The ideal cases is that we find a block code (n, k) whose distance is zero from DNA, i.e. it exactly models the coding structures in genome.

May et al. have tested above approach for $(5, 2)$ and $(8, 2)$ codes over the messenger RNA (mRNA) of Escherichia coli [26]. Both the $(5, 2)$ and $(8, 2)$ models were able to distinguish coding and non-coding parts of DNA.They both also indicate the existence of key regions within the mRNA leader sequence. Hence, this approach is also suitable for finding coding and non-coding regions in DNA as well as recognizing the ribosomal binding site (the location of the

ShineDalgarno sequence).

Nevertheless, this method is quite limiting. First of all, we have to search for all possible n and k as well as all reading frames (n reading frames for a block length of n). Furthermore, as mentioned in the paper, convolutional codes seem to be a better model in certain cases [26]. In their viewpoint, due the memory in convolutional codes, they can more accurately model the behavior of the ribosome as a decoder.

Recently, there has been some work on the relationships between iterative codes, such as LDPC, and genomics. The key properties of such codes are sparse matrices and fast decoding. The performance of the decoder improves over time and iteration by iteration. While this seems too bizarre to exist in DNA, there is growing amount of evidence that the coding techniques in genome, if any, are in a manner similar to that of LDPC-like codes. For one, as pointed out in [15], what DNA needs is a *fast* decoding algorithm. Therefore, employment of long block or convolutional codes seems to expensive for genome. However, iterative decoding could be useful by being fast and becoming better after every iteration. In this way, the biological system (either brain or DNA) could make *suboptimum* decisions and improve them over time, if situation allows for iterations in time.

This concept becomes more clear in context of GRNs. As discussed earlier, we can model GRNs with bipartite graphs to indicate the effects of different genes upon each other. The most widely used model for such graphs is the NK-model [19]. NK-model is a special case of Boolean Networks (BN) which has applications in biology, mathematics and communications. In fact, BNs contain LDPC codes as a special case [28].

There exist many biological indications that the proofreading mechanism of DNA transcription is intimately connected to the GRNs. Furthermore, malfunctions of the proofreading mechanism are known to cause disease such as cancer. Therefore, in [28] the authors conjecture that during the process of DNA replication, nodes of the genetic network are arranged in a special form that only allows valid genes as the codewords and an error-control code. If a mismatch occurs during the replication, some sort of fast decoding method is used to identify the erroneously copied genes. Then, these genes undergo another level of internal error control which leads to determining the erroneous nucleotides.

In other words, both genes and nucleotides are involved in the error control process. The global Genetic Error Control (GEC) code [28] uses genes rather than three nucleotides as its symbol while the local (internal) GEC uses the nucleotides as its symbols and acts within a gene. Therefore, gene interactions is (primarily) responsible for the error correction capabilities of the cell as well as the proofreading mechanisms of DNA. An interesting biological phenomenon to support this argument is given in [22]. It is know that the main repair pathway which is responsible for most of error correction capabilities is composed of enzymes called nucleases. These enzymes monitor DNA for any sign of damaged binding sites. To collect information about damages, nucleases move in saltatory manner, i.e. they jump from one location in genome to another. This suggests that the genome is a globally connected structure and the error correction code could operate on the global structure [28].

1) Soft Codes and Genetics: Battail has also introduced the concept of *soft codes* in genetics [4]. Basically, there are two ways of defining a code: specifying its construction rules (as communication engineers do) or specify the required constraints of the codewords. He suggests that the second approach is more appropriate for natural phenomena in which the assumption of deterministically specifying the construction rules seems nave.[3] Moreover, the constraints are expressed in a probabilistic matter. Therefore, the main parameters of a code, like its minimum distance, then become random variables.

To get a better understanding of soft codes, think of natural languages. In a natural language, there are a lot of constraints such as the properties of the vocal tract, which limits the number of possible words, constraints on the meaning of the actual words (lexicon) out of the pool of possible words, constraints of having meaningful sentences and so on. In other words, phonetics, lexicon, syntax (grammar) and overall meaning of the sentence are the constraints on natural languages. At the same time, natural languages have a great capability of error correction (both in oral and written forms).

Furthermore, a language is defined by distinct constraints acting at several hierarchical levels. For instance, phonetic constraints, which are due to the structure of the vocal tract, are more fundamental and rigid than constraints specific to a given language, which are mainly based on social conventions.

The same level of hierarchy and constraints are also present in DNA: chemical constraints on nucleotides and their pairings, the lexicon (genes) and meaningful (functional) proteins. These are examples of what is called *nested soft codes* [4].

In a nested code, some parts of data are protected more than other parts. In other words, some parts of the data are first coded and then the coded sequence is coded again using a possibly different coding approach and so on (similar to raptor codes).

There are some evidence that the error correcting mechanism in genome, if any, is a nested code. Vital genes need much more protection. In fact, these genes are preserved among generations of far-related species. For instance, the HOX genes which determine the organization plan of living beings are shared by humans and flies, which diverged from a common ancestor hundreds of millions of years ago [4]. A similar scheme has independently been used by Barbieri to describe the organic codes [1].

2) Technical Difficulties: Searching for error correction codes is much more difficult than one might imagine. First of all, what is the coding alphabet? The trivial answer to this question is that the alphabet are the four nucleotides used in construction of DNA. However, this is similar to saying that English alphabet is composed the set of vertical and horizontal lines that are used to build the actual letters. My colleagues and I have worked on this issue before and our results suggest that the letters of genetic language, if there is any language, is

[3] In my opinion, the first approach is similar to fixing the generator matrix while the second one is to determine the parity check matrix, in a probabilistic manner.

actually not the four nucleotides but a combination of them, just like the situation in natural languages [32].

Another challenge comes from the codes being nested. If this hypothesis is correct, i.e. the ECC in genome is nested, the set of alphabets used in each of the encoders in a nested code may be different from the other codes. Therefore, we have to find set of physiologically meaningful alphabets.

Furthermore, if a block coding scheme is used, we need to know the number of code and message bits, i.e. n and k in an n, k linear code. However, in case of DNA we neither know n nor k.

Moreover, we even do not know which type of error correction methods are used in genome, if it is similar to our current approaches in communications systems at all. As suggested by May et. al [26], convolutional codes seem to be a better model in certain cases.

In addition to all these problems is the reading frame issue. In a traditional coding method, the decoder knows the beginning of each new codeword. However, in a genome we have no clue about the beginning of a frame. Assuming codewords with length n, we must consider all possible reading frames from the beginning of the genome (total of n possible ways).

V. Applications of Coding in Gene Regulatory Networks

So far, we have only considered the most common application of coding theory in bioinformatics, i.e. existence of error correction algorithms in DNA. However, during recent years, different applications of coding theory in modeling different phenomena in genomics is proposed. A promising example of such applications is to use coding theory in order to model GRNs.

As mentioned before, GRNs are fully characterized by their topology and the functions determining influence of different genes on each other (see equation (1).

Recently, a new idea has been introduced, mainly due to Milenkovic and her team [7], which explores the use of coding theory in refining current models of GRNs. Here, the main problem we are interested in is inferring the influence functions of a GRNs from biological findings. Here, it is assumed that topology of the GRN is known, which is not a very limiting assumption, while the interaction functions f_i's in equation (1) must be determined.

In [7] and [8], the authors use polynomial interpolation techniques in coding theory to determine f_i's. In their approach, f_i is related to the polynomials used in Reed-Muller (RM) codes and show that the same approach used in decoding of RM codes could be used here to determine f_i if the number of observed biological data is sufficient, i.e. it is bigger than a threshold which depends on the minimum distance of RM codes.

Therefore, coding theory is used here to fully specify the parameters of the GRN model. interestingly, their approach has the advantage that it works in the noisy models in which the GRN is not a deterministic network but a probabilistic model which correctly reflects the stochastic behavior of biological phenomena. Furthermore, the data used in the biological analysis comes from DNA micro-arrays which scan the DNA of species. The micro-array itself introduces some noise in the data as well.

Hence, a coding-theoretic methods is advantageous here by being able to handle such noises. In addition, even though it is assumed that the topology of the network is perfectly known, the proposed framework can be extended to deal with uncertainties in the network topology [8]. Albeit this capability is achieved at the expense of larger amount of required measurements.

VI. Coding Theory in Constructing Phylogenetic Trees

Finding out the ancestors of current species and building the phylogentic tree of life is a very important field in bioinformatics. The main problem here can be stated as follows: we would like to a build a tree in a way that its leaves are current species. The one before last level, i.e. the parents of these leaves, are the ancestors of common species. For example, monkeys, chimpanzees and humans probably have a common ancestor which is their parent in the tree of life. Similarly for other species, their ancestors would be the parent node of these leaves in the phylogenetic tree, and so on until we end up with the first living species on earth as the root of the tree. Since most of these ancestors are extinct now, we would like to construct this tree merely based on the genomes of current species.

There are a lot of algorithms for constructing the phylogenetic tree, all of which are heuristics as the problem is NP-complete. Nevertheless, coding theory can have a very nice application here if we model the evolution as a communication channel in which an input (the ancestor of some species) is transmitted via the noisy communication channel (evolutionary time in this case). We have then sampled the outputs of the channel at different times (different noise values) to get different species. Our goal is to deduce the transmitted data based on the received noisy channel output. A similar idea is mentioned in [6]. In this case, noise is mutations that occur in a genome of a species during evolutionary time scales. Further research is definitely required in this area.

VII. Source Coding and DNA

While our may focus in this report was on channel coding, there also exists some approaches based on source coding to model different phenomena in molecular biology.

In [34], authors have investigated compression capabilities of DNA and its similarities to multiplexed codes [17]. Multiplexed codes are a family of suboptimal source codes designed to solve the issue of ”de-synchronization” linked with variable length codes. They are suboptimal in the sense that the compressed codeword length is minimized subject to the constraint that all codeword lengths are fixed. In multiplexed codes, we have two different sources: one with high priority and the other with low priority.

Let $A_H = \{a_1, , a_{N_h}\}$ be alphabets of the source with high priority. Furthermore, suppose that elements of multiplexed codes are composed of binary codewords of length c, where

$2^c > |A_H|$. Moreover, Let C be the set of binary codewords of length c. The multiplexed code is constructed in the following manner: divide C into $|A_H|$ subsets, called equivalence classes. Each equivalence class C_i is associated with one of the letters in A_H. Therefore, each letter in A_H is represented by *at least* one codeword in C. Nevertheless, it could also be represented by more than one codeword, depending on the size of its corresponding equivalence class. As a result, each codeword could be represented by two indices: i, j: the first one determines its class and the second one its rank inside the class. The second index could be used to encode the source with lower priority. However, in [34] the authors have not consider the way to code the information of the second source.

An important property of the multiplexed code is that there is no de-synchronization problem regarding the high priority source. In the presence of an erroneous symbol in the multiplexed code, at most one symbol of the high priority source will be affected and the de-synchronization will affect only the lower priority source [34].

Since we would like to be as close as possible to optimal compression, the size of equivalent classes are selected such that they minimized the expected length of the code, as shown in equation (3) [17].

$$(|C_1|, \ldots, |C_{A_H}|) = argmin(-\sum_{i=1}^{|A_H|} p(a_i)\log_2(\frac{|C_i|}{2^c})) \quad (3)$$

where $|C_i|$ indicates the cardinality of the equivalent class C_i and $p(a_i)$ is the probability of letter a_i appearing. If the size of equivalent classes are determined according to equation (3), then the resulting code will be able to convey a maximum information for the secondary source [34].

In [34] the authors have investigated the ”genetic code” from a source coding point of view. In the suggested framework, amino acids constitute the letters of the high priority source while codons represent the codewords for the source code. The authors have found the following similarities between the ”genetic code” and multiplexed codes:

- Both have fixed length, In multiplexed codes, the size of codewords are fixed. Likewise, in ”genetic code” all codons are composed of three nucleotides.
- Both contains equivalent classes. In the ”genetic code”, the set of codons that translate into a particular amino acid forms the equivalency class corresponding to that amino acid.
- In both cases, the number of codewords is greater than the source elements (64 versus 20 in the case of genetic code).

The similarities are not limited to the list above. Considering amino acids as the alphabet of high priority source, the authors have measured the frequency of amino acids for different species. Based on the distribution of amino acids, the cardinality of their equivalent classes are estimated according to equation (3). The estimation is then compared to actual cardinality of equivalence classes (the number of codons matched to each amino acid). The results show that the estimations are in good harmony with actual cardinalities for different species. Moreover, the estimations are more close to reality for species with higher evolutionary levels [34]. In other words, they become more accurate for vertebrates compared to invertebrates and bacteria, suggesting the role of natural selection in developing a kind of source coding mechanism in the ”genetic code”.

Furthermore, the authors have computed the expected length of the optimal multiplexed code for the ”genetic code” and compared it to the actual average length. Their results show that the expected length of the ”genetic code” is only very slightly greater than that of the multiplexed codes [34].

Another interesting finding of [34] is the optimality of the alphabet size in the ”genetic code” to convey information for the secondary source. If Q is the size of code alphabet ($Q = 4$ for genetic code), and c is the smallest integer such that $Q^c \geqslant |A_H|$, then the amount of information one can convey for the secondary source is $c/\log_Q(2)$. This value is maximized for $Q = 4$ among the values $Q = \{1, \ldots, 7\}$.

VIII. Conclusion

In this article, we did a literature review on applications of coding theory in molecular biology. Based on this survey, it seems that there are a lot of evidence on existence of error correction schemes in DNA. Furthermore, no contradictions were found between the hypothesis that natural genomic error-correcting means exist and the properties of the living world. On the contrary, it seems to account for a number of facts, especially of evolution, that conventional theories fail to explain. In addition, as discussed above, coding theory could have many other applications in this field other than finding ECC in DNA. Examples include applications of source coding in building more accurate models for GRNs and applications of source coding in investigating the ”genetic code”.

Nevertheless, it is quite soon to conclude that coding structures exist in genome. Because all the phenomena that could be explained using coding theory could also be possibly justified by the help of other hypotheses. However, coding theory makes a very strong candidate here and like other hypothesis, it requires deeper and more comprehensive studies and close collaboration of coding theorists with biologists.

In my opinion, this field is very promising, both for coding theorists and biologists as biologists could benefit from well developed coding theoretical models. Moreover, coding theorists could contribute a lot to this field and probably inspire from structures in DNA to design better codes.

References

[1] M. Barbieri, ”The Organic Codes”, Cambridge, UK: Cambridge Univ. Press, 2003.

[2] M. Barbieri, ”The Codes of Life”, Springer, 2008.

[3] G. Battail, Does information theory explain biological evolution? Europhys. Lett., 40(3): 343348, November 1997.

[4] G. Battail, ”Should Genetics Get an Information-Theoretic Education? ”, IEEE ENGINEERING IN MEDICINE AND BIOLOGY MAGAZINE, pp. 34-45

[5] P. Clote, R. Backofen, Computational Molecular Biology: an Introduction, John Wiley and Sons Ltd., 2000.

[6] Z. Dawy, P. Hanus, J. Weindl, J. Dingel, F. Morcos, ”On Genomic Coding Theory”, European Transactions on Telecommunications, Volume 18 Issue 8,Pages873-879, 2007

[7] J. Dingel, O. Milenkovic, ”Coding-Theoretic Methods for Reverse Engineering of Gene Regulatory Networks”, Proc. IEEE Information Theory Workshop, 2008, pp. 114-118.

[8] J. Dingel, O. Milenkovic, ”A List-Decoding Approach for Inferring the Dynamics of Gene Regulatory Networks ”, Proc. ISIT 2008, pp. 2282.

[9] J. W. Drake, B. Charlesworth, D. Charlesworth, and J. F. Crow, Rates of spontaneous mutation, Genetics, 148: 16671686, 1998.

[10] A. Bebenek, G. T. Carver, H. Kloos Dressman, F. A. Kadyrov, J. K. Haseman, V. Petrov, W. H. Konigsberg, J. D. Karam, and J. W. Drake, Dissecting the fidelity of bacteriophage RB69 DNA polymerase: Site-specific modulation of fidelity by polymerase accessory proteins, Genetics, 162: 10031018, 2002.

[11] S. J. Freeland, T. Wu, N. Keulmann, ”The Case for an Error Minimizing Standard Genetic Code”, Journal of Origins of Life and Evolution of Biospheres , Volume 33, Numbers 4-5 / October, 2003 , pp. 457-477

[12] Y. Gao, J. Culberson, ”An Analysis of Phase Transition in NK Landscapes”, Journal of Artificial Intelligence Research 17 (2002) 309-332

[13] I. Gat-Viks, A. Tanay, D. Raijman, and R. Shamir, A probabilistic methodology for integrating knowledge and experiments on biological networks. J Comput Biol, vol. 13, no. 2, pp. 16581, mar 2006.

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

[15] M. K. Gupta, ”The Quest for Error Correction in Biology”, IEEE Engineering in Medicine and Biology Magazine, Vol. 25, No. 1, pp. 46-53, 2006

[16] S. Itzkovitz, T. Tlusty, U. Alon, ”Coding limits on the number of transcription factors”, BMC Genomics, Vol. 7, 2006.

[17] H. Jegou, C. Guillemot, ”Robust multiplexed codes for compression of heterogeneous data,” IEEE Transactions on Information Theory, vol. 51, no. 4, pp. 1393-1407, Apr. 2005.

[18] H. de Jong, Modeling and simulation of genetic regulatory systems: a literature review. J Comput Biol, vol. 9, no. 1, pp. 67103, 2002.

[19] S. Kauftinan, Adaptation on Rugged Fimess Landscapes, in Lectwzs in the Science of Complexity, SFI Studies in the Science of Complexity, Lecture Volume I, ed. D. Stein, pp. 527-618, Addison-Wesley, Redwood City, 1989.

[20] M. Levine, R. Tjian, ”Transcription regulation and animal diversity,” Nature, Vol. 424, pp. 147-151, 2003.

[21] L. S. Liebovitch, Y. Tao, A. Todorov, and L. Levine, Is there an error correcting code in DNA? Biophys. J., 71: 15391544, 1996.

[22] S. Linn, R. Stephen-Lloyd, R. Roberts, Nucleases, Cold Spring Harbor Laboratory Press, 1993.

[23] D. A. 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, 33:433455, October 2003.

[24] E. E. May, M. A. Vouk, D. L. Bitzer, and D. I. Rosnick, Coding model for translation in E. coli K-12, Paper presented at the First Joint Conference of EMBS-BMES, Atlanta, GA, 1999. 49.

[25] E. E. May, M. A. Vouk, D. L. Bitzer, and D. I. Rosnick, An error-correcting code framework for genetic sequence analysis, J. Franklin Inst., 341: 89109, 2004

[26] E. E. May, M. A. Vouk, D. L. Bitzer, and D. I. Rosnick, Coding theory based models for protein translation initiation in prokaryotic organisms, BioSystems, 76: 249260, 2004.

[27] E. E. May, ”Error Control Codes and the Genome”, in: M. Akay, ”Genomics and proteomics engineering in medicine and biology”, Wiley-IEEE Press, 2007.

[28] O. Milenkovic, B. Vasic, ”Information Theory and Coding Problems in Genetics”, ITW 2004, San Antonio, Texas, October 24.29,2004

[29] L. E. Orgel, ”Adding to the genetic alphabet”, Nature, pp. 18-20, 1990.

[30] G. L. Rosen, J. D. Moore, ”Investigation of Coding Structures in DNA”, Proc. IEEE int. conf. Acoustics, Speech, and Signal Processing, 2003, vol. 2, pp. 361-364

[31] G. L. Rosen, ”Examining Coding Structure and Redundancy in DNA”, IEEE engineering in medicine and biology magazine, 2006, pp. 62-68

[32] Amir Hesam Salavati, Masih Nilchian, Mahnoosh Alizadeh, Saeid Bagheri, Mohammad Javad Emadi, Hadi Kiapour, Mehdi Sadeghi, Mohammad Reza Aref, Kaveh Kavousi, Mehdi Pakdaman, Genome Alphabet: Are the Letters of Genome Language the Four Nucleotides or a Combination of Them?, To be submitted to the Journal of Theoretical Biology

[33] D.C. Schmidt and E. E. May, Visualizing ECC properties of E. coli K-12 translation initiation sites, Paper presented at the Workshop on Genomic Signal Processing and Statistics, Baltimore, MD, 2004.

[34] G. Sicot, R. Pyndiah, ”Study on the Genetic Code: Comparison with Multiplexed Codes”, Proc. IEEE International Symposium on Information Theory (ISIT), pp. 2666-2670, 2007.

[35] H. Yockey, ”Information Theory and Molecular Biology”, Cambridge University Press, New York, 1992.