On Designing DNA Codes and their Applications

by

DIXITA LIMBACHIYA
201221014

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

March, 2019

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Doctor of Philosophy at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Dixita Limbachiya

Certificate

This is to certify that the thesis work entitled ON DESIGNING DNA CODES AND THEIR APPLICATIONS has been carried out by DIXITA LIMBACHIYA for the degree of Doctor of Philosophy at *Dhirubhai Ambani Institute of Information and Communication Technology* under my supervision.

Prof Manish K. Gupta
Thesis Supervisor

Acknowledgments

First, I want to thank my Ph.D. advisor and a Guru in a true sense, Prof. Manish Kumar Gupta. It is my privilege to be his Ph.D. student. I am thankful to him to introduce me with this new branch of *DNA Computing* and cross-linked field of biomathematics, which I was not aware before joining his group. His love for mathematics has helped me to explore the rich lands of research. He has taught me, both consciously and unconsciously, how to be an independent researcher and thriving in the academic profession. I admire all his contributions in terms of time, knowledge, kindness and ideas to make my Ph.D. experience wonderful. I will never forget his efforts that have put in me to cultivate different skills beside research, which are helpful for an academic career. He gave me an opportunity to work on various projects at Gupta Lab. He has also helped me in improving my writing and presentation skills. He has been available for rigorous discussion during all the tough times which pulled out best from me. I express my gratitude towards him for being an excellent example he has provided as a successful interdisciplinary researcher with best entrepreneur skills. He has been a beautiful human being which never fails to help his students and motivate them from every corner. Although these words would not justify my gratitude towards him, I sincerely thank him for his constant support and for his motivation, immense knowledge and patience.

Besides my advisor, I would like to thank my Ph.D. synopsis and research progress seminar (RPS) committee: Prof. Sanjay Srivastava, Prof. Bhaskar Chaudhury, Prof. Manish Narwariya, Prof. Gagan Garg for their critical comments and constructive suggestions which enhanced my work. I am also thankful to other faculty members of DA-IICT who has been my course instructors who contributed

significantly in developing my teaching skills and cooperative teamwork.

My sincere thanks goes to my coauthors Krishna Gopal Benerjee, Bansari Rao, Vaneet Aggarwal and Shalin Shah who provided me with an opportunity to work jointly in a team. Without their precious support, it would not be possible to conduct this valuable research. Especially, Krishna Gopal apart from being a wonderful friend has helped in dealing with problems related to mathematics. He has also been the moral support which helped in personally and professionally. I would also extend my gratitude to all undergraduate students Madhav Khakkar, Shikhar Gupta, Foram Joshi, Amay Aggarwal and others at Gupta Lab. It was an immense pleasure and enjoyable experience to work with all of them on different innovative projects.

My journey of Ph.D. was cheerful due to a wonderful group of people around me which made my stay at DA-IICT enjoyable and unforgettable. I extend love and gratitude to all my friends. I will always be thankful to Dr. Trupti Padiya who inspired me for giving my best.

My sincere thanks also goes to DA-IICT who provided me with an opportunity to pursue my Ph.D. and for financial support to present my work at various conferences. I am grateful to DA-IICT for giving me access to laboratory and research facilities. I respect all the efforts of the staff members of DA-IICT for providing smooth administrative and technical support especially Mrs. Deepa Poduval for helping me in informal ways.

Lastly and most importantly, I must thank my family. My parents and brothers, sisters whose blessing has given me immense strength towards my destination. My feeling for them cannot be expressed in words. My husband who is a fantastic friend, partner and guide who has not only supported me but pushed me to achieve the best in me. I owe my Ph.D. to him as without his love, support and motivation I could not have made this.

At last, I bow down to Almighty for giving me all strength and intelligence in fulfilling this dream.

Dixita Limbachiya

Contents

Abstract

Bio-computing uses the complexes of biomolecules such as DNA (Deoxyribonucleic acid), RNA (Ribonucleic acid) and proteins to perform the computational processes for encoding and processing the data. In 1994, L. Adleman introduced the field of DNA computing by solving an instance of the Hamiltonian path problem using the bunch of DNA sequences and biotechnology lab methods. An idea of DNA hybridization was used to perform this experiment. DNA hybridization is a backbone for any computation using the DNA sequences. However, it is also cause of errors. To use the DNA for computing, a specific set of the DNA sequences (DNA codes) which satisfies particular properties (DNA codes constraints) that avoid cross-hybridization are designed to perform a particular task. Contributions of this dissertation can be broadly divided into two parts as 1) Designing the DNA codes by using algebraic coding theory. 2) Codes for DNA data storage systems to encode the data in the DNA.

The main research objective in designing the DNA codes over the quaternary alphabets {A, C, G, T}, is to find the largest possible set of M codewords each of length n such that they are at least at the distance d and satisfies the desired constraints which are feasible with respect to practical implementation. In the literature, various computational and theoretical approaches have been used to design a set of DNA codes which are sufficiently dissimilar. Furthermore, DNA codes are constructed using coding theoretic approaches using fields and rings. In this dissertation, one such approach is used to generate the DNA codes from the ring $R = \mathbb{Z}_4 + w\mathbb{Z}_4$, where $w^2 = 2 + 2w$. Some of the algebraic properties of the ring R are explored. In order to define an isometry from the elements of the ring R to DNA, a new distance called Gau distance is defined. The Gau distance motivates

the distance preserving map called Gau map ϕ. Linear and closure properties of the Gau map are obtained. General conditions on the generator matrix over the ring R to satisfy reverse and reverse complement constraints on the DNA code are derived. Using this map, several new classes of the DNA codes which satisfies the Hamming distance, reverse and reverse complement constraints are given. The families of the DNA codes via the Simplex type codes, first order and r^{th} order Reed-Muller type codes and Octa type codes are developed. Some of the general results on the generator matrix to satisfy the reverse and reverse complement constraints are given. Some of the constructed DNA codes are optimal with respect to the bounds on M, the size of the code.

These DNA codes can be used for a myriad of applications, one of which is data storage. DNA is stable, robust and reliable. Theoretically, it is estimated that one gram of DNA can store 455 EB (1 Exabyte = 10^{18} bytes). These properties make the DNA a potential candidate for data storage. However, there are various practical constraints for the DNA data storage system. In this work, we construct DNA codes with some of the DNA constraints to design efficient codes to store data in DNA.

One of the practical constraints in designing DNA codes for storage is the repeated bases (runlengths) of the same DNA nucleotides. Hence, it is essential that each DNA codeword should avoid long runlengths. In this thesis, codes are proposed for data storage that will dis-allow runlengths of any base to develop DNA data storage error-free codes.

A fixed GC-weight u (the occurrence of G and C nucleotides in a DNA codeword) is another requirement for DNA codewords used in DNA storage. DNA codewords with large GC-weight lead to insertion and deletion (indel) errors in DNA reading and amplification process thus, it is crucial to consider a fixed GC-weight for DNA code.

In this work, we propose methods that generate families of codes for the DNA data storage systems that satisfy no-runlength and fixed GC-weight constraints for the DNA codewords used for data storage. The first is the constrained codes which use the quaternary code and the second is DNA Golay subcodes that use

the ternary encoding.

The constrained quaternary coding is presented to generate DNA codes for the data storage. We give a construction algorithm for finding families of DNA codes with the no-runlength and fixed GC-weight constraints. The number of DNA codewords of fixed GC-weight with the no-runlength constraint is enumerated. We note that the prior work only gave bounds on the number of such codewords while in this work we count the number of these DNA codewords exactly. We observe that the bound mentioned in the previous work does not take into account the distance of the code which is essential for data reliability. Thus, we consider distance to obtain a lower bound on the number of codewords along with the fixed GC-weight and no-runlength constraints.

In the second method, we demonstrate the Golay subcode method to encode the data in a variable chunk architecture of the DNA using ternary encoding. N. Goldman *et al.* introduced the first proof of concept of the DNA data storage in 2013 by encoding the data without using error correction in the DNA which motivated us to implement this method. While implementing this method, a bottleneck of this approach was identified which limited the amount of data that can be encoded due to fix length chunk architecture used for data encoding. In this work, we propose a modified scheme using a non-linear family of ternary codes based on the Golay subcode that includes flexible length chunk architecture for data encoding in DNA. By using the Golay ternary subcode, two substitution errors can be corrected.

In a nutshell, the significant contributions of this thesis are designing DNA codes with specific constraints. First, DNA codes from the ring using algebraic coding by defining a new type of distance (Gau distance) and map (Gau map) are proposed. These DNA codes satisfy reverse, reverse complement and complement with the minimum Hamming distance constraints. Several families of these DNA codes and their properties are studied. Second, DNA codes using constrained coding and Golay subcode method are developed that satisfy no-runlength and GC-weight constraints for a DNA data storage system.

List of Principal Symbols

$\mathbb{F}_q$	Field with q symbols
R	Finite Ring $\mathbb{Z}_4 + w\mathbb{Z}_4$, where $w^2 = 2 + 2w$
$\mathscr{C}$	Code over the Ring R
$\mathscr{C}_{DNA}$	DNA code
$\mathbf{x}$	Codeword
n	Length of a codeword
M	Size of a code
d_{Gau}	Minimum Gau distance
ϕ	Gau Map
d_H	Minimum Hamming distance
i, j	Indexes
$\mathbf{x}^r$	Reverse of a codeword $\mathbf{x} \in \mathscr{C}_{DNA}$
$\mathbf{x}^c$	Complement of a codeword $\mathbf{x} \in \mathscr{C}_{DNA}$
$\mathbf{x}^{rc}$	Reverse Complement of a codeword $\mathbf{x} \in \mathscr{C}_{DNA}$
u	GC-weight of a DNA codeword
$A_4^{RC}(n, d)$	Maximum size of reversible-complement DNA code
$A_4^{R}(n, d)$	Maximum size of reversible DNA code

$A_4^{GC}(n,d,u)$ Maximum size of DNA code with GC-weight u

$A_4^{RC,GC}(n,d,u)$ Maximum size of reversible-complement code with GC-weight u

$A_4^{R,GC}(n,d,u)$ Maximum size of reversible code with GC-weight u

$< G >$ Row span of the matrix G over the ring R

R_1 Finite Ring $\mathbb{Z}_2 + w_1\mathbb{Z}_4$, where $w_1^2 = 2w_1$

R_2 Finite Ring $\mathbb{Z}_4 + w_2\mathbb{Z}_2$, where $w_2^2 = 2$

R_3 Finite Ring $\mathbb{Z}_2 + w_3\mathbb{Z}_2$, where $w_3^2 = 0$

$< \mathbf{x}, \mathbf{y} >$ Inner product of $\mathbf{x}$ and $\mathbf{y}$

$\mathscr{C}^{\perp}$ Dual of code $\mathscr{C}$

σ_i The number of occurrence of $i \in R$ in a codeword

S_k^{α} Type α Simplex type code over the ring R

S_k^{β} Type β Simplex type code over the ring R

$\mathcal{R}(r,m)$ Reed-Muller type Code of order r and length m

ψ Map to convert ternary symbols to DNA alphabets

List of Tables

List of Figures

CHAPTER 1

Introduction

If you don't work on important problems, it's not likely that you'll do important work.

-Richard Hamming,

You and Your Research, Bell Communications Research Colloquium Seminar, 7 March 1986 [36]

1.1 Bio-molecular Computing: Introduction

Computation and biology have attained incredible breakthroughs by developing the Bio-Nano things [6] where these devices perform the computation that is possible due to the technological paradigm shift in communication, coding and information theory. Information Theory studies the amount of information that can be transmitted and stored through a channel and coding theory deals with correcting the errors during communication of the data through a noisy channel. Mathematics has been used powerfully to solve problems in engineering, communication theory and recently in biological science at a large scale giving rise to the field of mathematical biology and biological mathematics [68]. In particular, modeling the biological systems uses knowledge of biological mathematics to understand the behavior of a complex system involving different biological entities.

Bio-molecular computing is a branch of biological mathematics that includes the study of the biological molecules to perform a computation. The biocomputing employs biological molecules such as DNA (Deoxyribonucleic Acid), RNA

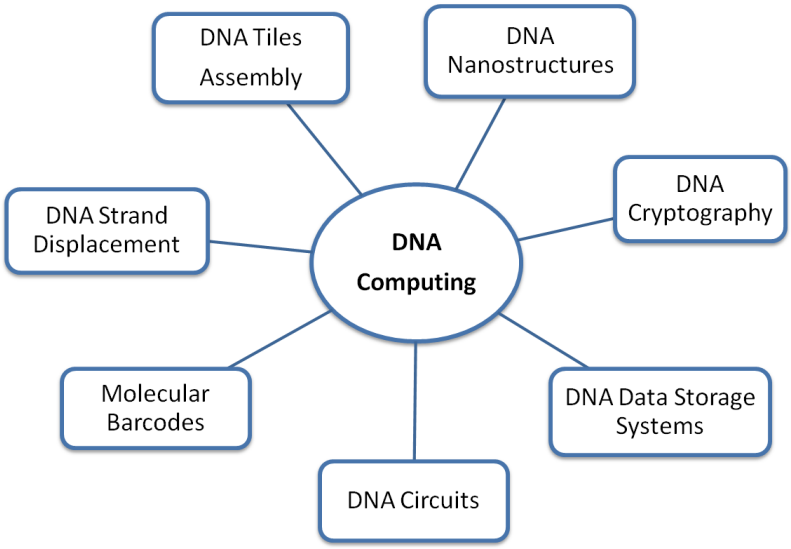

Figure 1.1: Different directions of DNA computing.

(Ribonucleic Acid) and proteins (peptides) for computation. From all of these natural media, a DNA had been the most promising candidate for the genetic manipulations and experimental studies. Tom Head in 1987 [60] proposed the idea of computing using the DNA, however, in 1994 L. Adleman [3] used the DNA sequences to solve a popular instance of the Hamiltonian path problem and pioneered the field of DNA computing. Subsequently Erik Winfree [135] has shown that self-assembly of DNA is Turing universal which has paved the way to all the current innovative directions of DNA computing [116] as shown in Figure 1.1.

The identification of mathematical properties of the DNA sequences [87], [113] inspired many researchers to explore this new amalgamation of biology and coding theory [55] to study the computational aspects of DNA.

1.2 DNA: A Computing Material

DNA computing includes computation using the DNA sequences. DNA is a double-stranded molecule formed by the pairing of four basic building units A-(Adenine), C-(Cytosine), G-(Guanine), T-(Thymine) which are called nucleotides. The DNA sequence is held by the hydrogen bond which connects the Watson-Crick complementary bases with each other denoted by $A^c = T$ and $G^c = C$

Figure 1.2: DNA structure with its four nucleotides A-Adenine, G-Guanine, C-Cytosine and T-Thymine. These are the basic building unit of DNA which forms a double helical structure via hydrogen bonds. Each DNA base is paired with the complementary bases such that A (red band) is connected to its complementary base T (green band) while C (yellow band) is paired with G (brown band).

(see Figure 1.2). An alternating chain of sugar and phosphate forms the backbone of DNA. Due to its properties such as robustness, high density and its self-replicating property, DNA is an ideal source of computing [30]. All the necessary instructions of the human life are stored in the human genome of size around 3 billion base pairs. DNA can be stable in extreme conditions and can survive in an adverse environment. DNA is dense, as 1 gram of it can store about 455 exabytes (1 EB $= 10^{18}$ bytes) of digital information [24]. DNA can replicate in every form of life accurately, and it can be amplified in a lab using Polymerase Chain Reaction (PCR). All these properties of DNA makes it suitable for computation.

Necessary steps of DNA computation are: i.) The first step is to encode the input data to DNA sequences ii.) Next, performing the molecular computation using DNA operations iii.) Decoding the encoded DNA sequences. The first step is executed using data encoding techniques to encode the data to DNA sequences. Once the data is encoded into DNA, DNA sequences are synthesized using DNA synthesis. DNA synthesis is producing DNA naturally or artificially. It is a regular practice to synthesize DNA sequences using different DNA synthesis platforms (explained briefly in Chapter 2). DNA can be read and decoded by using DNA sequencing protocols (described in Chapter 2).

The core of the DNA computing is the DNA hybridization; however, it provokes errors in DNA computation. During computation, unwanted DNA base

pairing leads to errors because DNA sequences may not bind with its perfect complementary sequence and forms mismatch pairs (See Figure 1.3). The set of DNA sequences in the solution, instead of reacting to the other sequence, it gets a fold and forms different loops and bulges. These structures are called secondary structures. During the computation, in an inappropriate condition, the DNA sequences form secondary structures which interfere with the subsequent computational steps and introduce errors in the computation. Thus, the DNA computing success depends on a designing the DNA sequences that avoid errors during computation.

These errors can be controlled by designing the DNA sequences using error correcting codes [86]. A set of these DNA sequences (also known as DNA codewords) is called a DNA code [87]. For DNA computing, a specific set of the DNA codes satisfying particular properties (called DNA constraints) that avoid cross-hybridization are designed to perform a particular task. Sufficiently dissimilar set of DNA codes have been constructed by using different approaches in the literature [83].

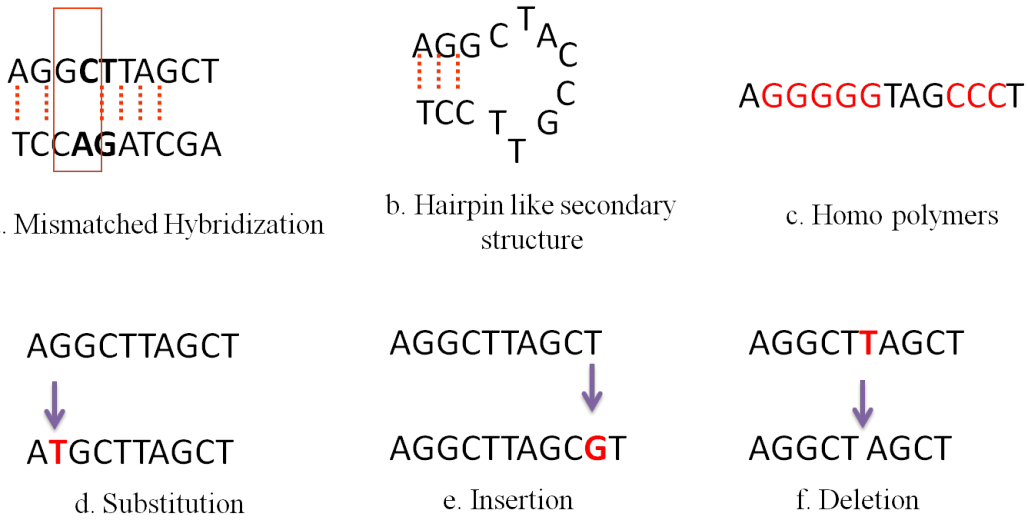

Figure 1.3: Types of errors in designing of DNA code are: (a.) Mis-hybridization of two sequences of DNA in which one of the bases have not perfectly paired with the other sequence. (b.) It is the formation of secondary structure by a pairing of DNA on itself. It shows a hairpin-like structure. (c.) Repetition of long runs of same bases leads to sequencing errors. (d.) During DNA synthesis process, the DNA base is misplaced by another base. (e.) The base may get inserted or (f.) deleted during DNA reading and writing process.

An objective of DNA code design problem is to construct the largest possible set of codewords M of length n that are at least at some distance d on alphabet A, C, G, T feasible with respect to some DNA constraints. Different approaches have been used to design the DNA codewords which can be classified as computational and theoretical approaches (discussed in chapter 2.2). In the theoretical approach, coding theoretic approaches [42] have been widely used to design DNA codes by using codes over finite fields and finite rings. This dissertation focuses on one such theoretical approach to construct DNA codes. We propose families of DNA codes satisfying certain combinatorial constraints.

The data encoded DNA codewords can be used in different applications such as DNA nanostructures [108] [115], Molecular barcodes for chemical libraries [20], data encryption [65], DNA circuits [103] [75], data storage [81] and many more. From all the applications, the data storage is the most blooming application. DNA has outperformed and have emerged as the most promising candidate compared to other emerging storage technologies. As an application of DNA codes, codes for data storage are presented in this thesis.

DNA is a promising archival data storage due to its high information density [37], persistence [48] and easy maintenance. Unlike disk storage, it can store the data for about 1000 years and recover it back. DNA has self-repair and an error correction mechanism which has been witnessed by many researchers [55]. Nonetheless, the cost of writing and reading data on DNA is a major concern. However, biotechnological improvements [50] urge to desire that DNA storage will be a future data archival storage technology.

There are three necessary steps for DNA data storage: (i) A binary data is converted to DNA codewords. (ii) DNA synthesizer is used to synthesize data encoded DNA codewords (data writing on DNA). These data encoded DNA codewords are stored. (iii) DNA sequencing is used to decode these DNA codewords (reading the data from DNA). However, there are practical challenges for DNA storage. To overcome these technological challenges, data encoding schemes with error correcting codes are designed, which can detect and correct errors in DNA storage.

Most popular errors as substitution, insertion, deletion, runlengths (repeated bases) of the base occur in the DNA data storage. Out of which, in this thesis, we focus on DNA codes with the substitution error and no runlengths since they have the higher probability than other errors [37].

The main problems addressed in this thesis are to construct the DNA codes with some constraints. The first contribution includes the development of DNA codes by using the algebraic structure of the ring which satisfies the combinatorial constraints. A unique correspondence between the elements of the rings and the DNA codes by defining a new Gau map ϕ such that the distance is preserved. For this matter, we propose a new distance called the Gau distance. These results in the classes of DNA codes where some of them give optimal results. The second contribution presents ternary and quaternary encoding schemes which generate families of DNA codes with important DNA constraints for DNA data storage systems. We give a theoretical bound on DNA codewords satisfying these DNA constraints.

1.3 Background on Algebraic Structures

Coding theory and data storage systems are two different sides of the same coin. Evolution of coding theory has improved the performance of data storage devices and development in storage systems have encouraged research in coding theory. As this dissertation contributes in both areas, it is essential to study elementary aspects of these domains.

Coding theory is the mathematical cornerstone for handling errors during the process of the information transmission through a noisy channel. It deals with detection and correction of errors while information is transferred through a channel. The foundation of coding theory is to develop efficient encoding and decoding schemes for data such that a maximum number of errors can be detected and corrected. For which, error detecting and correcting codes are designed with some properties discussed later in this chapter. Codes are a combinatorial object which are constructed using a wide range of mathematical tools from binary arithmetic

to algebra.

Mainly these codes are algebraic which are constructed by using fields and rings. In earlier days, fields were widely used for designing the codes that can be linear or nonlinear. Due to its rich mathematical features, linear codes attracted many researchers. But in 1994, the breakthrough paper by Hammons *et al.,* [58] showed that the non linear families of codes can be represented as linear codes over the ring $\mathbb{Z}_4$ via Gray map that leads to the development of many exciting codes over rings. These codes have several applications in lattice designing, cryptography and combinatorial designs. Not surprisingly, it was also applied to design DNA sequences for DNA computing.

Basic definitions are given to get familiar with algebraic aspects of codes.

1.3.1 Ring and its Properties

Ring is a fundamental algebraic structure used for developing codes in coding theory.

Definition 1.1. *A non-empty set R is an algebraic structure with two binary operations addition $(+)$ and multiplication $(.)$, a set R is called a **Ring** if $\forall\ a, b, c \in R$*

1. *$a + b \in R$, $ab \in R$ (Closure).*
2. *$(a + b) + c = a + (b + c)$, $(ab)c = a(bc)$ (Associative).*
3. *$a + b = b + a$ (Commutative).*
4. *$\exists\, 0 \in R, \forall\, a \in R, a + 0 = 0 + a = a$ (Additive Identity), $\exists\, 1 \in R, \forall a \in R, a1 = a$ (Multiplicative Identity).*
5. *$\forall\, a \in R, \exists - a \in R, a + (-a) = (-a) + a = 0$ (Additive inverse).*
6. *$a(b + c) = ab + ac, (a + b).c = a.c + b.c$ (Distributive).*

Definition 1.2. ***Commutative Ring**: A ring R is commutative if the multiplication is commutative. That is, if $ab = ba\ \forall\ a, b \in R$.*

Example 1.1. *For the ring $\mathbb{Z}_4$ given in Example 1.2, $ab = ba\ \forall\ a, b \in \mathbb{Z}_4$. Hence it is a commutative ring.*

Through out this thesis, our rings will be commutative rings with unity.

Definition 1.3. ***Finite Ring****: A ring with the finite number of elements is called finite ring.*

Example 1.2. *The set $\mathbb{Z}_4 = \{0, 1, 2, 3\}$ is a ring of integers modulo 4. $\mathbb{Z}_4$ is a finite ring with four elements.*

Definition 1.4. ***Zero Divisor****: A non zero element $a \in R$ is called a zero divisor if there exists a non-zero element $b \in R$ such that $ab = 0$ in R.*

Example 1.3. *For the ring $\mathbb{Z}_4$ given in Example 1.2, 2 is a zero divisor. Since $2.2 = 0$*

Definition 1.5. ***Unit****: For the ring R with unity $1(\neq 0)$, $a \in R$, is the unit element, if $\exists\, b \in R$, such that $ab = ba = 1$.*

Example 1.4. *For the ring $\mathbb{Z}_4$ given in Example 1.2, 1 and 3 are units.*

Definition 1.6. ***Ring with Unity****: If the ring R has a multiplicative identity, then R is called a ring with unity denoted by 1.*

Example 1.5. *For the ring $\mathbb{Z}_4$ given in Example 1.2, $1 \in \mathbb{Z}_4$ is a unity.*

Definition 1.7. ***Ideal of the Ring****: Any non empty subset I of R is called ideal if (i) $a + b \in I\ \forall a, b \in R$ (ii) $a.g$ (left), $g.a$ (right) $\in I$, where $a \in R, g \in I$.*

Definition 1.8. ***Generator of an Ideal****: If $g \in R$ generates the Ideal I then g is called a generator of an ideal I. We denote the generator of an Ideal I by $\big[g\big]$.*

Example 1.6. *For the ring $\mathbb{Z}_4$ given in Example 1.2, Ideal $I = \{0, 2\} \subset R$ is left and right ideal hence, it is ideal of the ring R generated by $\big[2\big]$.*

Definition 1.9. ***Proper Ideal****: A proper ideal I is an ideal of the ring R such that I is a proper subset of R and $I \neq R$.*

Definition 1.10. ***Maximal Ideal****: A proper ideal I is called a maximal ideal if there exists no other proper ideal J with I a proper subset of J.*

Example 1.7. *For the ring $\mathbb{Z}_4$ given in the Example 1.2, Ideal $I = \{0, 2\} \subset R$ is a maximal ideal of the ring R.*

Definition 1.11. ***Local Ring****: A ring R is a local ring if it has a unique maximal ideal* μ.

Remark 1.8. *For the local ring R with a unique maximal ideal* μ, $R \setminus \mu$ *forms the set of units of R.*

Note that the ring $\mathbb{Z}_4$ given in Example 1.2 is a local ring with one maximal ideal.

Example 1.9. *The ring* $\mathbb{Z}_{12}$ *is not a local ring as it has two maximal ideal* $\big[2\big] = \{0, 2, 4, 6, 8, 10\}$ *and* $\big[3\big] = \{0, 3, 6, 9\}$.

Definition 1.12. ***Principal Ideal****: An ideal I generated by a single element* $g \in R$ *is called principal ideal.*

Example 1.10. *For the ring* $\mathbb{Z}_4$ *given in Example 1.2, Ideal* $I = \{0, 2\} \subset R$ *is principal ideal.*

Definition 1.13. ***Principal Ring****: If all ideals of the ring are principal, then the ring is a principal ring.*

Definition 1.14. ***Chain Ring****: If ideals of a ring can be linearly ordered by inclusion (arranged in ascending order such that minimal ideal contains in maximal ideal) then it is said to be a chain ring.*

Definition 1.15. ***Sub Ring****: A non-empty subset S of a ring R is a sub ring of R if S is itself a ring with the operations of R.*

Definition 1.16. ***Field****: A Field* $\mathbb{F}$ *is a nonempty set of elements with two binary operations addition* $(+)$ *and multiplication* $(.)$ *satisfying the following axioms. For all* $a, b, c \in \mathbb{F}$*: (i)* $\mathbb{F}$ *is closed under* $+$ *and . i.e.* $a + b \in \mathbb{F}$ *and* $a.b \in \mathbb{F}$ $\forall$ $a, b \in \mathbb{F}$.
(ii) $a + b = b + a, a.b = b.a$ $\forall$ $a, b \in \mathbb{F}$. *(Commutative).*
(iii) $(a + b) + c = a + (b + c)$, $a.(b.c) = (a.b).c$. *(Associative).*
(iv) $a.(b + c) = a.b + a.c$, $(a + b).c) = a.c + b.c$ *(Distributive).*
(v) $a + 0 = a$ $\forall a \in \mathbb{F}$ *(Additive Identity).*
(vi) For any $a \in \mathbb{F}$, $\exists$ $-a \in \mathbb{F}$ *an additive inverse of a.*
(vii) $a.1 = a$ *and* $a.0 = 0$ $\forall a \in F$ *(Multiplicative Identity).*
(viii) For any $a \neq 0 \in \mathbb{F}$, $\exists$ $a^{-1} \in \mathbb{F}$ *a multiplicative inverse of a such that* $a.a^{-1} = a^{-1}.a = 1$.

Example 1.11. *A set $\mathbb{Z}_2 = \{0, 1\}$ with two binary operations addition $(+)$ and multiplication $(.)$ modulo 2 is a field with 2 elements.*

Definition 1.17 (Linear Codes over the field)**.** *A linear code C of length n and dimension k over the finite field $\mathbb{F}_q$ with q elements is called (n, k) linear code over $\mathbb{F}_q$.*

The main objective of coding theory is to design codes of given length and size such that each element of code has maximal pairwise distance. There are different types of distance studied in the literature, but the most common is the Hamming distance. The Hamming distance between two codewords $\mathbf{x}$ and $\mathbf{y}$ is the number of places in which two codewords differ from one another and it is denoted by $d_H(\mathbf{x}, \mathbf{y})$.

Definition 1.18 (Codes over the Ring)**.** *Let R be the ring. Any subset $\mathscr{C}$ of R^n is called a code over the ring R.*

Definition 1.19 (Linear Code over the Ring)**.** *Any R-submodule $\mathscr{C}$ of R^n is called a linear code over the ring R.*

For a given ring R, let a code $\mathscr{C}$ with the length n, size M and the minimum Hamming distance be denoted by $\mathscr{C}$ (n, M, d_H). The elements of code $\mathscr{C}$ are called codewords..

Definition 1.20 (Generator Matrix G)**.** *For a linear code $\mathscr{C}(n, M, d_H)$ over the ring R, a generator matrix G of $\mathscr{C}$ is a $k \times n$ matrix G such that the rows of G are linearly independent.*

1.4 Structure of the Thesis

The thesis is divided into seven chapters which are described in Figure 1.4. Chapters 1 and 2 include the introduction and preliminaries on DNA editing methods and DNA codes, respectively. First, along with the introduction, a short summary on algebraic structure is given in Chapter 1 to get familiar with the terminologies used in later chapters. In the next chapter a brief background on DNA manipulation techniques as DNA synthesis, sequencing and mathematical prospects of

DNA codes and related constraints are given in Chapter 2. As the research areas are flourishing and paving new ways to different technologies, it is important to have an extended survey. Chapter 2 and 5 describe reviews on DNA codes and DNA data storage systems, respectively. Chapters 3, 4 and 6 are the main contributions of this thesis. Chapter 7 concludes the contributions of this dissertation and summarize results. It also discusses the future scope.

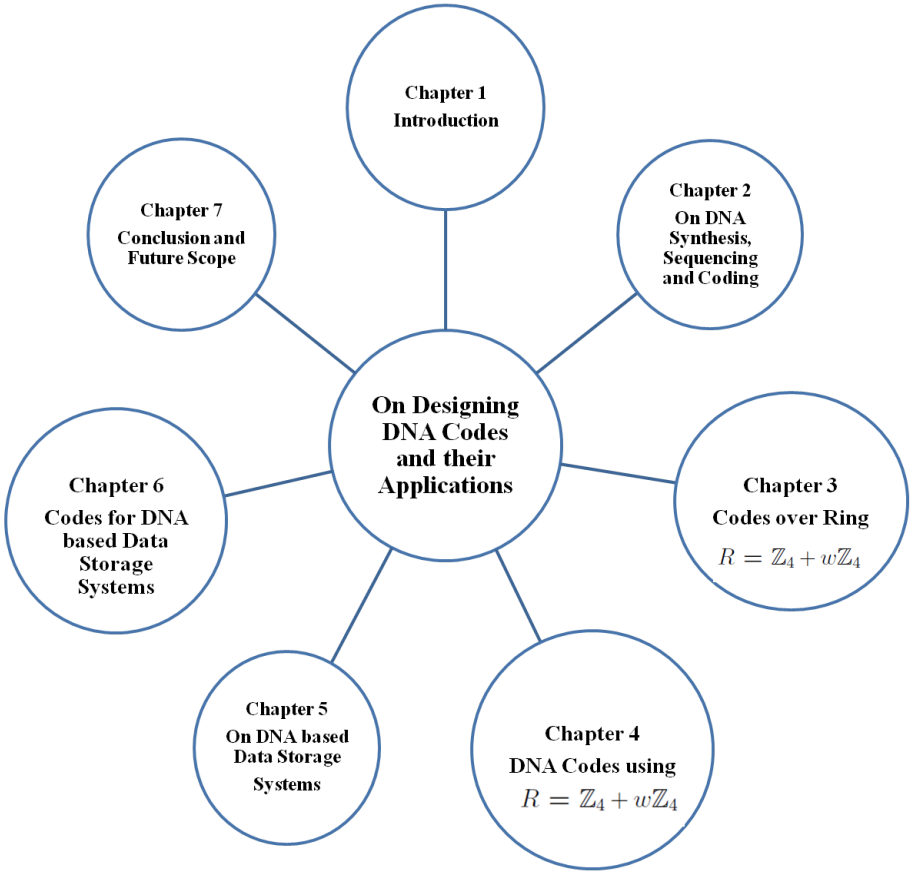

Figure 1.4: Thesis Outline

1.5 Thesis Contributions

This dissertation has the following contributions:

Chapter 3 includes the introduction to the codes over the ring $R = \mathbb{Z}_4 + w\mathbb{Z}_4 = \{a + bw : a, b \in \mathbb{Z}_4 \text{ and } w^2 = 2 + 2w\}$. It discusses the orthogonal properties of codes over the ring R. Rings of the ring R are also introduced.

Chapter 4 gives the constructions of the DNA codes via the ring R [80]. We de-

fine a Gau distance d_{Gau} on the ring R that motivated us to propose a new distance preserving map ϕ called the Gau map from the elements of the ring to the DNA nucleotides. The properties of the Gau map are discussed including the linearity and closure under the reverse and reverse complement constraints. Families of the DNA codes are developed via the Simplex type codes, Reed-Muller type codes (first order and r^{th} order) and Octa type codes. These DNA codes satisfy the minimum Hamming distance, reverse and reverse complement constraints. The DNA codes designed from the Reed-Muller type codes attain the better size of the codes compared to that exist in the literature for some values of n and d. For an instance, the DNA code $\mathscr{C}_{DNA}(n = 16, M = 8192, d_H = 4)$ is better than $\mathscr{C}_{DNA}(n = 16, M = 28, d_H = 4)$ [92]. Also the DNA code $\mathscr{C}_{DNA}(n = 8, M = 64, d_H = 4)$ is better than $\mathscr{C}_{DNA}(n = 8, M = 16, d_H = 4)$ [78]. One can observe that the Reed-Muller type codes attains the lower bound for $A_4^{GC}(n, d, u)$ and $A_4^{RC,GC}(n, d, u)$ [73] on the DNA code for some values of n, d_H and $u = n/2$. For the DNA code with $n = 8$, $d_H = 4$ and $u = 4$, the lower bound obtained for $A_4^{RC,GC}(8, 4)$ is *224* which is greater than the lower bound observed in [22] as 128. Some of the general results for the conditions on reversible and reversible-complement DNA codes are derived. We also obtain some results of the DNA codes constructed from the rings.

Chapter 6 proposes DNA codes with two additional constraints for DNA data storage. The first constraint is no-runlength, and the other is fixed GC-weight. For a designing of error-free codes for DNA data storage, it is essential to study the source of occurrence of errors. The long runs of the same nucleotides (sometimes also called as homopolymer runs) is one of the primary origins of errors in DNA storage [37]. Hence, it is vital that each DNA codeword should avoid the occurrence of repeated bases (runlengths). Such DNA codewords are called DNA codes with no-runlength constraints.

In particular, repeated bases in DNA may be read as a single base which may result in the missing of reading these repeated bases during DNA synthesis and sequencing. For example, in the DNA codeword $ACCCCTACAGTA$, C is repeated. The DNA sequencer may read the repeated runs of C's as a single base.

Hence, the long runs of bases lead to an increment of the dropout rates and decrement in the DNA read coverage [114] during DNA sequencing. In this chapter, DNA codes that satisfy no-runlength constraint for data storage are proposed.

The second necessity for DNA codewords is to have fixed GC-weight u (the number of positions in a DNA codeword where G and C are present) [87]. For example, $TCGCAGCCT$ is a DNA codeword with GC-weight $u = 6$. A higher GC-weight triggers errors like insertion and deletion of the base in the DNA codewords in the polymerase chain reaction (PCR). It also leads to the low DNA base coverage in DNA sequencing. DNA codes with fixed GC-weight (ranging from $50\% - 60\%$) are more stable than DNA codes with higher GC-weight (greater than 60%). Thus, it is significant to consider DNA codes with a fixed GC-weight.

Next in Chapter 6, we present a constraint based and Golay subcode methods for a DNA archival data storage which avoids runs of nucleotides and has fixed GC-weight u [82]. First, an altruistic algorithm which generates DNA codewords with the above constraints is given. It is further used to find the bound on the number of DNA codewords and with a guarantee on the minimum distance between the codewords for better error correction for DNA codes with these constraints.

The second method proposed in Chapter 6 is using a Golay subcode to encode the data in the DNA [79]. In this method, the non-linear family of the DNA subcodes is developed, and a DNA code with length $n = 11$ and $d_H = 5$ is used to encode the data in the DNA sequences. By using this approach, an overall length of the data encoded DNA is reduced by avoiding the redundancy of the data. A DNA chunk architecture with variable length is proposed which resulted in achieving better DNA data storage capacity while the Goldman's approach [47] used fixed length DNA chunk architecture. The proposed method allows 2 bit flips error correction in the DNA data storage. The scalability and code rate of the Golay subcode method is determined. It achieves the theoretical DNA net information density 115 EB (1 Exabyte = 10^{18} bytes) per gram of DNA with the length $n = 11$.

CHAPTER 2

DNA Synthesis, Sequencing and Coding

As usual, nature's imagination far surpasses our own, as we have seen from the other theories which are subtle and deep.

-Richard P. Feynman

In The Character of Physical Law (1965, 2001) [36]

This chapter introduces DNA manipulation techniques and preliminaries on DNA codes which are essential for understanding the contributions of this dissertation. The first section describes preliminaries on DNA wetlab techniques that includes groundwork for DNA sequencing and synthesis that is necessary for interpreting DNA storage systems. The later gives an introduction to DNA codes which includes definition on DNA constraints with relevant examples.

2.1 DNA Synthesis and Sequencing

DNA synthesis is the process of manufacturing DNA using different synthesis platforms. DNA sequencing is a process of reading the order of DNA bases in a given DNA sequence. There are different methods and platforms for DNA synthesis and sequencing. In this section, a brief introduction to DNA manipulation techniques is given. Terminologies from molecular biology, biotechnology and related to DNA storage system are described briefly. For more details on molecular biology, DNA synthesis and sequencing, a reader is referred to [106]. Schematic

representation of a DNA is presented in Figure 1.2 given in the introduction chapter.

2.1.1 Basic Terminologies of DNA

1. Nucleotides: Basic building blocks of DNA which consist of a nitrogen base, sugar and phosphate.

2. Codon: DNA of length three which codes for amino acids that are building blocks of proteins. For example, AUG is a start codon which triggers protein synthesis. There are 64 codons which code for 20 amino acids.

3. Oligonucleotides: A short DNA sequence typically of 15-20 length.

4. DNA hybridization: A process of forming double-stranded DNA (dsd) via hydrogen bonds between complementary bases.

5. DNA Melting (Denaturation): At a melting temperature, double-stranded DNA sequence segregates into single sequences by breaking the hydrogen bonds.

6. Annealing: It is a process of binding two single-stranded DNA (ssDNA). For instance, primer binding to ssDNA.

7. DNA Amplification: The process of copying DNA sequences by using techniques outside the living cells.

8. Polymerase Chain Reaction (PCR): It is a temperature dependent machine used to amplify the DNA sequence using primers, DNA polymerase and nucleotides which includes three basic steps: denaturation (at 90-96 degree Celsius), annealing (at 55-65 degree Celsius) and extension (at 72 degree Celsius). The extension is to repeat the cycles of PCR. For a given DNA sequence and for l number of cycles it will generate 2^l amount of DNA.

9. Primers: A short DNA sequence used for DNA amplification in PCR and DNA sequencing. Generally, it is of length 18-20 nucleotides.

10. Gel Electrophoresis: A process of segregation of DNA sequences with respect to the length of the DNA. DNA sample is loaded on the gel and the basis of the length of DNA, it travels across the pores of the agar gel to the positive end of the electrode as the DNA is negatively charged. The phosphate group on the DNA is a reason for its negative charge.

11. Sequencing Coverage/Depth: Number of times each base is read during the sequencing. For instance, 15x coverage represents that each nucleotide is read for at least 15 times.

2.1.2 DNA Synthesis

DNA can be synthesized naturally or by using the chemical. In nature, DNA is replicated in cells by with the help of DNA polymerase enzyme using a template sequence. This enzyme reads the DNA and adds bases from one end to other end making a copy of template sequences. Another method to replicate DNA is by using molecular biology techniques such as PCR and chemicals. The most popular method is chemical synthesis (also called Denovo synthesis) of artificial DNA by adding each nucleotide by using synthesis by sequencing [41]. Designing oligonucleotides using phosphoramidite chemistries is the most robust which allowed scalable oligonucleotides synthesis. Phosphoramidite-based synthesis of oligonucleotides consists of four basic steps which are: (a.) Deprotection, (b.) Coupling, (c.) Capping (d.) Oxidation. Each base is added to growing oligonucleotides sequences attached on the solid support. Once the DNA is synthesized, it is detached from the solid substrate. DNA sequences are synthesized using column based synthesizers or microarray-based synthesizers [74]. A column based synthesizer can typically synthesize DNA of 100 nt approximately with error rates less than 0.5% and have efficient productivity. However, it decreases with the increase in the length of the oligonucleotide. The typical cost of a DNA base range from $\$0.05 - \$0.17$ depending on the length of synthetic DNA and the service provider which synthesize the DNA [74]. In the microarray method, a series of distinct probes are attached on microarray chips to which the growing DNA base is synthesized using light-activated chemistries. It is cost effective compared to

column base method. Nevertheless, the custom array, a chip-based technology designed by Agilent and CombiMatrix [45] has revolutionized the synthesis because these technologies are cheaper ($\$0.00001 - \$0.0001$ per nt), accurate and facilitate multiplexing for large-scale synthesis that was not possible with column based method.

Chemical synthesis of synthetic DNA is required for different applications like for using as a probe for DNA hybridization detection in some molecular biology techniques, synthesizing parts of a gene fragment, to induce a mutation artificially, to study effect of mutation on gene, also for designing primer and adapter sequences for PCR, DNA sequencing and plasmid transformation. Modern day application of a chemical DNA synthesis is DNA based data storage. Data encoded in DNA sequences are synthesized artificially and stored. However, synthesis of small DNA sequences (50-150 bps) is done smoothly and efficiently. While the synthesis of longer DNA sequences (>250 bps) is challenging, therefore in DNA data storage, smaller DNA sequences are used to encode data as they are less error-prone. The significant drawbacks of these synthesis technologies are the cost associated with the synthesis methods and less yield. They are also time-consuming and less efficient. These limitation hinders the DNA storage to be a regular storage medium. Nevertheless, researchers are working on technological innovations to facilitate this feat which will allow large-scale and low-cost production of synthetic DNA.

2.1.3 DNA Sequencing

DNA sequencing is a process to identify the specific position of the DNA base in the sequence. The recent explosion in biological data has urged the sustainable improvement in DNA sequencing methods, cost and productivity. DNA synthesis by sequencing has evolved as the first generation sequencing (whole genome shotgun sequencing), next-generation sequencing (NGS high throughput sequencing) [126] and the third generation of sequencing (single molecule long read [19] and nanopore sequencing [134]). Since the Human Genome was sequenced in 2001 [133], DNA reading/writing technologies have shown extraor-

dinary progress in analyzing the complex genetic architecture. Some of these technologies focus on specific goals. A brief introduction to these methods is given in this section.

2.1.3.1 First Generation Sequencing

The first method invented by Sanger in 1977 was based on the Sanger chain termination method such that an inhibitor terminates a DNA sequence extension at specific points [118]. The method includes amplification of each DNA sequence by cloning it in E.*coli* bacteria. The cloned sequences are amplified by using PCR primers during which a fluorescent-labeled dideoxyribonucleotide (ddNTP), corresponding to the nucleotide identified at the terminal position is added. These DNA sequences are separated by using a Polyacrylamide gel and the fluorescent labels are read through emission spectrum which depicts the particular nucleotide at the particular position. This method can typically read $400 - 100$ bps with a 99.99 % accuracy. Merits of this approach are that it produces high-quality DNA sequences and reads longer length DNA sequences while demerits are higher cost and lower productivity.

A decade later this process was automated by the invention of capillary electrophoresis based machines which allowed sequencing of DNA with Kilobases (kb) in a single run. Later, in order to read the sequences with the length greater than 1 Kb, a shotgun sequencing method was adapted that determined the order of DNA sequence by reading the overlapping ends of cloned DNA fragments [88]. However, the assembly of these cloned DNA fragments for large-scale DNA sequencing is challenging. Thus, a second generation method using pyrosequencing which involves identifying the DNA bases in determining the production of pyrophosphate [95]. Pyrosequencing was then commercialized by Roche to produce large-scale parallel DNA sequencing facility in a single run which gave birth to next-generation sequencing [112].

2.1.3.2 Next Generation Sequencing

Next Generation Sequencing (NGS) is massive parallel sequencing [123] which avoids using electrophoresis, laborious preparation of DNA clones and produces

large scale high throughput data, that were limited in previous sequencing methods. A large number of NGS platforms such as IIlumina, SOLiD emerged by using the concept of bridge amplification [101] which allow efficient (error rates less than 1%) and high yield DNA sequencing [111]. NGS reads short as well as long length DNA for different applications. Short reads are of low cost and give high accuracy while long reads are costly, with high error rates and low efficiency. Later, other sequencing methods like single molecule sequencing (SMS) [59] and single-molecule real-time (SMRT) sequencing [35] that reads a single molecule avoiding the requirement of bridge amplification were discovered. However, the most promising and recently emerging DNA sequencing [29] is by using nanopore developed by Oxford Nanopore Technologies [25].

Nanopore sequencing platforms [66] is the most exciting field on interest which infers the order of DNA by measuring the electric current intensity varying with the length of DNA when the molecule is passed through a nanochannel [28]. It does not need fluorescent labeling which enables cheaper and rapid DNA sequencing. A first portable, pen drive sized DNA sequencer called MinION [84] was released by Oxford Nanopore Technologies which paves the way for developing on-spot DNA reading and writing machines. Such High throughput DNA sequencing has revolutionized the process of DNA reading and thus it is anticipated that the advancements in DNA reading machines will aid in achieving the commercial success of DNA data storage systems [137].

2.2 On DNA Codes

In this section, we give an introduction to DNA codes and its constraints. We describe methods to construct DNA codes. In general, there are two methods to design DNA codes which are computational and theoretical approaches. We give a survey on different rings which are used to construct DNA codes.

2.2.1 Introduction to DNA Codes

DNA code design deals with a designing of the DNA codes that satisfies the set of constraints. There are different constraints (discussed in Section 2.3) proposed in the literature for the DNA codes.

In order to use DNA sequences for DNA computation, they must be error-free [69]. There are two sources of errors in DNA computation which are mis-hybridization and secondary structure formation like the formation of bulges, loops and hairpin structures occurs, that will hinder the computation process (shown in Figure 1.3). There are some types of common errors like insertion, deletion, substitution of a DNA base. During the DNA synthesis and sequencing process, DNA base may get substituted by another base just like bit-flip errors. Also, DNA base may get inserted or deleted in the reading and writing process of DNA. Together, insertion and deletion errors are called indel errors. By taking those DNA codewords that are distinct in at least d places, these types of errors can be prevented.

Definition 2.1. *Let $\Sigma_{DNA} = \{A, G, T, C\}$. The set of M distinct DNA sequences each of length n such that the Hamming distance d_H between any two DNA sequences is at least d_H is called DNA code.*

We denote minimum Hamming distance $d_H = \min\{d_H(\mathbf{x}, \mathbf{y}) : \mathbf{x} \neq \mathbf{y}, \forall\, \mathbf{x}, \mathbf{y} \in \mathscr{C}_{DNA}\}$.

Mathematically, we denote the DNA code as $\mathscr{C}_{DNA}(n, M, d_H) \subseteq \Sigma_{DNA}^n = \{A, T, G, C\}^n$.

Example 2.1. *$\mathscr{C}_{DNA} = \{AGTC, GAAG, GAGA, AGAG, AGGA, AGCT, GACT, TCAG, TCGA, TCCT, GATC, CTCT, CTTC, TCTC, CTAG, CTGA\}$ is a DNA code $\mathscr{C}_{DNA}$ with parameters $n = 4$, $M = 16$ and $d_H = 2$.*

Generally, the Hamming distance between the DNA codewords defines the dissimilarity measure used for the DNA codewords. However, to ignore the cross-hybridization between the DNA codewords, it is essential to consider a distance between the given DNA codewords, its reverse and reverse complement DNA

codewords. The notion of distance between DNA codewords propels researchers to define the combinatorial constraints on DNA codes.

For a given DNA codeword $\mathbf{x} = (x_1\ x_2\ \ldots\ x_{n-1}\ x_n)$, the reverse of the DNA codeword is defined as $\mathbf{x}^r = (x_n\ x_{n-1}\ \ldots\ x_2\ x_1)$, the Watson - Crick complement or simply complement of the DNA codeword is defined as $\mathbf{x}^c = (x_1^c\ x_2^c\ \ldots\ x_{n-1}^c\ x_n^c)$, the reverse complement of the DNA codeword is defined as $\mathbf{x}^{rc} = (x_n^c\ x_{n-1}^c\ \ldots\ x_2^c\ x_1^c)$, where $x_i \in \{A, C, G, T\}$ for each $i = 1, 2, \ldots, n$ and $A^c = T$, $T^c = A$, $G^c = C$, $C^c = G$. For example, if $\mathbf{x} = ATCT$ then $\mathbf{x}^r = TCTA$, $\mathbf{x}^{rc} = AGAT$. It ensures that the DNA hybridizes with the perfect complementary base and thus, prevent errors in the computation [87, 73].

In the next section, we discuss combinatorial constraints of DNA codes.

2.3 Constraints on DNA Codes

DNA constraints are categorized as combinatorial constraints and thermodynamic constraint constraints. In this thesis, we consider combinatorial constraints for the construction of the DNA codes. Combinatorial constraints are related to distance between DNA codes, its reverse and reverse complement sequence. These constraints prevent mis-hybridization that may lead to errors in the computation.

1. *Hamming distance constraint:* The Hamming distance constraint for a DNA code $\mathscr{C}_{DNA}$ is that $d_H(\mathbf{x}, \mathbf{y}) \geq d_H$, $\forall\ \mathbf{x}, \mathbf{y} \in \mathscr{C}_{DNA}$ with $\mathbf{x} \neq \mathbf{y}$, for a given minimum Hamming distance d_H.

2. *Reverse constraint:* The reverse constraint for a DNA code is that $d_H(\mathbf{x}, \mathbf{y}^r) \geq d_H$, $\forall\ \mathbf{x}, \mathbf{y} \in \mathscr{C}_{DNA}$ (may be $\mathbf{x} = \mathbf{y}$). Let $\mathbf{x} = \mathbf{y} = AACC, \mathbf{y}^r = CCAA$ then $d_H(\mathbf{x}, \mathbf{y}^r) = 4$

   Definition 2.2 (Reversible Code)**.** *A DNA code $\mathscr{C}_{DNA}$ is reversible if for all $\mathbf{x} \in \mathscr{C}_{DNA}, \mathbf{x}^r \in \mathscr{C}_{DNA}$.*

   Example 2.2. *In Table 2.1, the DNA code $\mathscr{C}_{DNA}(n = 4, M = 16, d_H = 2)$ is a reversible code. Note that $AAAA, ATTA, TTTT, GTTG, CCCC, CAAC, GGGG, TCCT$ are self reversible.*

AACC	TTGG	AAAA	ATTA	GGTT	CCAA	TTTT	GTTG
GTCA	ACTG	CCCC	CAAC	TGAC	CAGT	GGGG	TCCT

Table 2.1: Example of Reversible DNA Code

3. *Complement constraint:* The complement constraint is that $d_H(\mathbf{x}, \mathbf{y}^c) \geq d_H$, $\forall\ \mathbf{x}, \mathbf{y}^c \in \mathscr{C}_{DNA}$ (sometime $\mathbf{x} = \mathbf{y}$). Let $\mathbf{x} = \mathbf{y} = AACC, \mathbf{y}^c = TTGG$ then $d_H(\mathbf{x}, \mathbf{y}^c) = 4$.

   Definition 2.3 (Complement Code)**.** *A DNA code $\mathscr{C}_{DNA}$ is complement if for all $\mathbf{x} \in \mathscr{C}_{DNA}, \mathbf{x}^c \in \mathscr{C}_{DNA}$.*

   Example 2.3. *A DNA code $\mathscr{C}_{DNA}(n = 4, M = 8, d_H = 2)$ $\{AACC, TTGG, AAAA, CCCC, ATGC, TACG, TTTT, GGGG\}$ is complement code.*

   Remark 2.4. *Note that for complement $\mathscr{C}_{DNA}$ DNA code, $\mathbf{x} \neq \mathbf{x}^c$.*

4. *Reverse complement constraint:* The reverse complement constraint for a DNA code is that $d_H(\mathbf{x}, \mathbf{y}^{rc}) \geq d_H$, $\forall\ \mathbf{x}, \mathbf{y}^{rc} \in \mathscr{C}_{DNA}$. Note that $\mathbf{x}$ may be equal to $\mathbf{y}$. Let $\mathbf{x} = \mathbf{y} = AACC, \mathbf{y}^{rc} = GGTT$ then $d_H(\mathbf{x}, \mathbf{y}^{rc}) = 4$.

   Definition 2.4 (Reversible-Complement Code)**.** *A DNA code $\mathscr{C}_{DNA}$ is reversible-complement code if for all $\mathbf{x} \in \mathscr{C}_{DNA}, \mathbf{x}^{rc} \in \mathscr{C}_{DNA}$.*

   Example 2.5. *In Table 2.2, DNA code $\mathscr{C}_{DNA}(n = 4, M = 20, d_H = 2)$ is reversible-complement code.*
   Note that $ACGT, TGCA, GATC, CTAG, GTAC, CATG, AGCT, TCGA$ are self reversible-complement.

AACC	TTGG	AAAA	ACGT	TGCA
GGTT	GTCA	ACTG	CCCC	GTAC
CATG	TGAC	CCAA	TTTT	GATC
CTAG	CAGT	GGGG	AGCT	TCGA

Table 2.2: Example of Reversible-Complement DNA Code

5. *GC-weight constraint:* For a DNA code, if the total occurrence of Gs and Cs nucleotides in each codeword is constant integer u then the DNA code

$\mathscr{C}_{DNA}(n, M, d_H, u)$ satisfies the u GC-weight constraint. Generally, for a DNA code $\mathscr{C}_{DNA}$, the GC-weight is $\lfloor n/2 \rfloor$.

Example 2.6. *In Table 2.3, DNA code $\mathscr{C}_{DNA}(n = 4, M = 16, d_H = 2, u = 2)$ is a DNA code with fixed GC-weight $u = 2$*

AGAG	AGGA	AGCT	AGTC	GAAG	GAGA	GACT	GATC
CTAG	CTGA	CTCT	CTTC	TCAG	TCGA	TCCT	TCTC

Table 2.3: Example of DNA Code with fixed GC-weight $u = 2$.

6. No-Runlength constraint: For a DNA code $\mathscr{C}(n, M, d_H)$, no-runlength constraint implies that no two consecutive elements in a codeword are the same ie. For $\mathbf{x} \in \mathscr{C}_{DNA}(n, M, d_H)$, $\mathrm{x}_i \neq \mathrm{x}_{i+1}$. These kind of DNA codes are also denoted as forbidden sequences in the literature.

   Example 2.7. *In Table 2.4, DNA code $\mathscr{C}_{DNA}(n = 4, M = 12, d_H = 2, u = 2)$ is a DNA code with no repeated nucleotides and with a fixed GC-weight $u = 2$*

AGAG	AGCT	AGTC	GAGA	GACT	GATC
CTAG	CTGA	CTCT	TCAG	TCGA	TCTC

Table 2.4: Example of DNA code with fixed GC-weight $u = 2$ and no-runlength constraints.

Following associations between the reversible, complement and reversible-complement codes are obvious.

- Reversible-Complement $\not\Rightarrow$ Reversible

  Example 2.8. *If $\mathscr{C}_{DNA} = \{AAAA, AAGG, CCTT, TTTT\}$ then $\forall\, \boldsymbol{x} \in \mathscr{C}_{DNA}, \mathbf{x}^{rc} \in \mathscr{C}_{DNA}$ but $\mathbf{x}^{r} \notin \mathscr{C}_{DNA}$ i.e. $GGAA, TTCC \notin \mathscr{C}_{DNA}$.*

- Reversible-Complement $\not\Rightarrow$ Complement.

  Example 2.9. *If $\mathscr{C}_{DNA} = \{AAAA, AAGG, CCTT, TTTT\}$ then $\forall\, \boldsymbol{x} \in \mathscr{C}_{DNA}, \mathbf{x}^{rc} \in \mathscr{C}_{DNA}$ but $\mathbf{x}^{c} \notin \mathscr{C}_{DNA}$ i.e. $TTCC, GGAA \notin \mathscr{C}_{DNA}$.*

- Reversible $\not\Rightarrow$ Reversible-Complement

Example 2.10. *If* $\mathscr{C}_{DNA} = \{AAAA, AAGG, GGAA, TTTT\}$ *then* $\forall\, \boldsymbol{x} \in \mathscr{C}_{DNA}, \mathbf{x}^r \in \mathscr{C}_{DNA}$ *but* $\mathbf{x}^{rc} \notin \mathscr{C}_{DNA}$ *i.e.* $CCTT, TTCC \notin \mathscr{C}_{DNA}$.

- Complement $\nRightarrow$ Reversible-Complement

  Example 2.11. *If* $\mathscr{C}_{DNA} = \{AAAA, AAGG, TTCC, TTTT\}$ *then* $\forall\, \boldsymbol{x} \in \mathscr{C}_{DNA}, \mathbf{x}^c \in \mathscr{C}_{DNA}$ *but* $\mathbf{x}^{rc} \notin \mathscr{C}_{DNA}$ *i.e.* $CCTT, GGAA \notin \mathscr{C}_{DNA}$.

- Reversible and Complement $\Rightarrow$ Reversible-Complement

- Reversible and Reversible-Complement $\Rightarrow$ Complement

- Complement and Reversible-Complement $\Rightarrow$ Reversible

  Example 2.12. *If* $\mathscr{C}_{DNA} = \{AAAA, AAGG, CCTT, TTTT, GGAA, TTCC\}$ *then* $\forall\, \boldsymbol{x} \in \mathscr{C}_{DNA}, \mathbf{x}^r$ *and* $\mathbf{x}^c \in \mathscr{C}_{DNA} \implies \forall\, \boldsymbol{x} \in \mathscr{C}_{DNA}, \mathbf{x}^{rc} \in \mathscr{C}_{DNA}$.

We will utilize these interconnections to construct reversible and reversible- complement code. For any given $\mathbf{x}, \mathbf{y} \in \mathscr{C}_{DNA}$, note that $d_H(\mathbf{x}^r, \mathbf{y}^c) = d_H(\mathbf{x}^{rc}, \mathbf{y}) = d_H(\mathbf{x}, \mathbf{y}^{rc}) = d_H(\mathbf{x}^c, \mathbf{y}^r)$.

It is desirable to have a DNA code to be either reversible, complement or reversible-complement. The DNA code given in Example 2.1 is reversible, complement and reversible-complement.

Given a length n and a distance d, the main DNA coding problem is to construct a DNA code $\mathscr{C}_{DNA}$ (either reversible, complement or reversible-complement) that has maximum codewords, which motivates to define these functions on the size of the code. For a given length n and a distance d, $A_4^{RC}(n, d)$ denotes the maximum size of reversible-complement DNA code. Similarly, $A_4^{R}(n, d)$ denotes the maximum size of reversible DNA code. The maximum size of DNA code with fix GC-weight u in each codeword is indicated by $A_4^{GC}(n, d, u)$. For the each DNA codeword with a fix GC-weight u in each codeword, reversible and reversible-complement code is denoted by $A_4^{R,GC}(n, d, u)$ and $A_4^{RC,GC}(n, d, u)$ respectively. Different bounds have been studied on these functions [87] using various approaches for construction of DNA codes.

2.4 Approaches for the Construction of DNA Codes

Many researchers introduced different approaches for the construction of the DNA codes with finite length n, defined distance d and set of constraints with respect to the application [83, 130]. The set of constraints essential for the DNA codes is subject to the application. There are few attempts made to design the DNA sequences using groups [120] and graphs [99, 129]. The construction of the DNA codes can obtain an optimal DNA code in a way that every codeword in the set follows a maximum number of constraints for a large value of n and large minimum distance d with minimum errors in DNA computation. In the next section, computational and theoretical approaches are discussed.

2.4.1 Computational Approaches

Coding theory has been fortunate to use several computational approaches [71] for the constructions of the record-breaking codes thus it was natural to use similar approaches for the construction of the DNA codes. Many such approaches such as Tabu search, greedy method, stochastic local search and genetic algorithms, seed building [91], clique search, hybrid search [105], greedy [12], Variable neighborhood search (VNS), Lexicographic Approach, Simulated Annealing Approach, Stochastic Local Search Approach [131] have been used to develop DNA codes. Although DNA codes constructed using computational approach achieved bounds, but they have high computational complexity and hence does not allow to construct codes for higher lengths. Therefore, theoretical methods are used which gives flexibility for developing DNA codes with higher lengths and distance.

2.4.2 Theoretical Approaches

The theoretical approaches have received much attention. Theoretical constructions like algebraic coding theory [42], algebraic number theory [62] and formal language [60] have been used in the literature to develop DNA codes.

Out of which, algebraic coding theory approach has been widely used by many researchers for constructing DNA codes from different fields and rings. Construction through this approach has achieved the lower bounds on a different set of constraints. For constructing the DNA codes, a mapping is defined in such a way that all the elements of fields and rings are in one-to-one correspondence with the elements of DNA of the certain lengths.

2.4.2.1 DNA Codes from Finite Fields

Linear codes over the field $GF(4)$ with four elements $\{0,\ 1,\ \omega, \omega^2\}$ are directly mapped to the DNA elements {A, C, G, T} respectively to construct linear DNA codes [42, 132]. These DNA codes have improved the lower bounds on GC-weight constraints with the length of DNA code $n \leq 30$. By using the field $GF(4)$, nonlinear cyclic codes BCH type DNA codes were studied in [38]. In [94], N. Aboluion used the computer algebra systems to construct the DNA codes satisfying GC-weight and Hamming distance constraints. DNA codes of higher lengths $4 \leq n \leq 30$ were obtained using additive codes over $GF(4)$. Bounds on the size of DNA codes satisfying these constraints were improved by shortening and puncturing of DNA codes. Using the linear and additive codes of odd lengths over $GF(4)$, DNA codes satisfying the Hamming distance and reverse complement constraints were generated by defining an alternative mapping as mentioned in [1]. The mapping used for DNA codes is $0, \omega, \overline{\omega}, 1$ to $\{A, C, G, T\}$ respectively with $\omega^2 + \omega + 1 = 0$ [2].

In [98], the concept of lifted polynomial over $\mathbb{F}_4$ was used that generated the reversible codes of odd length from $\mathbb{F}_{16}$. To construct DNA codes of an even length, special kind of mapping that preserves Hamming distance and reverse constraints was defined. For example, the element corresponding to GC is mapped to the fourth power of the element of $\mathbb{F}_{16}$ viz. $\alpha^2 \rightarrow$ GC then $(\alpha^2)^4 \rightarrow$ CG. DNA codes using $\mathbb{F}_{16}$ using lifted polynomial were also studied in [98].

Although codes over fields pioneered to develop DNA codes by algebraic coding concept, researchers explored their interest in using different ring structures to construct good DNA codes with rich algebraic properties.

2.4.2.2 DNA Codes from Rings

The advancement of DNA codes from an algebraic coding has driven the interest of coding theorist for developing the DNA codes using finite rings [83]. In 2001, V.V Rykov *et al.* developed the reversible cyclic DNA codes using the quaternary alphabets in [117]. However, it considered only the reverse constraint for the DNA codes. P. Gaborit and O. D. King in the year 2005, constructed the linear DNA codes from the ring $\mathbb{Z}_4$ [42] that satisfies reverse and reverse complement constraints. Researchers explored notion of cyclic codes to DNA codes which resulted into series of interesting constructions of DNA codes.

I. Siap *et al.* proposed an idea of using cyclic codes from the finite ring $\mathbb{F}_2 + u\mathbb{F}_2$ with similarity measures (a special kind of distance similarity) instead of the Hamming distance [124]. The cyclic DNA codes from the similar ring $\mathbb{F}_2 + u\mathbb{F}_2$, where $u^2 = 1$ based on the deletion distance was also introduced by I. Siap *et al.* in [125]. By using the ring, $\mathbb{F}_2 + u\mathbb{F}_2$ with $u^2 = 0$, Liang and Wang constructed the DNA cyclic codes of an even length in [78]. K. Guenda and T.A. Gulliver studied the DNA cyclic irreversible odd length codes of the type Simplex, BCH codes and Reed-Muller type codes from the ring $\mathbb{F}_2 + u\mathbb{F}_2$ in [52]. Odd length DNA codes were given which satisfies the Hamming distance constraints from the commutative ring $\mathbb{F}_2[u]/ < u^4 - 1 >$ with $u^4 = 1$ in [53].

A ring $\mathbb{Z}_4 + u\mathbb{Z}_4$, where $u^2 = 0$ with 16 elements was introduced by B. Yildiz and S. Karadeniz in [138] and DNA codes of odd lengths from the ring were studied in [100]. Bayram *et al.* [11] constructed DNA codes using skew constacyclic over the ring $\mathbb{F}_4 + v\mathbb{F}_4$. A direct map between 64 elements of the ring to 64 DNA codons (three nucleotides) were given in [13] over the ring $R = \mathbb{F}_2[u]/(u^6)$. The ring $\mathbb{F}_4[u]/ < u^2 + 1 >$ with $u^2 = 1$ was used for DNA codes of the length 6 in [85]. Reversible-complement DNA codes were developed Srinivasulu B and M. Bhaintwal using the ring $\mathbb{F}_4 + u\mathbb{F}_4$, $u^2 = 0$ in [127] by defining the gray map form the ring to $\mathbb{F}_4^2$. Using the similar approach, very recently, cyclic DNA codes of odd length were studied from the commutative ring $\mathbb{F}_2 + u\mathbb{F}_2 + v\mathbb{F}_2 + uv\mathbb{F}_2 + v^2\mathbb{F}_2 + uv^2\mathbb{F}_2$, where $u^2 = 0, v^3 = v, uv = vu$ [33] in which they presented correspon-

dence between elements of the ring to DNA codon by using a gray map from the ring to $\mathbb{F}_2 + u\mathbb{F}_2$, where $u^2 = 0$.

For the DNA codes designing, the reversible DNA codes from the skew cyclic codes, a non-commutative rings were used. Generalized non-chain ring $\mathbb{F}_{16} + u\mathbb{F}_{16} + v\mathbb{F}_{16} + uv\mathbb{F}_{16}$ was used to construct the reversible DNA codes in [56], [57]. In the past year, Oztas *et al.* developed the reversible DNA codes using the ring $\mathbb{F}_2[u]/(u^{2k} - 1)$ [97] by using a novel concept of coterm polynomials.

The properties of the ring $R = \mathbb{Z}_4 + w\mathbb{Z}_4$ was studied by Choie and Dougherty for the first time in [23]. Using the structure of the ring, recently the DNA cyclic codes of odd lengths from the ring $\mathbb{Z}_4 + w\mathbb{Z}_4$, where $w^2 = 2$ was proposed in [31].

These results motivate us to investigate the structure of the ring $R = \mathbb{Z}_4 + w\mathbb{Z}_4$ from [23] and use it to construct the DNA codes of even lengths. In Chapter 3, the ring $\mathbb{Z}_4 + w\mathbb{Z}_4$, where $w^2 = 2 + 2w$ with 16 elements is proposed. A correspondence between the ring elements and the DNA codewords of length 2 is defined via a distance preserving Gau map ϕ (described in Chapter 4). We present a new type of distance called the Gau distance on the ring R. Several new families of the DNA codes are obtained which satisfies the Hamming distance, reverse and reverse complement constraints. For each family of code, the generator matrix over the ring R is defined and DNA codes are developed by using defined mapping.

CHAPTER 3

Codes over the Ring $\mathbb{Z}_4 + w\mathbb{Z}_4$

The art of doing mathematics consists in finding that special case which contains all the germs of generality.

-David Hilbert

Bulletin of the American Mathematical Society (Jan 1966), 72, No. 1, Part 2, 65. [36]

This chapter introduces the structure of the ring $R = \mathbb{Z}_4 + w\mathbb{Z}_4$, where $w^2 = 2 + 2w$. Some of the properties of codes over the ring R are given. It also includes discussion on rings R_1, R_2 and R_3 of the ring R. It presents some of the results on the orthogonality of codes over the ring and derives its orthogonal properties.

The ring $R = \mathbb{Z}_4 + w\mathbb{Z}_4 = \{a + bw : a, b \in \mathbb{Z}_4 \text{ and } w^2 = 2 + 2w\}$ is a finite ring with 16 number of elements. It is a commutative ring. For the ring R, the set of zero divisors is $\{a + bw : a \in 2\mathbb{Z}_4, b \in \mathbb{Z}_4\} = \{0, 2, w, 2 + w, 2w, 2 + 2w, 3w, 2 + 3w\}$ and the set of units is $\{a + bw : a \in 2\mathbb{Z}_4 + 1, b \in \mathbb{Z}_4\} = \{1, 3, 1 + w, 3 + w, 1 + 2w, 3 + 2w, 1 + 3w, 3 + 3w\}$. The ring R has 5 distinct ideals as follows.

$$[0] = \{0\}$$

$$[2w] = \{0, 2w\}$$

$$[2] = \{0, 2, 2w, 2+2w\} = [2+2w]$$

$$[w] = \{0, 2, w, 2+w, 2w, 2+2w, 3w, 2+3w\} = [2+w] = [3w] = [2+3w]$$

$$[1] = R = [3] = [1+w] = [3+w] = [1+2w] = [3+2w] = [1+3w] = [3+3w]$$

Note that the ring is a chain ring because ideals are linearly ordered as $[0] \subset [2w] \subset [2] \subset [w] \subset R$. The ring R is a local ring as it has one maximal ideal generated by $w, 2+w, 3w$ or $2+3w$. Observe that all ideals are principal ideals hence, the ring is a principal ring.

The standard form of the generator matrix G of the linear code $\mathscr{C}$ over the ring $R = \mathbb{Z}_4 + w\mathbb{Z}_4$ is given by:

$$G = \begin{pmatrix} I_{k_0} & A_{0,1} & A_{0,2} & A_{0,3} & A_{0,4} \\ 0 & wI_{k_1} & wA_{1,2} & wA_{1,3} & wA_{1,4} \\ 0 & 0 & 2I_{k_2} & 2A_{2,3} & 2A_{2,4} \\ 0 & 0 & 0 & 2wI_{k_3} & 2wA_{3,4} \end{pmatrix}, \tag{3.1}$$

where the matrices $A_{i,j}$ are defined over the ring R for $0 \leq i < j \leq 4$. A generator matrix defined in this form for the code $\mathscr{C}$ is of type $\{k_0, k_1, k_2, k_3\}$ and the code has $16^{k_0}8^{k_1}4^{k_2}2^{k_3}$ codewords [23]. We denote the row span of the matrix G on the ring R by $< G >_R$.

Codes over rings have been discussed in various papers [34] and are applied to different applications in coding theory and communication. In this work, we develop codes over the ring R and showcase some of its properties. In order to study the behavior of the ring with respect to set of zero divisors and units associated with DNA nucleotide pairs, rings of the ring R is explored. An interesting relation between the ring R and its rings are discussed in the next section.

3.1 Codes over Rings R_1, R_2 and R_3

Consider $R_1 = \mathbb{Z}_2 + w_1\mathbb{Z}_4 = \{a + bw_1 : a \in \mathbb{Z}_2, b \in \mathbb{Z}_4 \text{ and } w_1^2 = 2w_1\}$, $R_2 = \mathbb{Z}_4 + w_2\mathbb{Z}_2 = \{a + bw_2 : a \in \mathbb{Z}_4, b \in \mathbb{Z}_2 \text{ and } w_2^2 = 2\}$ and $R_3 = \mathbb{Z}_2 + w_3\mathbb{Z}_2 = \{a + bw_3 : a, b \in \mathbb{Z}_2 \text{ and } w_3^2 = 0\}$.

For the ring R_1, the set of zero divisors is $\{0, w_1, 2w_1, 3w_1\}$ and set of units is $\{1, 1 + w_1, 1 + 2w_1, 1 + 3w_1\}$. For the ring R_2, the set of zero divisors is $\{0, 2, w_2, 2 + w_2\}$ and set of units is $\{1, 3, 1 + w_2, 3 + w_2\}$. For the ring R_3, the zero divisors are $\{0, w_3\}$ and units are $\{1, 1 + w_3\}$.

Any subset of R_i^n $(1 \leq i \leq 3)$ is called a code over the ring R_i. Any R_i submodule of R_i^n is a linear code $\mathscr{C}$ over the ring R_i. The linear code over rings R, R_1, R_2 and R_3 are denoted by $\mathscr{C} \leq R^n$, $\mathscr{C}_{R_1} \leq R_1^n$, $\mathscr{C}_{R_2} \leq R_2^n$, $\mathscr{C}_{R_3} \leq R_3^n$ respectively [54].

The standard form of a generator matrix G_{R_1} of a linear code $\mathscr{C}_{R_1}$ over the ring $R_1 = \mathbb{Z}_2 + w_1\mathbb{Z}_4$, where $w_1^2 = 2w_1$ is

$$G_{R_1} = \begin{pmatrix} I_{k_0} & A_{0,1} & A_{0,2} & A_{0,3} \\ 0 & w_1 I_{k_1} & w_1 A_{1,2} & w_1 A_{1,3} \\ 0 & 0 & 2w_1 I_{k_2} & 2w_1 A_{2,3} \end{pmatrix}, \tag{3.2}$$

where the matrices $A_{i,j}$ are matrices over the ring R_1 for $0 \leq i < j \leq 3$. For the code $\mathscr{C}_{R_1}$, a generator matrix of type $\{k_0, k_1, k_2\}$ and the code has $8^{k_0}4^{k_1}2^{k_2}$ codewords.

The standard form of a generator matrix G_{R_2} of a linear code $\mathscr{C}_{R_2}$ over the ring $R_2 = \mathbb{Z}_4 + w_2\mathbb{Z}_2$, where $w_2^2 = 2$ is

$$G_{R_2} = \begin{pmatrix} I_{k_0} & A_{0,1} & A_{0,2} & A_{0,3} \\ 0 & w_2 I_{k_1} & w_2 A_{1,2} & w_2 A_{1,3} \\ 0 & 0 & 2I_{k_2} & 2A_{2,3} \end{pmatrix}, \tag{3.3}$$

where the matrices $A_{i,j}$ are matrices over the ring R_2 for $0 \leq i < j \leq 3$. A generator matrix of the code $\mathscr{C}_{R_2}$ is of form $\{k_0, k_1, k_2\}$ and the code has $8^{k_0}4^{k_1}2^{k_2}$ codewords.

The standard form of a generator matrix G_{R_3} of a linear code $\mathscr{C}_{R_3}$ over the ring

$R_3 = \mathbb{Z}_2 + w_3\mathbb{Z}_2$, where $w_3^2 = 0$ is

$$G_{R_3} = \begin{pmatrix} I_{k_0} & A_{0,1} & A_{0,2} \\ 0 & w_3 I_{k_1} & w_3 A_{1,2} \end{pmatrix}, \tag{3.4}$$

where the matrices $A_{i,j}$ are matrices over the ring R_3 for $0 \leq i < j \leq 2$. A code $\mathscr{C}_{R_3}$ with a generator matrix is of type $\{k_0, k_1\}$ and the code has $4^{k_0}2^{k_1}$ codewords.

Dual codes are an important class of codes widely studied for different rings. In the next section, we define orthogonal codes over the ring R, R_1, R_2 and R_3. Conditions for the existence of the orthogonality of the codes over the ring is also given.

3.2 Inner Product and Orthogonal Codes

To study the orthogonal properties of the codes over the ring R, R_1, R_2 and R_3, the inner product can be defined as $< \mathbf{x}, \mathbf{y} > = x_1y_1 + x_2y_2 + \cdots + x_ny_n$. $\mathscr{C}$ is self orthogonal if and only if $\mathscr{C} \subset \mathscr{C}^{\perp}$, where the dual code is defined as $\mathscr{C}^{\perp} = \{x \in R^n | < \mathbf{x}, \mathbf{y} > = 0 \ \forall \ y \in \mathscr{C}\}$ [138]. Moreover, $\mathscr{C}$ is self dual if and only if $\mathscr{C} = \mathscr{C}^{\perp}$ [23].

Definition 3.1. *For a codeword $\boldsymbol{x} = (x_1\ x_2 \ldots\ x_n) \in \mathscr{C}$, let σ_i denote the occurrence of $i \in R$ in a codeword $\boldsymbol{x}$.*

Example 3.1.

For $\mathbf{x} = (w\ 1+2w\ 2w\ w\ 3w\ 2w\ 0\ w)$ the occurrence of each element in $\mathbf{x}$ are $\sigma_w = 3$, $\sigma_{2w} = 2$, $\sigma_{1+2w} = \sigma_0 = \sigma_{3w} = 1$.

Theorem 3.2. *A linear code $\mathscr{C}$ over the ring R, is self-orthogonal if and only if generator matrix G of the code $\mathscr{C}$ satisfies the following.*

1. *Each row of the generator matrix G has $\sigma_w + \sigma_{2+w} + \sigma_{3w} + \sigma_{2+3w} \equiv 0 \pmod 2$ and $\sigma_1 + \sigma_3 + \sigma_{1+2w} + \sigma_{3+2w} + 2(\sigma_w + \sigma_{2+w} + \sigma_{3w} + \sigma_{2+3w}) + 3(\sigma_{1+w} + \sigma_{3+w} + \sigma_{1+3w} + \sigma_{3+3w}) \equiv 0 \pmod 4$.*

2. *Every pair of rows of the generator matrix G, is orthogonal.*

Proof. For $x \in R$,

$$x^2 = \begin{cases} 0 & x = 0, 2, 2w, 2+2w, \\ 2+2w & x = w, 3w, 2+w, 2+3w, \\ 1 & x = 1, 3, 1+2w, 3+2w, \\ 3 & x = 1+w, 3+w, 1+3w, 3+3w. \end{cases}$$

If σ_x be the occurrence of the ring element x in a vector $\mathbf{x} \in R^n$ then $< \mathbf{x}, \mathbf{x} > = 0$ if and only if $(2+2w)(\sigma_w + \sigma_{3w} + \sigma_{2+w} + \sigma_{2+3w})$+$(\sigma_1 + \sigma_3 + \sigma_{1+2w}+_{3+2w})$+$3(\sigma_{1+w} + \sigma_{3+w} + \sigma_{1+3w} + \sigma_{3+3w}) = 0 \pmod 4$. □

Example 3.3. *Let*

$$G = \begin{pmatrix} w & 2+w & 3w & 2+3w \\ 2+3w & 3w & 2+w & w \end{pmatrix}$$

be the generator matrix of the code $\mathscr{C}$ over the ring R. For $\boldsymbol{x} = (w\ 2+w\ 3w\ 2+3w)$, $\boldsymbol{y} = (2+3w\ 3w\ 2+w\ w)$, one can observe that $< \boldsymbol{x}, \boldsymbol{y} >= 0$ with $\sigma_w + \sigma_{2+w} + \sigma_{3w} + \sigma_{2+3w} = 0 \pmod 2$ and $2(\sigma_{2+3w}) + 2(\sigma_{3w}) + 2(\sigma_{2+w}) + 2(\sigma_w) = 0 \pmod 4$, where $\boldsymbol{x}, \boldsymbol{y} \in G$.

Example 3.4. *Let*

$$G = \begin{pmatrix} w & 2+w & 3w \\ 3w & 3 & w \end{pmatrix}$$

be the generator matrix of the code $\mathscr{C}$ over the ring R. For $\boldsymbol{x} = (w\ 2+w\ 3w)$, $\boldsymbol{y} = (3w\ 3\ w)$, one can observe that $< \boldsymbol{x}, \boldsymbol{y} >\neq 0$ with $\sigma_w + \sigma_{2+w} + \sigma_{3w} \neq 0 \pmod 2$ and $2(\sigma_{3w}) + \sigma_3 + 2(\sigma_3) \neq 0 \pmod 4$, where $\boldsymbol{x}, \boldsymbol{y} \in G$.

Proves of Theorems 3.8, 3.5, 3.11 are similar to the proof of Theorem 3.2.

Theorem 3.5. *A linear code $\mathscr{C}_{R_1}$ over the ring R_1 with $w_1^2 = 2w_1$ is self-orthogonal if and only if each generator matrix G_{R_1} of code $\mathscr{C}_{R_1}$ satisfies the following.*

1. *Each row of the generator matrix G_{R_1} has $\sigma_1 + \sigma_{1+w_1} + \sigma_{1+2w_1} + \sigma_{1+3w_1} \equiv 0 \pmod 2$ and $\sigma_{w_1} + \sigma_{3w_1} \equiv 0 \pmod 2$.*

2. *Every pair of rows of the generator matrix G_{R_1} is orthogonal.*

Example 3.6. *Let*

$$G_{R_1} = \begin{pmatrix} w & 3w_1 & 1+w_1 & 1+3w_1 \\ 1+3w_1 & 1+w_1 & 3w_1 & w_1 \end{pmatrix}$$

be the generator matrix of the code $\mathscr{C}_{R_1}$ over the ring R_1. For $\boldsymbol{x} = (w_1\ 3w_1\ 1+w_1\ 1+3w_1)$, $\boldsymbol{y} = (1+3w_1\ 1+w_1\ 3w_1\ w_1)$, one can observe that $< \boldsymbol{x}, \boldsymbol{y} >= 0$ with $\sigma_{w_1} + \sigma_{3w_1} + \sigma_{1+w_1} + \sigma_{1+3w_1} = 0 \pmod 2$, where $\boldsymbol{x}, \boldsymbol{y} \in G_{R_1}$.

Example 3.7. *Let*

$$G_{R_1} = \begin{pmatrix} 1 & 0 & w_1 \\ 1+w_1 & 1 & 0 \end{pmatrix}$$

be the generator matrix of the code $\mathscr{C}_{R_1}$ over the ring R_1. For $\boldsymbol{x} = (1\ 0\ w_1)$, $\boldsymbol{y} = (1+w_1\ 1\ 0)$, one can observe that $< \boldsymbol{x}, \boldsymbol{y} >\neq 0$ with $\sigma_1 + \sigma_0 + \sigma_{w_1} \neq 0 \pmod 2$ and $\sigma_{1+w_1} + \sigma_1 + \sigma_0 \neq 0 \pmod 2$, where $\boldsymbol{x}, \boldsymbol{y} \in G_{R_1}$.

Theorem 3.8. *A linear code $\mathscr{C}_{R_2}$ over the ring R_2 with $w_2^2 = 2$ is self-orthogonal if and only if each generator matrix G_{R_2} of the code $\mathscr{C}_{R_2}$ satisfies the following.*

1. *Each row of the generator matrix G_{R_2} has $\sigma_1 + \sigma_3 + 2(\sigma_{w_2} + \sigma_{2+w_2}) + 3(\sigma_{1+w_2} + \sigma_{3+w_2}) \equiv 0 \pmod 4$.*

2. *Every pair of rows of the generator matrix G_{R_2} is orthogonal.*

Example 3.9. *Let*

$$G_{R_2} = \begin{pmatrix} 1 & w_2 & 3 \\ 3 & w_2 & 1 \end{pmatrix}$$

be the generator matrix of the code $\mathscr{C}_{R_2}$ over the ring R_2. For $\boldsymbol{x} = (1\ w_2\ 3)$, $\boldsymbol{y} = (3\ w_2\ 1)$, one can observe that $< \boldsymbol{x}, \boldsymbol{y} >= 0$ with $\sigma_1 + 2(\sigma_{w_2}) + \sigma_3 = 0 \pmod 4$, where $\boldsymbol{x}, \boldsymbol{y} \in G_{R_2}$.

Example 3.10. *Let*

$$G_{R_2} = \begin{pmatrix} 1 & w_2 & 1+w_2 \\ 3 & 2+w_2 & 3+w_2 \end{pmatrix}$$

be the generator matrix of the code $\mathscr{C}_{R_2}$ over the ring R_2. For $\boldsymbol{x} = (1\ w_2\ 1+w_2)$, $\boldsymbol{y} = (3\ 2+w_2\ 3+w_2)$, one can observe that $<\boldsymbol{x},\boldsymbol{y}> \neq 0$ with $\sigma_1 + 2(\sigma_{w_2}) + 3(\sigma_{1+w_2}) \neq 0 \pmod 4$, where $\boldsymbol{x},\boldsymbol{y} \in G_{R_2}$.

Theorem 3.11. *A linear code $\mathscr{C}_{R_3}$ over the ring R_3 is self-orthogonal if and only if each generator matrix G_{R_3} of the code $\mathscr{C}_{R_3}$ satisfies the following.*

1. *Each row of the generator matrix G_{R_3} has $\sigma_1 + \sigma_{1+w_3} \equiv 0 \pmod 2$.*
2. *Every pair of rows of the generator matrix G_{R_3} is orthogonal.*

Example 3.12. *Let*

$$G_{R_3} = \begin{pmatrix} 1 & 1+w_3 & w_3 \\ 1+w_3 & 1 & w_3 \end{pmatrix}$$

be the generator matrix of the code $\mathscr{C}_{R_3}$ over the ring R_3. For $\boldsymbol{x} = (1,\ 1+w_3,\ w_3)$, $\boldsymbol{y} = (1+w_3, 1, w_3)$, one can observe that $<\boldsymbol{x},\boldsymbol{y}> = 0$ with $\sigma_1 + \sigma_{1+w_3} = 0 \pmod 2$, where $\boldsymbol{x},\boldsymbol{y} \in G_{R_3}$.

Example 3.13. *Let*

$$G_{R_3} = \begin{pmatrix} 1 & w_3 & 0 \\ 1 & 1+w_3 & w_3 \end{pmatrix}$$

be the generator matrix of the code $\mathscr{C}_{R_3}$ over the ring R_3. For $\boldsymbol{x} = (1\ w_3\ 0)$, $\boldsymbol{y}$ = $(1\ 1+w_3\ w_3)$, one can observe that $<\boldsymbol{x},\boldsymbol{y}> \neq 0$ with $\sigma_1 + \sigma_{1+w_3} \neq 0 \pmod 2$, where $\boldsymbol{x},\boldsymbol{y} \in G_{R_3}$.

Motivated from the above results, following are the observations on the self dual codes over the rings R, R_1, R_2 and R_3 [23].

Remark 3.14. *Let $\mathscr{C}$ be code over R of type $\{k_0, k_1, k_2, k_3\}$ and $\mathscr{C}^\perp$ of type $\{n-k_0-k_1-k_2-k_3, k_3, k_2, k_1\}$. If $\mathscr{C}$ is self dual, then $k_1 = k_3$ and $k_0 + k_1 + \frac{k_2}{2} = \frac{n}{2}$.*

Let $l = \{n-k_0-k_1-k_2-k_3\}$ and $w^2 = 2+2w$. Then, the generator matrix $G^\perp$ of $\mathscr{C}^\perp$ in Remark 3.14 is given as

$$G^\perp = \begin{pmatrix} B_{01}^T & B_{02}^T & B_{03}^T & B_{04}^T & I_l \\ -w(A_{01}(A_{12}A_{23}-A_{13})+A_{02}A_{23}-A_{03})^T & w(A_{12}A_{23}-A_{13})^T & -wA_{23}^T & wI_{k_3} & \mathbf{0} \\ 2(A_{01}A_{12}-A_{02})^T & -2A_{12}^T & 2I_{k_2} & \mathbf{0} & \mathbf{0} \\ -2wA_{01}^T & 2wI_{k_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

where $B_{04} = -A_{34}, B_{03} = A_{23}A_{34} - A_{24}, B_{02} = -A_{12}(A_{23}A_{34} - A_{24}) + A_{13}A_{34} - A_{14}$ and $B_{01} = A_{01}(-B_{02}) - A_{02}(A_{23}A_{34} - A_{24}) + A_{03}A_{34} - A_{04}$. Note that A_{ij} are in the form of the generator matrix in Equation 3.1.

Remark 3.15. *Let $\mathscr{C}_{R_1}$ and $\mathscr{C}_{R_2}$ be the code over R_1 and R_2 of type $\{k_0, k_1, k_2\}$, respectively. Then dual code $\mathscr{C}_{R_1}^{\perp}$ and $\mathscr{C}_{R_2}^{\perp}$ of type $\{n - k_0 - k_1 - k_2, k_2, k_1\}$ for $\mathscr{C}_{R_1}$ and $\mathscr{C}_{R_2}$, respectively. If $\mathscr{C}_{R_1}$ and $\mathscr{C}_{R_2}$ are self dual, then $k_1 = k_2$ and $k_0 + k_1 = \frac{n}{2}$.*

Let $l = n - k_0 - k_1 - k_2$ and $w_1^2 = 2w_1$. Then, the generator matrix $G_{R_1}^{\perp}$ of $\mathscr{C}_{R_1}^{\perp}$ in Remark 3.15 is given as

$$G_{R_1}^{\perp} = \begin{pmatrix} -(A_{01}(A_{12}A_{23} - A_{13}) + A_{02}A_{23} - A_{03})^T & (A_{12}A_{23} - A_{13})^T & -A_{23}^T & I_l \\ w_1(A_{01}A_{12} - A_{02})^T & -w_1A_{12}^T & w_1I_{k_2} & \mathbf{0} \\ -2w_1A_{01}^T & 2w_1I_{k_1} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

Note that the A_{ij} are in the form of the generator matrix given in Equation 3.2.

Let $l = n - k_0 - k_1 - k_2$ and $w_2^2 = 2$. Then, the generator matrix $G_{R_2}^{\perp}$of $\mathscr{C}_{R_2}^{\perp}$ in Remark 3.15 is given as

$$G_{R_2}^{\perp} = \begin{pmatrix} -(A_{01}(A_{12}A_{23} - A_{13}) + A_{02}A_{23} - A_{03})^T & (A_{12}A_{23} - A_{13})^T & -A_{23}^T & I_l \\ w_2(A_{01}A_{12} - A_{02})^T & -w_2A_{12}^T & w_2I_{k_2} & \mathbf{0} \\ -21A_{01}^T & 2I_{k_1} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

Note that the A_{ij} are in the form of the generator matrix given in Equation 3.3.

Remark 3.16. *Let $\mathscr{C}_{R_3}$ be the code over R_3 of type $\{k_0, k_1\}$ and $\mathscr{C}_{R_3}^{\perp}$ of type $\{n - k_0 - k_1, k_1\}$. If $\mathscr{C}_{R_3}$ is self dual, then $k_0 + \frac{k_1}{2} = \frac{n}{2}$.*

In the next chapter, we give a mechanism via the Gau map to construct the DNA codes from the ring R that satisfies the Hamming distance, reverse and reverse complement constraints.

CHAPTER 4

DNA Codes using $\mathbb{Z}_4 + w\mathbb{Z}_4$

Biology - DNA - is technology. It is coding. It is physical coding, but still code.

Ryan Bethencourt [18]

In this chapter, we introduce the DNA codes from the ring $R = \mathbb{Z}_4 + w\mathbb{Z}_4$, where $w^2 = 2 + 2w$. We present a distance preserving Gau map ϕ. To construct the DNA codes that satisfy the Hamming distance, reverse and reverse complement constraints, we give general conditions on the generator matrix of the code over the ring R. Some of the constructed DNA codes are optimal. To obtain the DNA codewords with constant GC weight, DNA codes on these rings are developed. Part of this chapter is published in [80] [1].

4.1 Gau Distance on the Ring $\mathbb{Z}_4 + w\mathbb{Z}_4$

A correlation between the ring R elements and the DNA alphabets is required to construct the DNA codes using the ring R. We give an isometry (distance preserving map) between the codes over the ring and the DNA codes. To define a distance on the ring R (hence eventually on R^n), the elements of the ring and the DNA alphabets can be arranged in a manner (see the Matrix $\mathscr{M}$ in Equation 4.1) such that the Hamming distance d_H between any two distinct DNA nucleotides

[1] © [2018] IEEE. Reprinted, with permission, from Dixita Limbachiya, Krishna Gopal Benerjee, Bansari Rao and Manish K Gupta, *On DNA Codes using the Ring $\mathbb{Z}_4 + w\mathbb{Z}_4$*, In Proceedings of IEEE International Symposium on Information Theory (ISIT), pp. 2401-2405, 2018

pairs in the same row or same column is 1, otherwise it is 2. This impels to define a new distance called the Gau distance on the elements of the ring R such that this property is conserved.

$$\mathscr{M} = \begin{array}{c} \\ A \\ G \\ C \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & G & C & T \end{array} \\ \begin{pmatrix} 0 & 1 & 2+3w & 3+3w \\ 3 & 2 & 1+3w & 3w \\ 2+w & 3+w & 2w & 1+2w \\ 1+w & w & 3+2w & 2+2w \end{pmatrix} \end{array} \tag{4.1}$$

For $x, y \in R$, let $x = m_{i,j} \in \mathscr{M}$, $y = m_{i^{'},j^{'}} \in \mathscr{M}$ for some $0 \leq i, j \leq 3$ and $0 \leq i^{'}, j^{'} \leq 3$ (sum of the indices is modulo 4) then, the Gau distance d_{Gau} can be defined as

$$d_{Gau}(x, y) = min\{1, i + 3i^{'}\} + min\{1, j + 3j^{'}\}. \tag{4.2}$$

Example 4.1. *For $x = 2$, $i = 1$, $j = 1$ and $y = 2 + 2w$, $i^{'} = 3$, $j^{'} = 3$, the Gau distance $d_{Gau}(2, 2 + 2w) = 2$.*

Lemma 4.2. *The Gau distance d_{Gau} is a metric on the elements of the ring R.*

Proof. For $x, y \in R$ let $x = m_{i,j} \in \mathscr{M}$, $y = m_{i^{'},j^{'}} \in \mathscr{M}$ for some $0 \leq i, j \leq 3$ and $0 \leq i^{'}, j^{'} \leq 3$, $d_{Gau}(x, y) = min\{1, i + 3i^{'}\} + min\{1, j + 3j^{'}\}$, (sum of the indices is modulo 4) satisfies the following properties:

1. If $x = y$ then $d_{Gau}(x, y) = 0$. For

$$\begin{aligned} x = y &\iff i = i^{'} \text{ and } j = j^{'} \\ &\iff i + 3i^{'} = 0 \text{ and } j + 3j^{'} = 0 \\ &\iff min\{1, i + 3i^{'}\} = 0 \text{ and } min\{1, j + 3j^{'}\} = 0 \\ &\iff min\{1, i + 3i^{'}\} + min\{1, j + 3j^{'}\} = 0 \\ &\iff d_{Gau}(x, y) = 0. \end{aligned} \tag{4.3}$$

2. For $d_{Gau}(x, y) = d_{Gau}(y, x)$, one can observe that if $i, i^{'} \in \mathbb{Z}_4$ with $i \neq i^{'}$ then $i + 3i^{'} \neq 0$ and $i^{'} + 3i \neq 0$. Therefore, $min\{1, i + 3i^{'}\} = 1$ and $min\{1, i^{'} +$

$3i\} = 1$. Hence,

$$min\{1, i + 3i^{'}\} = min\{1, i^{'} + 3i\}. \tag{4.4}$$

Similarly, for $j, j^{'} \in \mathbb{Z}_4$ with $j \neq j^{'}$, one can show that

$$min\{1, j + 3j^{'}\} = min\{1, j^{'} + 3j\}. \tag{4.5}$$

Now

$$\begin{aligned} d_{Gau}(x, y) &= min\{1, i + 3i^{'}\} + min\{1, j + 3j^{'}\} \\ &= min\{1, j + 3j^{'}\} + min\{1, i + 3i^{'}\} \\ &= min\{1, j^{'} + 3j\} + min\{1, i^{'} + 3i\} \text{ (by Equations 4.4 and 4.5)} \\ &= d_{Gau}(y, x). \end{aligned}$$

3. For $x, y, z \in R$, $d_{Gau}(x, y) \leq d_{Gau}(x, z) + d_{Gau}(y, z)$. Consider these different cases.

   - Case 1: For $x = y$, from Equation 4.3, it is clear that $0 \leq d_{Gau}(x, z) + d_{Gau}(y, z)$.
   - Case 2: For $x \neq y$, $x = z$ and $y \neq z$, $d_{Gau}(x, z) = 0$ and $d_{Gau}(x, y) = d_{Gau}(z, y)$, hence $d_{Gau}(x, y) \leq d_{Gau}(x, z) + d_{Gau}(y, z)$.
   - Case 3: If $x \neq y$, $x \neq z$ and $y \neq z$ then

     $$1 \leq d_{Gau}(x, y) \leq 2 \tag{4.6}$$

     $$1 \leq d_{Gau}(x, z) \leq 2 \text{ and } 1 \leq d_{Gau}(z, y) \leq 2 \tag{4.7}$$

     Hence, by Equations 4.6 and 4.7

     $$2 \leq d_{Gau}(x, z) + d_{Gau}(z, y) \leq 4 \tag{4.8}$$

     Therefore, by Equations 4.6, 4.7 and 4.8, it can be verified that $d_{Gau}(x, y) \leq d_{Gau}(x, z) + d_{Gau}(y, z)$.

Thus, by cases 1, 2 and 3, it is proved that d_{Gau} satisfies all the properties of distance. Hence, $d_{Gau}(x,y) = min\{1, i+3i^{'}\} + min\{1, j+3j^{'}\}$ is a distance over the ring R. □

Using the set of zero divisors $Z = \{0, 2, w, 2+w, 2w, 2+2w, 3w, 2+3w\}$ and set of units $U = \{1, 3, 1+w, 3+w, 1+2w, 3+2w, 1+3w, 3+3w\}$ of R, one can simplify the formula for the Gau distance d_{Gau}. For both x and $y \in Z$ or both x and $y \in U$ we have,

$$d_{Gau}(x,y) = \begin{cases} 0 & if\ x = y, \\ 1 & if\ x \neq y \text{ and } x+3y \in \{2+w, 2+3w\}, \\ 2 & \text{otherwise.} \end{cases}$$

For any $x \in \{0, 2, 2w, 2+2w\}$ and $y \in U$,

$$d_{Gau}(x,y) = \begin{cases} 0 & if\ x = y, \\ 1 & if\ x \neq y \text{ and } x+3y \in \{1, 3, 1+w, 3+3w\}, \\ 2 & if\ x \neq y \text{ and } x+3y \in \{1+3w, 3+w, 1+2w, 3+2w\}. \end{cases}$$

For any $x \in \{w, 3w, 2+w, 2+3w\}$ and $y \in U$,

$$d_{Gau}(x,y) = \begin{cases} 0 & if\ x = y, \\ 1 & if\ x \neq y \text{ and } x+3y \in \{1, 3, 3+w, 1+3w\}, \\ 2 & if\ x \neq y \text{ and } x+3y \in \{3+3w, 1+w, 1+2w, 3+2w\}. \end{cases}$$

For any two arbitrary vectors $\mathbf{x} = (x_1\ x_2\ \ldots\ x_n) \in R^n$ and $\mathbf{y} = (y_1\ y_2\ \ldots\ y_n) \in R^n$, the Gau distance $d_{Gau}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} d_{Gau}(x_i, y_i)$ is a metric on R^n induced by the metric on the elements of the ring R. We use the same notation d_{Gau} for both the metrics on R and R^n. For a linear code $\mathscr{C}$ on R, one can define a minimum Gau distance $d_{Gau} = min\{d_{Gau}(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in \mathscr{C} \text{ and } \mathbf{x} \neq \mathbf{y}\}$.

Example 4.3. *For* $\boldsymbol{x} = (2\ 2+2w\ 0\ 2w)$ *and* $\boldsymbol{y} = (0\ 2\ 2w\ 2+2w)$, *the Gau distance* $d_{Gau}(\boldsymbol{x}, \boldsymbol{y}) = 8$.

Ring element x	DNA image $\phi(x)$	Ring element x	DNA image $\phi(x)$
0	AA	1	AG
2	GG	3	GA
w	TG	$1+w$	TA
$2+w$	CA	$3+w$	CG
$2w$	CC	$1+2w$	CT
$2+2w$	TT	$3+2w$	TC
$3w$	GT	$1+3w$	GC
$2+3w$	AC	$3+3w$	AT

Table 4.1: A bijective mapping ϕ: $R \rightarrow \Sigma_{DNA}^2$ is illustrated. © [2018] IEEE. Reprinted, with permission, from Dixita Limbachiya, Krishna Gopal Benerjee, Bansari Rao and Manish K Gupta, *On DNA Codes using the Ring* $\mathbb{Z}_4 + w\mathbb{Z}_4$, In Proceedings of IEEE International Symposium on Information Theory (ISIT), pp. 2401-2405, 2018.

4.2 Gau Map and its Properties

Here, we define a Gau map ϕ (as shown in Table 4.1) from the elements of R to all DNA vectors of length 2 as:

$$\phi : R \rightarrow \Sigma_{DNA}^2 \tag{4.9}$$

One can observe the following properties of the Gau map ϕ.

1. The additive inverse of each element $x \in R$ is unique, similarly the reverse $\phi(x)^r$ of each $\phi(x) \in \Sigma_{DNA}^2$ is unique.

2. Four elements $0, 2, 2w, 2+2w \in R$ are self invertible under the addition operation. One can observe that the DNA nucleotides, $\phi(0) = AA, \phi(2) = GG, \phi(2w) = CC, \phi(2+2w) = TT$ are also self reversible.

3. For each $\phi(x) \in \Sigma_{DNA}^2$, $\exists\ \phi(y) \neq \phi(x)$ such that $\phi(x)^c = \phi(y)$ and $\phi(y)^c = \phi(x)$. Similarly, for any $x \in R$ there exists $y \in R$ $(y \neq x)$ such that $y = x + a = a + x$ and $x = y + a = a + y$, for some $a \in R$. In this work, we have considered $a = 2 + 2w$.

4. The Gau map ϕ has a property $\phi^{-1}(\phi(x)^c) = x + (2+2w)$ and $x + \phi^{-1}(\phi(x)^r) = 0$ for each $x \in R$.

5. For the ring R, $\exists$ four distinct elements $3+3w, 1+w, 3+w, 1+3w \in R$ such that $x+a$ is additive inverse of x, where $a = 2+2w$. From Table 4.1, one can observe that $\phi(x)^r = \phi(x)^c$ for only $x \in \{3+3w, 1+w, 3+w, 1+3w\}$.

4.3 Properties of DNA Codes from the Ring $\mathbb{Z}_4 + w\mathbb{Z}_4$

For any $\mathbf{x} = (x_1\, x_2\, \ldots\, x_{n-1}\, x_n) \in R^n$, we define $\phi(\mathbf{x}) = (\phi(x_1)\, \phi(x_2)\, \ldots \phi(x_{n-1})\, \phi(x_n)) \in \sum_{DNA}^{2n}$ and $\phi^{-1}(\phi(\mathbf{x})) = ((\phi^{-1}(\phi(x_1))(\, \phi^{-1}(\phi(x_2)) \ldots \phi^{-1}(\phi(x_n))) \in R^n$. The reverse of $x \in R$ is denoted by $\phi^{-1}(\phi(\mathbf{x})^r) = (\phi^{-1}(\phi(x_n)^r)\ (\phi^{-1}(\phi(x_{n-1})^r) \ldots \phi^{-1}(\phi(x_1)^r)) \in R^n$. The complement of $x \in R$ is indicated as $\phi^{-1}(\phi(\mathbf{x})^c) = (\phi^{-1}(\phi(x_1)^c)\ (\phi^{-1}(\phi(x_2)^c) \ldots \phi^{-1}(\phi(x_n)^c)) \in R^n$.

For any subset $\mathscr{C} \subseteq R^n$, the DNA code $\mathscr{C}_{DNA} = \phi(\mathscr{C}) = \{\phi(\mathbf{x}) :\ \forall\, \mathbf{x} \in \mathscr{C}\} \subseteq \sum_{DNA}^{2n}$.

4.3.1 Distance Preserving Gau Map

Theorem 4.4. *$\phi : (R^n, d_{Gau})$ to (Σ_{DNA}^{2n}, d_H) is a distance preserving map.*

Proof. We prove it for $n = 1$. Higher case is similar. For $x, y \in R$, let $x = m_{i,j} \in \mathscr{M}$, $y = m_{i',j'} \in \mathscr{M}$ for some $0 \le i, j \le 3$ and $0 \le i', j' \le 3$. Let us discuss the different cases of $x, y \in R$.

1. If $i = i'$ and $j = j'$, then $x = y$ and $\phi(x) = \phi(y) \implies d_H(\phi(x), \phi(y)) = 0$ and $d_{Gau}(x, y) = 0$.
2. If $i \neq i'$ and $j = j'$, then $d_H(\phi(x), \phi(y)) = 1$ and $d_{Gau}(x, y) = 1$.
3. If $i = i'$ and $j \neq j'$, then $d_H(\phi(x), \phi(y)) = 1$ and $d_{Gau}(x, y) = 1$.
4. If $i \neq i'$ and $j \neq j'$ then $d_H(\phi(x), \phi(y)) = 2$ and $d_{Gau}(x, y) = 2$.

Considering all the above cases, it is obvious that $\phi : (R^n, d_{Gau})$ to (Σ_{DNA}^{2n}, d_H) is an isometry. □

Example 4.5. *For $\boldsymbol{x} = (2\,0\,2+2w\,2)$, $\boldsymbol{y} = (2+2w\,0\,2w\,2)$, $\phi(\boldsymbol{x}) = (GG\ AA\ TT\ GG)$, $\phi(\boldsymbol{y}) = (TT\ AA\ CC\ GG)$, the Gau distance $d_{Gau}(\boldsymbol{x}, \boldsymbol{y}) = 4$, one can observe that the Hamming distance $d_H(\phi(\boldsymbol{x}), \phi(\boldsymbol{y})) = 4$.*

4.3.2 Closure Properties of DNA Codes

Remark 4.6. *For any $\boldsymbol{x} \in \mathscr{C}$, $\phi^{-1}(\phi(\boldsymbol{x})^r) \in \mathscr{C}$ if and only if the DNA code $\phi(\mathscr{C})$ is reversible.*

Example 4.7. *An example of a reversible DNA code $\phi(\mathscr{C})$ is given in Table 4.2. For $\boldsymbol{x}$, $\phi^{-1}(\phi(\boldsymbol{x})^r) \in \mathscr{C}$, let $\boldsymbol{x} = (1\ 2w)$, $\phi^{-1}(\phi(\boldsymbol{x})^r) = (2w\ 3)$ then $\phi(\boldsymbol{x}) = (AG\ CC)$, $\phi(\boldsymbol{x})^r = (CC\ GA) \in \phi(\mathscr{C})$ which implies that the DNA code $\phi(\mathscr{C})$ is a reversible as for each $\boldsymbol{x} \in \mathscr{C}$, $\phi^{-1}(\phi(\boldsymbol{x})^r) \in \mathscr{C}$.*

$\mathscr{C}$	$\phi(\mathscr{C})$
$(0\ 0)$	$AA\ AA$
$(1\ 2w)$	$AG\ CC$
$(2w\ 3)$	$CC\ GA$
$(2\ 2)$	$GG\ GG$

Table 4.2: Reversible Code Example

Remark 4.8. *For any $\boldsymbol{x} \in \mathscr{C}$, $\phi^{-1}(\phi(\boldsymbol{x})^c) \in \mathscr{C}$ if and only if the DNA code $\phi(\mathscr{C})$ is complement code.*

Example 4.9. *An example of a complement DNA code $\phi(\mathscr{C})$ is given in Table 4.3. For $\boldsymbol{x}$, $\phi^{-1}(\phi(\boldsymbol{x})^c) \in \mathscr{C}$, let $\boldsymbol{x} = (1\ 2w)$, $\phi^{-1}(\phi(\boldsymbol{x})^c) = (3 + 2w\ 2)$, $\phi(\boldsymbol{x}) = (AG\ CC)$, $\phi(\boldsymbol{x})^r = (TC\ GG) \in \phi(\mathscr{C})$. Hence, the DNA code $\phi(\mathscr{C})$ is a complement code as for each $\boldsymbol{x} \in \mathscr{C}$, $\phi^{-1}(\phi(\boldsymbol{x})^c) \in \mathscr{C}$.*

$\mathscr{C}$	$\phi(\mathscr{C})$
$(0\ 0)$	$AA\ AA$
$(2 + 2w\ 2 + 2w)$	$TT\ TT$
$(1\ 2w)$	$AG\ CC$
$(3 + 2w\ 2)$	$TC\ GG$

Table 4.3: Complement Code Example

Remark 4.10. *For any $\boldsymbol{x} \in \mathscr{C}$, if $\phi(\boldsymbol{x})^r \in \phi(\mathscr{C})$ and $(\phi(\boldsymbol{x})^c \in \phi(\mathscr{C})$ then $\phi^{-1}(\phi(\boldsymbol{x})^{rc}) \in \phi(\mathscr{C})$.*

Example 4.11. *An example of a reversible-complement DNA code $\phi(\mathscr{C})$ is given in Table 4.4. For $\boldsymbol{x}$, $\phi^{-1}(\phi(\boldsymbol{x})^{rc}) \in \mathscr{C}$, let $\boldsymbol{x} = (1\ 2w)$, $\phi^{-1}(\phi(\boldsymbol{x})^{rc}) = (2\ 1 + 2w)$ $\phi(\boldsymbol{x}) = (AG\ CC)$, $\phi(\boldsymbol{x})^r = (GG\ CT) \in \phi(\mathscr{C})$. Thus, the DNA code $\phi(\mathscr{C})$ is a reversible-complement code as for each $\boldsymbol{x} \in \mathscr{C}$, $\phi^{-1}(\phi(\boldsymbol{x})^{rc}) \in \mathscr{C}$.*

$\mathscr{C}$	$\phi(\mathscr{C})$
$(0\ 0)$	$AA\ AA$
$(2+2w\ 2+2w)$	$TT\ TT$
$(1\ 2w)$	$AG\ CC$
$(2\ 1+2w)$	$GG\ CT$

Table 4.4: Reversible-Complement DNA Code Example

4.3.3 Linearity on DNA Codes

Lemma 4.12. *For any $\boldsymbol{x}, \boldsymbol{y} \in R^n, \phi^{-1}(\phi(a\boldsymbol{x}+b\boldsymbol{y})^r) = a\phi^{-1}(\phi(\boldsymbol{x})^r) + b\phi^{-1}(\phi(\boldsymbol{y})^r)$, where $a, b \in R$.*

Proof. For any $\mathbf{x}, \mathbf{y} \in R^n$, $\mathbf{x} = (x_1\ x_2 \dots x_n)$ and $\mathbf{y} = (y_1\ y_2 \dots y_n)$.

$$
\begin{aligned}
\textit{Consider } \phi(a\mathbf{x}+b\mathbf{y})^r =& (\phi(ax_n+by_n)^r\ \phi(ax_{n-1}+by_{n-1})^r \dots \phi(ax_1+by_1)^r). \textit{ Thus}\\
\phi^{-1}(\phi(a\mathbf{x}+b\mathbf{y})^r) =& (\phi^{-1}(\phi(ax_n+by_n)^r)\ \phi^{-1}(\phi(ax_{n-1}+by_{n-1})^r) \dots \phi^{-1}(\phi(ax_1+by_1)^r))\\
=& ((3ax_n+3by_n)\ (3ax_{n-1}+3by_{n-1}) \dots (3ax_1+3by_1))\\
=& (a\phi^{-1}(\phi(x_n)^r)+b\phi^{-1}(\phi(y_n)^r)\ a\phi^{-1}(\phi(x_{n-1})^r)+b\phi^{-1}(\phi(y_{n-1})^r) \dots\\
& \dots a\phi^{-1}(\phi(x_1)^r)+b\phi^{-1}(\phi(y_1)^r))\\
=& a(\phi^{-1}(\phi(x_n)^r)\ \phi^{-1}(\phi(x_{n-1})^r)\ \dots \phi^{-1}(\phi(x_1)^r))\\
& + b(\phi^{-1}(\phi(y_n)^r)\ \phi^{-1}(\phi(y_{n-1})^r) \dots \phi^{-1}(\phi(y_1)^r))\\
=& a\phi^{-1}(\phi(\mathbf{x})^r)+b\phi^{-1}(\phi(\mathbf{y})^r).
\end{aligned}
$$

□

Example 4.13. *Let $a = 2, b = 3, \mathbf{x} = (2+3w\ 2)$ and $\mathbf{y} = (1\ 1+w)$ then $a\boldsymbol{x}+b\boldsymbol{y} = (3+2w\ 3+3w)$. For $\phi(a\boldsymbol{x}+b\boldsymbol{y}) = (TC\ AT)$, $\phi(a\boldsymbol{x}+b\boldsymbol{y})^r = (TA\ CT)$.*

$$\phi^{-1}(\phi(a\boldsymbol{x}+b\boldsymbol{y})^r) = (1+w\ 1+2w) \tag{4.10}$$

- $\phi(\boldsymbol{x}) = (AC\ GG)$ *and* $\phi(\boldsymbol{y}) = (AG\ TA)$
- $\phi(\boldsymbol{x})^r = (GG\ CA)$ *and* $\phi(\boldsymbol{y})^r = (AT\ GA)$
- $\phi^{-1}(\phi(\boldsymbol{x})^r) = (2\ 2+w)$ *and* $\phi^{-1}(\phi(\boldsymbol{y})^r) = (3+3w\ 3)$
- $a\phi^{-1}(\phi(\boldsymbol{x})^r) = (0\ 2w)$ *and* $b\phi^{-1}(\phi(\boldsymbol{y})^r) = (1+w\ 1)$

$$a\phi^{-1}(\phi(\boldsymbol{x})^r) + b\phi^{-1}(\phi(\boldsymbol{y})^r) = (1+w\ 1+2w) \tag{4.11}$$

By Equations 4.10 and 4.11, $\phi^{-1}(\phi(a\boldsymbol{x}+b\boldsymbol{y})^r) = a\phi^{-1}(\phi(\boldsymbol{x})^r) + b\phi^{-1}(\phi(\boldsymbol{y})^r)$.

One can obtain similar result for the higher order in Remark 4.14 on the linearity of the reversible code.

Remark 4.14. *For any positive integer* k *and* $1 \leq i \leq k$, *if* $\boldsymbol{x}_i \in R^n$, *then* $\phi^{-1}(\phi(\sum_{i=1}^{k} a_i\boldsymbol{x}_i)^r) = \sum_{i=1}^{k} a_i\phi^{-1}(\phi(\boldsymbol{x}_i)^r)$, *where* $a_i \in R$.

Corollary 4.15. *For any* $\boldsymbol{x}, \boldsymbol{y} \in R^n, \phi^{-1}(\phi(a\boldsymbol{x}+b\boldsymbol{y})^c) = a\phi^{-1}(\phi(\boldsymbol{x})^c) + b\phi^{-1}(\phi(\boldsymbol{y})^c)$ *if* $a+b \in \{1, 1+2w, 3, 3+2w\}$, *where* $a, b \in R$.

Proof. Note that for any $a+b \in \{1, 3, 1+2w, 3+2w\}$, $(a+b)(2+2w) = 2+2w$. For any $\mathbf{x}, \mathbf{y} \in R^n$, $\mathbf{x} = (x_1\ x_2 \ldots x_n)$ and $\mathbf{y} = (y_1\ y_2 \ldots y_n)$.

$$
\begin{aligned}
\textit{Consider } \phi(a\mathbf{x}+b\mathbf{y})^c =& (\phi(ax_1+by_1)^c\ \phi(ax_2+by_2)^c \ldots \phi(ax_n+by_n)^c). \textit{ Thus}\\
\phi^{-1}(\phi(a\mathbf{x}+b\mathbf{y})^c) =& (\phi^{-1}(\phi(ax_1+by_1)^c)\ \phi^{-1}(\phi(ax_2+by_2)^c) \ldots \phi^{-1}(\phi(ax_n+by_n)^c))\\
=& (ax_1+by_1+2+2w\ ax_2+by_2+2+2w \ldots ax_n+by_n+2+2w)\\
=& (ax_1+by_1+(a+b)(2+2w)\ ax_2+by_2+(a+b)(2+2w) \ldots\\
& \ldots ax_n+by_n+(a+b)(2+2w))\\
=& (a(x_1+2+2w)+b(y_1+2+2w)\ a(x_2+2+2w)+b(y_2+2+2w) \ldots\\
& \ldots a(x_n+2+2w)+b(y_n+2+2w))\\
=& (a\phi^{-1}(\phi(x_1)^c)+b\phi^{-1}(\phi(y_1)^c)\ a\phi^{-1}(\phi(x_2)^c)+b\phi^{-1}(\phi(y_2)^c) \ldots\\
& \ldots a\phi^{-1}(\phi(x_n)^c)+b\phi^{-1}(\phi(y_n)^c))\\
=& a((\phi^{-1}(\phi(x_n)^c)\ \phi^{-1}(\phi(x_{n-1})^c)\ \ldots \phi^{-1}(\phi(x_1)^c))\\
& + b(\phi^{-1}(\phi(y_n)^c)\ \phi^{-1}(\phi(y_{n-1})^c) \ldots \phi^{-1}(\phi(y_1)^c))\\
=& a\phi^{-1}(\phi(\mathbf{x})^c) + b\phi^{-1}(\phi(\mathbf{y})^c).
\end{aligned}
$$

□

Example 4.16. *Let* $a = 2, b = 3, \mathbf{x} = (2+3w\ 2)$ *and* $\mathbf{y} = (1\ 1+w)$ *then* $a\boldsymbol{x}+b\boldsymbol{y} = (3+2w\ 3+3w)$. *For* $\phi(a\boldsymbol{x}+b\boldsymbol{y}) = (TC\ AT)$, $\phi(a\boldsymbol{x}+b\boldsymbol{y})^c = (AG\ TA)$

$$\phi^{-1}(\phi(a\boldsymbol{x}+b\boldsymbol{y})^c) = (1\ 1+w) \tag{4.12}$$

- $\phi(\boldsymbol{x}) = (AC\ GG)$ *and* $\phi(\boldsymbol{y}) = (AG\ TA)$
- $\phi(\boldsymbol{x})^c = (TG\ CC)$ *and* $\phi(\boldsymbol{y})^c = (TC\ AT)$

- $\phi^{-1}(\phi(\boldsymbol{x})^c) = (w\ 2w)$ *and* $\phi^{-1}(\phi(\boldsymbol{y})^c) = (3+2w\ 3+3w)$
- $a\phi^{-1}(\phi(\boldsymbol{x})^c) = (2w\ 0)$ *and* $b\phi^{-1}(\phi(\boldsymbol{y})^c) = (1+2w\ 1+w)$

$$a\phi^{-1}(\phi(\boldsymbol{x})^c) + b\phi^{-1}(\phi(\boldsymbol{y})^c) = (1\ 1+w) \tag{4.13}$$

By Equations 4.12 and 4.13, $\phi^{-1}(\phi(a\boldsymbol{x}+b\boldsymbol{y})^c) = a\phi^{-1}(\phi(\boldsymbol{x})^c) + b\phi^{-1}(\phi(\boldsymbol{y})^c)$.

Similarly, observe Remark 4.17 on the linearity for higher order on the DNA codes for the complement constraint.

Remark 4.17. *For any positive integer k and* $1 \leq i \leq k$, *if* $\boldsymbol{x}_i \in R^n$, *then* $\phi^{-1}(\phi(\sum_{i=1}^{k} a_i\boldsymbol{x}_i)^c) = \sum_{i=1}^{k} a_i\phi^{-1}(\phi(\boldsymbol{x}_i)^c)$ *if* $\sum_{i=1}^{k} a_i \in \{1, 1+2w, 3, 3+2w\}$, *where* $a_i \in R$.

4.3.4 Closure of Reversible, Complement and Reversible-Complement Codes

Lemma 4.18. *For any row* $\boldsymbol{x}$ *of G over the ring R, the DNA code* $\phi(< G >_R)$ *is closed under reverse if and only if* $\phi^{-1}(\phi(\boldsymbol{x})^r) \in < G >_R$, *the row span of G over R.*

Proof. Let $\phi(\boldsymbol{y}) \in \phi(< G >_R) = \phi(\mathscr{C})$ *for some* $\boldsymbol{y} \in \mathscr{C}$. *Thus* $\boldsymbol{y} = \sum_{i=1}^{k} a_i\boldsymbol{x}_i$ *for rows* $\boldsymbol{x}_i$ *of G. Consider* $\phi^{-1}(\phi(\boldsymbol{y})^r) = \phi^{-1}(\phi(\sum_{i=1}^{k} a_i\boldsymbol{x}_i)^r) = \sum_{i=1}^{k} a_i\phi^{-1}(\phi(\boldsymbol{x}_i)^r) \in \mathscr{C}$ *(by using Remark 4.14). Thus* $\phi^{-1}(\phi(\boldsymbol{y})^r) \in \mathscr{C}$ *which directs* $\phi(\boldsymbol{y})^r \in \phi(\mathscr{C})$. *Hence the DNA code is closed under reverse (by Remark 4.6).* The otherside is obvious. □

Example 4.19. *For*

$$G = \begin{pmatrix} 2 & 2 & 2 & 2 \\ 3+w & 3+w & 3+w & 3+w \end{pmatrix},$$

the DNA code $\phi(< G >_R)$ *is closed under reverse (see Table 4.5). For* $\boldsymbol{x} = (3+w\ 3+w\ 3+w\ 3+w)$, $\phi(\boldsymbol{x}) = (CG\ CG\ CG\ CG) \implies \phi^{-1}(\phi(\boldsymbol{x})^r) = (1+3w\ 1+3w\ 1+3w\ 1+3w)$, $\phi(\boldsymbol{x})^r = (GC\ GC\ GC\ GC) \implies \phi(\boldsymbol{x})^r \in \phi(< G >_R)$.

Lemma 4.20. *For* $\boldsymbol{x} = (x_1\ x_2\ \ldots x_n) \in R^n$, $\phi(\boldsymbol{x}) = \phi(\boldsymbol{x})^r$ *if and only if* $x_i + x_{n-i+1} = 0$ *for* $i = 1, 2, \ldots, n$.

No.	$< G >_R$	$\phi(< G >_R)$
1	$(0\,0\,0\,0)$	$(AA\ AA\ AA\ AA)$
2	$(3+w\ 3+w\ 3+w\ 3+w)$	$(CG\ CG\ CG\ CG)$
3	$(2+2w\ 2+2w\ 2+2w\ 2+2w)$	$(TT\ TT\ TT\ TT)$
4	$(1+3w\ 1+3w\ 1+3w\ 1+3w)$	$(GC\ GC\ GC\ GC)$
5	$(2+w\ 2+w\ 2+w\ 2+w)$	$(CA\ CA\ CA\ CA)$
6	$(1+2w\ 1+2w\ 1+2w\ 1+2w)$	$(CT\ CT\ CT\ CT)$
7	$(3w\ 3w\ 3w\ 3w)$	$(GT\ GT\ GT\ GT)$
8	$(3\,3\,3\,3)$	$(GA\ GA\ GA\ GA)$
9	$(2w\ 2w\ 2w\ 2w)$	$(CC\ CC\ CC\ CC)$
10	$(3+3w\ 3+3w\ 3+3w\ 3+3w)$	$(AT\ AT\ AT\ AT)$
11	$(2\,2\,2\,2)$	$(GG\ GG\ GG\ GG)$
12	$(1+w\ 1+w\ 1+w\ 1+w)$	$(TA\ TA\ TA\ TA)$
13	$(2+3w\ 2+3w\ 2+3w\ 2+3w)$	$(AC\ AC\ AC\ AC)$
14	$(1\,1\,1\,1)$	$(AG\ AG\ AG\ AG)$
15	$(w\ w\ w\ w)$	$(TG\ TG\ TG\ TG)$
16	$(3+2w\ 3+2w\ 3+2w\ 3+2w)$	$(TC\ TC\ TC\ TC)$

Table 4.5: Example of Closure of Reversible Code

Proof. For $\mathbf{x} = (x_1\ x_2\ \ldots x_n) \in R^n$, consider

$$\begin{aligned}
&\phi(\mathbf{x}) = \phi(\mathbf{x})^r \\
\Leftrightarrow\ &\phi(x_i) = \phi(x_{n-i+1})^r && (\phi(\mathbf{x})^r = (\phi(x_n)^r\phi(x_{n-1})^r \ldots \phi(x_1)^r)) \\
\Leftrightarrow\ &x_i = \phi^{-1}(\phi(x_{n-i+1})^r) && (\phi \text{ is bijective}) \\
\Leftrightarrow\ &x_i = 3x_{n-i+1} && (x + \phi^{-1}(\phi(x)^r) = 0) \\
\Leftrightarrow\ &x_i + x_{n-i+1} = 0.
\end{aligned}$$

□

Lemma 4.21. *For a matrix G over the ring R, the DNA code $\phi(< G >_R)$ is closed under complement if and only if* $\mathbf{2+2w} \in < G >_R$*, where* $\mathbf{2+2w}$ *is a string with each element* $2+2w$*.*

Proof. Let $\phi(\mathbf{x}) \in \phi(< G >_R) = \phi(\mathscr{C})$ for some $\mathbf{x} \in \mathscr{C}$ and $\mathbf{2+2w} = (2+2w\ 2+2w\ 2+2w\ \ldots\ 2+2w) \in \mathscr{C}$. Thus $\mathbf{x} + \mathbf{2+2w} \in \mathscr{C}$ but $\phi^{-1}(\phi(\mathbf{x})^c) = \mathbf{x} + \mathbf{2+2w}$ (by using Table 4.1). Thus $\phi^{-1}(\phi(\mathbf{x})^c) \in \mathscr{C}$ and therefore $\phi(\mathbf{x})^c \in \phi(\mathscr{C})$. Hence, the DNA code $\mathscr{C}_{DNA}$ is closed under complement using Remark 4.8. For the other side of the statement, if $\phi(\mathscr{C})$ is closed under complement, then $\phi^{-1}(\phi(\mathbf{0})^c) \in \mathscr{C}$

(Since $\mathbf{0} \in \mathscr{C}$). But $\phi^{-1}(\phi(\mathbf{0})^c) = \mathbf{0} + \mathbf{2+2w} = \mathbf{2+2w}$. Hence $\mathbf{2+2w} \in \mathscr{C}$. □

Example 4.22. *If* $G = (2\,2\,2\,2)$, *then* $< G >_R = \{(2\,2\,2\,2), (2w\ 2w\ 2w\ 2w), (0\,0\,0\,0), (2+2w\ 2+2w\ 2+2w\ 2+2w)\}$. *Thus,* $\phi(< G >_R) = \{(GG\ GG\ GG\ GG), (CC\ CC\ CC\ CC), (AA\ AA\ AA\ AA), (TT\ TT\ TT\ TT)\}$. *Let* $\boldsymbol{x}$ = *(*$2+2w\ 2+2w\ 2+2w\ 2+2w$*) and* $\phi(\boldsymbol{x}) = (TT\ TT\ TT\ TT) \implies \phi^{-1}(\phi(\boldsymbol{x})^c) =$ *(*$0\,0\,0\,0$*),* $\phi(\boldsymbol{x})^c = (AA\ AA\ AA\ AA) \implies \phi(\boldsymbol{x})^c \in \phi(< G >_R)$

Lemma 4.23. *For* $\boldsymbol{x} = (x_1\ x_2\ \ldots x_n) \in R^n$, $\phi(\boldsymbol{x}) = \phi(\boldsymbol{x})^{rc}$ *if and only if* $x_i + x_{n-i+1} = 2 + 2w$ *for* $i = 1, 2, \ldots, n$.

Proof. For $\mathbf{x} = (x_1\ x_2\ \ldots x_n) \in R^n$, consider

$$\begin{aligned}
&\phi(\mathbf{x}) = \phi(\mathbf{x})^{rc} \\
\Leftrightarrow\ &\phi(x_i) = \phi(x_{n-i+1})^{rc} \qquad (\phi(\mathbf{x})^{rc} = (\phi(x_n)^{rc}\phi(x_{n-1})^{rc} \ldots \phi(x_1)^{rc})) \\
\Leftrightarrow\ &x_i = \phi^{-1}(\phi(x_{n-i+1})^{rc}) \quad (\phi \text{ is bijective}) \\
\Leftrightarrow\ &x_i = 3x_{n-i+1} + 2 + 2w \quad (\phi^{-1}(\phi(x)^r) = x + 2 + 2w) \\
\Leftrightarrow\ &x_i + x_{n-i+1} = 2 + 2w.
\end{aligned}$$

□

The following Theorem is obtained using Lemmas 4.12, 4.18 and 4.21.

Theorem 4.24. *Let* $\mathscr{C}(n, M, d_{Gau})$ *be a code over the ring R with the length n, the number of the codewords M and the minimum Gau distance* d_{Gau}*, such that each rows of the generator matrix of* $\mathscr{C}$ *satisfy the conditions given in Lemma 4.18 and 4.21, then* $\phi(\mathscr{C})$ *is a* $\mathscr{C}_{DNA}(2n, M, d_H)$ *DNA code with the length* $2n$*, the number of the codewords M and the minimum Hamming distance* d_H*. The DNA code* $\mathscr{C}_{DNA}$ *is reversible, complement and reversible-complement.*

Proof. Every x of R is mapped to an ordered pair of DNA alphabets through the bijective mapping $\phi : R \to \Sigma^2_{DNA}$ and $\mathscr{C}_{DNA} = \{\phi(\mathbf{x}) : \forall\ \mathbf{x} \in \mathscr{C}\}$, thus the DNA code has length $2n$. One can observe that ϕ is a bijective map from R to Σ^2_{DNA} implies that $\mathscr{C}_{DNA}$ has M codewords. Moreover, from Theorem 4.4, ϕ is the distance preserving from R to $\mathscr{C}_{DNA}$ leads to $\mathscr{C}_{DNA}$ has the minimum Hamming distance $d_H = d_{Gau}$. □

Using the results of Theorems 4.18, 4.21 and 4.24, families of the DNA codes from Octacodes types codes, Simplex type code and Reed-Muller type codes are developed over the ring R which satisfies the Hamming distance, reverse and reverse complement constraints in the next section.

4.4 Families of DNA Codes from the Ring $\mathbb{Z}_4 + w\mathbb{Z}_4$

In this section, using the results discussed in Section 4, we give new classes of the DNA codes that satisfies the Hamming, reverse and reverse complement constraints.

4.4.1 DNA Codes from Octacodes Type Codes

There has been an interesting history of connecting two non-linear binary codes (Kerdock and Preparata) with linear codes over $\mathbb{Z}_4$ [40]. The Octacode (a linear self dual code of length $n = 8$, code size $M = 256$ and minimum Lee weight 8 over $\mathbb{Z}_4$) turns out to be a special case connecting the binary non-linear codes (the Nordstrom-Robinson code) [40]. In this section, we construct a code similar to the original Octacode from the ring $\mathbb{Z}_4 + w\mathbb{Z}_4$. The self dual code is generated by a generator matrix consisting of the cyclic shifts of the vector $(0\ 2\ 2w\ 2+2w\ 0\ 2\ 2w\ 2+2w)$ over R. The DNA code is generated from the generator matrix $\mathscr{O}$ of the Octacode in Example 4.25. It satisfies the reverse and reverse complement constraints.

Example 4.25. *The DNA code $\mathscr{C}_{DNA}(n = 16, M = 64, d_H = 8)$ can be obtained from cyclic shifts of the vector $(0\ 2w\ 2\ 2+2w\ 0\ 2w\ 2\ 2+2w)$.*

$$\mathscr{O} = \begin{pmatrix} 0 & 2w & 2 & 2+2w & 0 & 2w & 2 & 2+2w \\ 2+2w & 0 & 2w & 2 & 2+2w & 0 & 2w & 2 \\ 2 & 2+2w & 0 & 2w & 2 & 2+2w & 0 & 2w \\ 2w & 2 & 2+2w & 0 & 2w & 2 & 2+2w & 0 \end{pmatrix} \tag{4.14}$$

AAAAAAAAAAAAAAAA	TTAACCGGTTAACCGG	CCAAAACCCCAAAACC	GGAACCTTGGAACCTT
CCGGTTAACCGGTTAA	GGGGGGGGGGGGGGGG	AAGGTTCCAAGGTTCC	TTGGGGTTTTGGGGTT
AACCCCAAAACCCCAA	TTCCAAGGTTCCAAGG	CCCCCCCCCCCCCCCC	GGCCAATTGGCCAATT
CCTTGGAACCTTGGAA	GGTTTTGGGGTTTTGG	AATTGGCCAATTGGCC	TTTTTTTTTTTTTTTT
GGTTAACCGGTTAACC	CCTTCCTTCCTTCCTT	TTTTAAAATTTTAAAA	AATTCCGGAATTCCGG
TTCCTTCCTTCCTTCC	AACCGGTTAACCGGTT	GGCCTTAAGGCCTTAA	CCCCGGGGCCCCGGGG
GGGGCCCCGGGGCCCC	CCGGAATTCCGGAATT	TTGGCCAATTGGCCAA	AAGGAAGGAAGGAAGG
TTAAGGCCTTAAGGCC	AAAATTTTAAAATTTT	GGAAGGAAGGAAGGAA	CCAATTGGCCAATTGG
CCCCAAAACCCCAAAA	GGCCCCGGGGCCCCGG	AACCAACCAACCAACC	TTCCCCTTTTCCCCTT
AATTTTAAAATTTTAA	TTTTGGGGTTTTGGGG	CCTTTTCCCCTTTTCC	GGTTGGTTGGTTGGTT
CCAACCAACCAACCAA	GGAAAAGGGGAAAAGG	AAAACCCCAAAACCCC	TTAAAATTTTAAAATT
AAGGGGAAAAGGGGAA	TTGGTTGGTTGGTTGG	CCGGGGCCCCCGGGCC	GGGGTTTTGGGGTTTT
TTGGAACCTTGGAACC	AAGGCCTTAAGGCCTT	GGGGAAAAGGGGAAAA	CCGGCCGGCCGGCCGG
GGAATTCCGGAATTCC	CCAAGGTTCCAAGGTT	TTAATTAATTAATTAA	AAAAGGGGAAAAGGGG
TTTTCCCCTTTTCCCC	AATTAATTAATTAATT	GGTTCCAAGGTTCCAA	CCTTAAGGCCTTAAGG
GGCCGGCCGGCCGGCC	CCCCTTTTCCCCTTTT	TTCCGGAATTCCGGAA	AACCTTGGAACCTTGG

Table 4.6: DNA codewords generated from the matrix $\phi(\mathscr{O})$ with $n = 16, M = 64, d_H = 8$ obtained from Octacode in Example 4.25 satisfies reverse and reverse complement constraints.

Let

$$\mathscr{O} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{pmatrix}$$

be a generator matrix for the Octacode in Example 4.25. By Lemma 4.21, the DNA code $\mathscr{C}_{DNA} = \phi(< \mathscr{O} >)$ is closed under complement because $\mathbf{2+2w} \in < \mathscr{O} >$, (because $\mathbf{2+2w} = (1+w)(\mathbf{x}_1 + \mathbf{x}_3)$). For the rows $\mathbf{x}_i \in \mathscr{O}$ $(1 \leq i \leq 4)$, $\phi^{-1}(\phi(\mathbf{x}_i)^r) = \mathbf{2+2w} + \mathbf{x}_i$, for $i = 1, 3$ and $\phi^{-1}(\phi(\mathbf{x}_i)^r) = \mathbf{2w} + \mathbf{x}_i$, for $i = 2, 4$. Thus by using Lemma 4.18, the DNA code $\mathscr{C}_{DNA}$ satisfies the reverse constraint.

Remark 4.26. *On the similar lines, one can obtain Octacodes type DNA codes with parameters listed in Table 4.7 using the different first row vector.*

First Row Vector	DNA Code $\mathscr{C}_{DNA}(n, M, d_H)$
$(0\ 2\ 2w\ 2+2w)$	$(8, 16, 4)$
$(0\ 2w\ 2\ 2+2w)$	$(8, 64, 4)$
$(0\ 2w\ 2\ 2+2w\ 0\ 2w\ 2\ 2+2w)$	$(16, 64, 8)$
$(0\ 2\ 2w\ 2+2w\ 0\ 2\ 2w\ 2+2w)$	$(16, 16, 8)$

Table 4.7: Octacodes types DNA code $\mathscr{C}_{DNA}$.

DNA codewords generated from the Octacodes have interesting properties from the application point of view. By removing the trivial codewords (codewords with no GCs or all GCs), one can obtain the DNA codewords with 50% GC-weight which are essential for the DNA computation.

4.4.2 DNA Codes from Simplex Type Codes

Binary simplex codes have a unique geometrical significance (being the dual of the Hamming codes and each having a fixed weight) and were known in 1945 [39] in statistical connections before the actual discovery of the Hamming codes by R. Hamming in 1948. Many researchers have considered simplex codes over rings [54]. The simplex codes of type α and β over the ring $\mathbb{Z}_4$ have been studied in [14]. The simplex codes over the ring $\mathbb{F}_2 + u\mathbb{F}_2$ were given in [7]. Recently, K. Chatouh et al. generalized the simplex codes over the ring R_q in [21]. It is natural to study the DNA codes using the simplex codes. DNA codes that avoid the secondary structure formation were designed in [90] by using the cyclic simplex codes. DNA codes satisfying the GC-weight constraint were also studied using simplex codes in [52]. In this section, we have designed the DNA codes using the simplex type codes over the ring R. We give a generator matrix G_k^β for the simplex type β over the ring R. The DNA codes which satisfies the reverse and reverse complement constraints are given in Example 4.29.

Let G_k^β be a matrix over R defined inductively by

$$G_k^\beta = \left(\begin{array}{c|c|c|c} 0\ldots0 & 2\ldots2 & 2w\ldots2w & 2+2w\ldots2+2w \\ \hline G_{k-1}^\beta & G_{k-1}^\beta & G_{k-1}^\beta & G_{k-1}^\beta \end{array}\right), k \geq 3 \tag{4.15}$$

with

$$G_2^\beta = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 2 & 2w & 2+2w \\ 0 & 2 & 2w & 2+2w & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{4.16}$$

Let S_k^β be a code generated by the generator matrix of type β simplex type code over the ring R then for $k > 1$, $n = 2^{2k-1}$, $M = 2^{2k+4}$ and $d_{Gau} = 2^{2k-1}$.

Remark 4.27. *If A_{k-1} denotes an array of codewords in S_{k-1}^β and if $\boldsymbol{i} = (i\ i\ \ldots\ i)$,*

where, $i \in \{2, 2w, 2+2w\}$ then an array of all codewords of S_k^β is given by

$$\begin{pmatrix} A_{k-1} & A_{k-1} & A_{k-1} & A_{k-1} \\ A_{k-1} & \boldsymbol{2} + A_{k-1} & \boldsymbol{2w} + A_{k-1} & \boldsymbol{2+2w} + A_{k-1} \\ A_{k-1} & \boldsymbol{2w} + A_{k-1} & A_{k-1} & \boldsymbol{2w} + A_{k-1} \\ A_{k-1} & \boldsymbol{2+2w} + A_{k-1} & \boldsymbol{2w} + A_{k-1} & \boldsymbol{2} + A_{k-1} \end{pmatrix} \tag{4.17}$$

Theorem 4.28. *If $S_k^\beta(n, M, d_{Gau})$ is a simplex β type code over the ring $R = \mathbb{Z}_4 + w\mathbb{Z}_4$ then the parameters of the corresponding DNA code $\mathscr{C}_{DNA}(n, M, d_H)$ are $n = 2^{2k}$, $M = 2^{2k+4}$ and $d_H = 2^{2k-1}$. $\mathscr{C}_{DNA}(n, M, d_H)$ satisfies the reverse and reverse complement constraints.*

Proof. The proof has two parts. The first part contains the proof for the parameters of the DNA code $\mathscr{C}_{DNA} = \phi(S_k^\beta)$. The second part proves that the DNA code satisfies reverse and reverse complement constraints.

1. By using the induction on k, one can observe that the length of the DNA code $\mathscr{C}_{DNA}$ is $n = 2^{2k}$ with the initial condition on G_2^β. The matrix G_k^β is type $\{2, 0, k-2, 0\}$ matrix. Hence, the number of DNA codewords $M = 2^{2k+4}$. For G_k^β, the minimum Gau distance is proved by using induction on k. For $k = 2$, the base case G_2^β is trivial. For some positive integer $k-1$, let the minimum Gau distance of S_{k-1}^β is $2^{2(k-1)-1}$. For each $z \in \{0, 2, 2w, 2+2w\}$, note that $(z\, z \ldots z) \in S_{k-1}^\beta$. Thus, by the matrix structure, the minimum Gau distance of S_k^β can not be more than four times of $(2^{2(k-1)-1})$ *i.e.* 2^{2k-1}. But, $d_{Gau}(\mathbf{0}, z\mathbf{x}_k) = 2^{2k-1}$, where $\mathbf{0}$ is all zero vector and $\mathbf{x}_k$ is the last row of the matrix G_k^β. Hence, the minimum Gau distance of S_k^β is 2^{2k-1}. Using Theorem 4.24, the result holds for the Hamming distance of the DNA code $\mathscr{C}_{DNA}$.

2. To prove the closure of complement, observe that $\boldsymbol{2+2w}$ is $2+2w$ times the sum of the last two rows of G_k^β. Hence $\boldsymbol{2+2w} \in S_k^\beta$. Thus by Lemma 4.21, the DNA code $\mathscr{C}_{DNA}$ is closed under complement. The closure of $\mathscr{C}_{DNA}$ with respect to the reverse can be proved by using Lemma 4.18. Let $\mathbf{x}_i$ be the i^{th} row of the matrix G_k^β, where $i = 1, 2, \ldots, k$. For each $\mathbf{x}_i \in G_k^\beta$, observe that

$$\phi^{-1}(\phi(\mathbf{x}_i)^r) = \begin{cases} (2+2w)(\mathbf{x}_k + \mathbf{x}_{k-1}) + \mathbf{x}_i & \text{if } i = 1, 2, \ldots, k-2, \\ (2+2w)\mathbf{x}_{k-1} + 3\mathbf{x}_k & \text{if } i = k-1, \\ (2+2w)\mathbf{x}_k + 3\mathbf{x}_{k-1} & \text{if } i = k. \end{cases}$$

As $2+2w,\ \mathbf{x}_k,\ \mathbf{x}_{k-1} \in S_k^\beta, \phi^{-1}(\phi(\mathbf{x}_i)^r) \in S_k^\beta$. Thus, $\mathscr{C}_{DNA}$ satisfies the reverse constraint (from Remark 4.18). Hence, by using Remark 4.10, the DNA code satisfies reverse and reverse complement constraints.

□

Example 4.29. *From the matrix G_2^β, the DNA codes with parameters $(16, 256, 8)$ satisfying the reverse and reverse complement constraints can be generated. Inductively from G_3^β, DNA codes with length $n = 64$, $M = 1024$, $d_H = 32$ is constructed.*

The DNA codes obtained from the simplex class have purine rich DNA codewords (composition of AGs in the DNA codeword). These DNA codewords are essential for the DNA motifs (a small length of functional DNA codewords). They also play a significant role in the transcription of genes.

4.4.3 DNA codes from the First order Reed-Muller Type Codes

The Reed-Muller codes (RM) are one of the best known oldest error correcting codes discovered by Reed and Muller in 1954 [93]. First, the binary Reed-Muller codes were introduced and then it was generalized to any $q-$ary alphabets [70]. DNA codes from the Reed-Muller codes have been constructed in [52] from the ring $\mathbb{F}_2 + u\mathbb{F}_2$ with $u^2 = 0$ satisfying reverse and reverse complement and GC-weight using the defined Gray map. In [52], only odd lengths codes were consider. In this thesis, we construct even length DNA codes using a special type of Reed-Muller codes via the proposed Gau map ϕ. These DNA codes follows the reverse and reverse complement and GC-weight constraints.

In this work, we define a few kinds of Reed-Muller type code over the ring R. The Reed-Muller code is denoted as $\mathcal{R}(r, m)$, where r is the order of the code, and m determines the length of the code, $n = 2^m$.

For each positive integer m ($r \leq m$), the first order Reed-Muller Type code $\mathcal{R}(1,m)$ over R where, $w^2 = 2 + 2w$, is defined by the generator matrix:

$$G_{1,m+1} = \begin{pmatrix} G_{1,m} & G_{1,m} \\ 0 \ldots 0 & z \ldots z \end{pmatrix},\ m \geq 1$$

with

$$G_{1,1} = \begin{pmatrix} 1 & 1 \\ 0 & z \end{pmatrix},$$

where $z \in \{2, w, 2+w, 2w, 2+2w, 3w, 2+3w\}$.

Theorem 4.30. *For the first order Reed-Muller Type code $\mathcal{R}(1,m)$ over $\mathbb{Z}_4 + w\mathbb{Z}_4$, $\exists$ the DNA code $\mathscr{C}_{DNA}(n, M, d_H)$ that satisfies reverse and reverse complement constraints with $n = 2^{m+1}$,*

$$M = \begin{cases} 2^{m+4} & if \quad z \in \{2w\}, \\ 2^{2m+4} & if \quad z \in \{2, 2+2w\}, \\ 2^{3m+4} & if \quad z \in \{w, 2+w, 3w, 2+3w\}, \end{cases}$$

and

$$d_H = \begin{cases} 2^m & if \quad z \in \{2w, 2, 2+2w\}, \\ 2^{m-1} & if \quad z \in \{w, 2+w, 3w, 2+3w\}. \end{cases}$$

Proof. The proof has two parts. The first part contains the proof for the parameters of the DNA code $\mathscr{C}_{DNA} = \phi(\mathcal{R}(1,m))$. The second part proves that the DNA codes satisfy reverse and reverse complement constraints.

1. Using induction on m, one can observe that the length $n = 2^{m+1}$ of $\mathscr{C}_{DNA}$ with the base case $G_{1,1}$. For $z \in \{w, 2+w, 3w, 2+3w\}$, the matrix $G_{1,m}$ is of type $\{1, m, 0, 0\}$, for $z \in \{2, 2+2w\}$, the matrix $G_{1,m}$ is of type $\{1, 0, m, 0\}$ and for $z \in \{2w\}$, the matrix $G_{1,m}$ is of type $\{1, 0, 0, m\}$, hence the result holds for number of codewords M (using Theorem 4.24). Due to the symmetry of the matrix $G_{1,m}$, note that any two codewords differ at least at 2^{m-1} positions. Hence, the minimum Gau distance d_{Gau} is $d2^{m-1}$, where

$d = min\{d_{Gau}(x,y) : x \neq y \text{ and } x, y \in < z >\}$. Therefore for all the different cases of zero divisors z, the results hold.

2. For each integer $m \geq 1$, the codeword **2+2w** is obtained by multiplying $2 + 2w$ to the row **1** of the matrix $G_{1,m}$ over the ring R, where **1** = $(1\,1\ldots1)$. From Lemma 4.21, the DNA code $\mathscr{C}_{DNA}$ is closed under complement. The closure of $\mathscr{C}_{DNA}$ with respect to reverse can be proved by using Lemma 4.18. If $\mathbf{x}_i$ is the i^{th} row of the matrix $G_{1,m}$, then for each $\mathbf{x}_i \in G_{1,m}$, observe that

$$\phi^{-1}(\phi(\mathbf{x}_i)^r) = \begin{cases} \mathbf{3} & \text{if } i = 1, \\ 3\mathbf{z} + \mathbf{x}_i & \text{if } i = 2, 3, \ldots, m+1. \end{cases}$$

Therefore, $\phi^{-1}(\phi(\mathbf{x}_i)^r) \in \mathcal{R}(1,m)$ because $\mathbf{x}_i$, **3**, $\mathbf{z} \in \mathcal{R}(1,m)$ for each $i = 1, 2, \ldots, m+1$. Thus from Lemma 4.18, the DNA code $\mathscr{C}_{DNA}$ is closed under reverse. Hence, by using Remark 4.10, the DNA code satisfies reverse and reverse complement constraints.

□

Example 4.31. *The DNA code $\mathscr{C}_{DNA}(n = 16, M = 8192, d_H = 4)$ of Reed-Muller Type code $\mathcal{R}(1,3)$ obtained by the generator matrix*

$$G_{1,3} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & w & w & w & w \\ 0 & 0 & w & w & 0 & 0 & w & w \\ 0 & w & 0 & w & 0 & w & 0 & w \end{pmatrix} \tag{4.18}$$

satisfies the reverse and reverse complement constraints.

One can obtain various DNA codes of type Reed-Muller using different m and z as shown in Table 4.8.

4.5 Improved Results on the DNA Codes

We achieve several improvements on the lower bound of size of the reversible and reversible-complement DNA codes. We compare the lower bound on $A_4^{RC,GC}(n,d,u)$

m	Zero divisor z	DNA Code $\mathscr{C}_{DNA}(n, M, d_H)$
1	2	$(4, 64, 2)$
2	2	$(8, 256, 4)$
3	2	$(16, 1024, 8)$
2	w	$(8, 1024, 2)$
3	w	$(16, 8192, 4)$

Table 4.8: DNA code $\mathscr{C}_{DNA}$ generated by Reed-Muller Type code for the zero divisor z with the different values of m. © [2018] IEEE. Reprinted, with permission, from Dixita Limbachiya, Krishna Gopal Benerjee, Bansari Rao and Manish K Gupta, *On DNA Codes using the Ring* $\mathbb{Z}_4 + w\mathbb{Z}_4$, In Proceedings of IEEE International Symposium on Information Theory (ISIT), pp. 2401-2405, 2018.

for some instances with n, d and u with the DNA codes constructed using the similar rings in the literature (see Table 4.9). Some of them are listed here. For $n = 8$, $d = 4$ and $u = 4$ from $\mathcal{R}(1, 2)$ (for the zero divisor $z = 2$, see Table 4.8), the lower bound obtained for $A_4^{RC,GC}(8, 4, 4)$ is *224* which is greater than 128, the lower bound observed in [22, 104]. We observe that the DNA codes derived in this thesis are better than the examples described in [78, 33, 104, 127]. For example, the DNA code $\mathscr{C}_{DNA}(n = 16, M = 8192, d_H = 4)$ which is better than $\mathscr{C}_{DNA}(n = 16, M = 28, d_H = 4)$ [92, 127]. Another instance is the DNA code $\mathscr{C}_{DNA}(n = 8, M = 256, d_H = 4)$ which is better than $\mathscr{C}_{DNA}(n = 8, M = 16, d_H = 4)$ [78, 104]. It can be concluded that the Reed-Muller type codes attains the lower bound on size M for $A_4^{GC}(n, d_H, u)$ and $A_4^{RC,GC}(n, d_H, u)$ [73] on the DNA code for some values of n, d_H and $u = n/2$. For example, if $z = 2$ or $z = 2w$, the first order Reed-Muller type code for $(n, d_H, u) = (4, 2, 2)$ bound achieving code with respect to $A_4^{GC}(n, d_H, u)$ and $A_4^{RC,GC}(n, d_H, u)$ respectively. Table 4.10 summarizes improvements of the results obtained and its comparison with the literature.

n	d_H	u	M Our Work	M Previous Work
16	4	-	8192	28 [92]
8	4	-	256	16 [104]
8	4	4	224	128 [104, 22]

Table 4.9: Comparison of our results for DNA codes. © [2018] IEEE. Reprinted, with permission, from Dixita Limbachiya, Krishna Gopal Benerjee, Bansari Rao and Manish K Gupta, *On DNA Codes using the Ring* $\mathbb{Z}_4 + w\mathbb{Z}_4$, In Proceedings of IEEE International Symposium on Information Theory (ISIT), pp. 2401-2405, 2018.

n/d		2	3	4	5	6	7	8	9	10
4	B	16	6	2						
	A	**64**	.	**4**						
6	B	1024	62	28	4	4				
	A	**1024**	16	.		**4**				
8	B	8192	643	128	30	**16**	2	**4**		
	A	4096	.	**256***	.	**16**		**4**		
10	B	2786	8064	2016	496	120	17	8	2	4
	A	**4096***	.	256	.	.	.	**8**	.	**4**
12	B	2734	2609	32640	4032	2016	120	31	12	4
	A	2048	.	**4096**	.	64	.	.	.	**4**

Table 4.10: Table of lower bounds for $A_4^{RC}(n, d_H)$ on DNA codes using rings. The entry A presents our results. The entry B presents the best results of previous study of DNA codes from rings. * is improvement by our method on the previous bound and bold are results matching the values of the bound with previous DNA codes from rings.

In the next section, we define the r^{th} order Reed-Muller type codes.

4.6 The r^{th} order Reed-Muller Type Codes

For the given zero divisor $z \in R$ and each integers r, m ($0 \leq r \leq m$), the r^{th} order Reed-Muller code $\mathcal{R}(r, m)$ over the ring R is defined by the generator matrix

$$G_{r,m} = \begin{pmatrix} G_{r,m-1} & G_{r,m-1} \\ 0 & G_{r-1,m-1} \end{pmatrix}, \ 1 \leq r \leq m-1$$

with

$$G_{m,m} = \begin{pmatrix} G_{m-1,m} \\ 0\,0\ldots\,0\,z \end{pmatrix}$$

and $G_{0,m} = (1\,1\ldots 1)$ is the all one matrix with length 2^m.

Theorem 4.32. *For the r^{th} order Reed-Muller code $\mathcal{R}(r, m)$ over the ring R, the DNA code $\mathscr{C}_{DNA} = \phi(\mathcal{R}(r, m))$ satisfies the reverse and reverse complement constraints. The*

(n, M, d_H) parameters of the DNA code $\mathscr{C}_{DNA}$ are $n = 2^{m+1}$,

$$M = \begin{cases} 2^{4b-3a} & if \quad z \in \{2w\}, \\ 2^{4b-2a} & if \quad z \in \{2, 2+2w\}, \\ 2^{4b-a} & if \quad z \in \{w, 2+w, 3w, 2+3w\}, \end{cases}$$

and

$$d_H = \begin{cases} 2^{m-r+1} & if \quad z \in \{2w, 2, 2+2w\}, \\ 2^{m-r} & if \quad z \in \{w, 2+w, 3w, 2+3w\}, \end{cases}$$

where $a = \sum_{i=0}^{r-1} \binom{m-1}{i}$ and $b = \sum_{i=0}^{r} \binom{m}{i}$.

Proof. The proof follows in two parts. The first part proves the parameters of the DNA code $\mathscr{C}_{DNA}$ and the second part proves the reverse and reverse complement constraints.

1. Using induction on m, one can observe that the length $n = 2^{m+1}$ of $\mathscr{C}_{DNA}$ with the base case $G_{1,1}$. Note that in the matrix $G_{r,m}$, the total number of rows which contain the zero divisor z is $b = \sum_{i=0}^{r} \binom{m}{i}$ and the total number of rows is $a = \sum_{i=0}^{r-1} \binom{m-1}{i}$. For $z \in \{w, 2+w, 3w, 2+3w\}$, the matrix $G_{r,m}$ is of type $\{b-a, a, 0, 0\}$, for $z \in \{2, 2+2w\}$, the matrix $G_{r,m}$ is of type $\{b-a, 0, a, 0\}$ and for $z \in \{2w\}$, the matrix $G_{r,m}$ is of type $\{b-a, 0, 0, a\}$, hence the result holds for number of codewords M. Due to the symmetry of the matrix $G_{r,m}$, note that any two distinct codewords are differ at least at 2^{m-1} positions. Hence, the minimum Gau distance d_{Gau} is $d2^{m-1}$, where $d = min\{d_{Gau}(x, y) : x \neq y \text{ and } x, y \in < z >\}$. Therefore for all the different cases of zero divisor z, the result holds for the distance by using Theorem 4.24.

2. For each integers r, m $(0 \leq r \leq m)$, the codeword **2+2w** is obtained by multiplying $2 + 2w$ to the row **1** of the matrix $G_{1,m}$ over the ring R, where **1** = $(1\ 1 \ldots 1)$. For the DNA code $\mathscr{C}_{DNA}$, the reverse constraint can be proved using induction on r. For any integer $m \geq 0$, the matrix $G_{0,m} = (1\ 1 \ldots 1)$ with 2^m columns and therefore $(3\ 3 \ldots 3) \in R(0, m)$. Using Lemma 4.18, the

DNA code $\phi(R(0,m))$ is closed under the reverse constraint. Now, assume that the DNA code $\phi(R(r-1,m))$ is closed under reverse for each integer $m \geq r-1$. For the given integer r, the reverse constraint of $\phi(R(r,m))$ can be proved using induction on m $(\geq r)$. For the base case $m = r$, we will prove that $\phi(R(r,r))$ is closed under reverse. Note that

$$G_{r,r} = \begin{pmatrix} G_{r-1,r} \\ 0\,0\ldots0\,z \end{pmatrix} = \begin{pmatrix} G_{r-1,r-1} & G_{r-1,r-1} \\ \mathbf{0} & G_{r-1,r-1} \end{pmatrix}$$

Using Lemma 4.41, $\phi(R(r,r))$ is closed under reverse. By recurrence construction of the matrix, each row of the matrix $G_{r-1,m-1}$ is the row of the matrix $G_{r,m-1}$. Using Lemma 4.41, the DNA code $\mathscr{C}_{DNA}$ is closed under reverse.

□

Example 4.33. *The DNA code $\mathscr{C}_{DNA}(n = 8, M = 1024, d_H = 2)$ of Reed-Muller Type code $\mathcal{R}(2,2)$ obtained by the generator matrix*

$$G_{2,2} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 \end{pmatrix} \tag{4.19}$$

is reversible and reversible-complement DNA code.

4.7 General Results

The results obtained on the reverse and reverse complement constraints for the families of DNA codes suggest the following general theorems and remarks.

For a positive integer k, let P be a matrix over the ring R with $4k$ length vector $(2\,2\ldots2) \in < P >_R$. For $i = 1,2,3,4$, all the four elements $z_i \in \{0, 2, 2w, 2+2w\}$ are distinct and all the four vectors $\mathbf{z}_i = (z_i\ z_i \ldots z_i)$ have same length k. Now,

consider the matrix

$$G = \begin{pmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \mathbf{z}_3 & \mathbf{z}_4 \\ & & P & \end{pmatrix}.$$

For the matrix G, parameters and constraints are discussed in the following theorems 4.34 and 4.36.

Theorem 4.34. *If the parameters of the DNA code $\phi(< P >_R)$ are $(8k,\ M^P,\ d_H^P)$ then the parameters of the DNA code $\phi(< G >_R)$ are $(8k,\ M^G,\ d_H^G)$, where $d_H^G \leq min\{4k,\ d^P\}$ and*

$$M^G = \begin{cases} M^P & \text{if } (z_1\ z_2\ z_3\ z_4) \in < P >_R, \\ 4M^P & \text{if } (z_1\ z_2\ z_3\ z_4) \notin < P >_R\ . \end{cases}$$

Proof. Both the matrices G and P have $4k$ number of columns so, by using Theorem 4.24, the length of a codeword of the DNA code $\phi(< G >_R)$ is $8k$. Let the matrix P be of type $\{k_0, k_1, k_2, k_3\}$. If $(\mathbf{z}_1\ \mathbf{z}_2\ \mathbf{z}_3\ \mathbf{z}_4) \in < P >_R$ then the matrix G is of type $\{k_0, k_1, k_2, k_3\}$ and if $(\mathbf{z}_1\ \mathbf{z}_2\ \mathbf{z}_3\ \mathbf{z}_4) \notin < P >_R$ then the matrix G is of type $\{k_0, k_1, k_2 + 1, k_3\}$. Hence, the result holds for the code size M. The minimum Gau distance for $< \mathbf{z}_1\ \mathbf{z}_2\ \mathbf{z}_3\ \mathbf{z}_4 >_R$, is $4k$ and the minimum Gau distance for $< P >_R$, is d_{Gau}^P. Hence, the minimum Gau distance for $< G >_R$ is bounded by $min\{4k,\ d^P\}$. Using Theorem 4.24, the result holds. □

Example 4.35. *For $k = 1, s = 2$ and the matrix*

$$P_{2\times 4} = \begin{pmatrix} 0 & 0 & 2w & 2w \\ 2 & 2 & 2 & 2 \end{pmatrix} \tag{4.20}$$

then a DNA code $\mathscr{C}_{DNA}(n = 8, M^P = 8, d_H^P = 4)$ is generated by the matrix P. The matrix

$$G = \begin{pmatrix} 0 & 2 & 2w & 2+2w \\ 0 & 0 & 2w & 2w \\ 2 & 2 & 2 & 2 \end{pmatrix}, \tag{4.21}$$

generates the DNA code $\mathscr{C}_{DNA}(n = 8, M^G = 32, d_H^G = 4)$.

Theorem 4.36. *If the DNA code $\phi(< P >_R)$ is closed under the reverse constraint then the DNA code $\phi(< G >_R)$ will be closed under the reverse and reverse complement constraints.*

Proof. For the complement constraint, note that $(2+2w\ 2+2w \ldots 2+2w) \in < G >_R$ because $(2\ 2 \ldots 2) \in < P >_R$. Using Lemma 4.21, the DNA code $\phi(< G >_R)$ is closed under complement constraint. For the reverse constraint, consider $\phi^{-1}(\phi(\mathbf{z}_1\ \mathbf{z}_2\ \mathbf{z}_3\ \mathbf{z}_4)^r) = (\mathbf{z}_4\ \mathbf{z}_3\ \mathbf{z}_2\ \mathbf{z}_1)$ because $\phi^{-1}(\phi(z)^r) = z$, for any $z \in \{0, 2, 2w, 2+2w\}$. For the elements $0, 2, 2w, 2+2w$ of the ring R, note that the sum of any two distinct elements is equal to the sum of the another two distinct elements. If the length of the vectors $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4$ is same then $\mathbf{z}_1 + \mathbf{z}_4 = \mathbf{z}_2 + \mathbf{z}_3$. If $\mathbf{z}_1 + \mathbf{z}_4 = \mathbf{z}_2 + \mathbf{z}_3 = \mathbf{z}$ then $(\mathbf{z}_1\ \mathbf{z}_2\ \mathbf{z}_3\ \mathbf{z}_4) + (\mathbf{z}_4\ \mathbf{z}_3\ \mathbf{z}_2\ \mathbf{z}_1) = (\mathbf{z}\ \mathbf{z}\ \mathbf{z}\ \mathbf{z})$ for some $z \in \{2, 2w, 2+2w\}$, where $\mathbf{z} = (z\, z \ldots z)$ is a k length codeword. But, $(2\ 2 \ldots 2) \in < P >_R$ so $(\mathbf{z}\ \mathbf{z}\ \mathbf{z}\ \mathbf{z}) \in < P >_R$ and therefore $(\mathbf{z}_4\ \mathbf{z}_3\ \mathbf{z}_2\ \mathbf{z}_1) = (\mathbf{z}\ \mathbf{z}\ \mathbf{z}\ \mathbf{z}) - (\mathbf{z}_1\ \mathbf{z}_2\ \mathbf{z}_3\ \mathbf{z}_4) \in < G >_R$ which directs $\phi^{-1}(\phi(\mathbf{z}_1\ \mathbf{z}_2\ \mathbf{z}_3\ \mathbf{z}_4)^r) \in < G >_R$. By using Lemma 4.18, the DNA code $\phi(< G >_R)$ is closed under reverse constraint. Thus by using Remark 4.10, the results holds. □

Example 4.37. *For $k = 1, s = 3$ and the matrix*

$$P_{3\times 4} = \begin{pmatrix} 0 & 0 & 2w & 2w \\ 2 & 2 & 2 & 2 \\ 2w & 2w & 0 & 0 \end{pmatrix} \tag{4.22}$$

then a DNA code $\mathscr{C}_{DNA}(n = 8, M^P = 8, d_H^P = 4)$ is generated by the matrix P. The matrix

$$G = \begin{pmatrix} 0 & 2 & 2w & 2+2w \\ 0 & 0 & 2w & 2w \\ 2 & 2 & 2 & 2 \\ 2w & 2w & 0 & 0 \end{pmatrix}, \tag{4.23}$$

generates the DNA code $\mathscr{C}_{DNA}(n = 8, M^P = 32, d_H^P = 4)$.

Lemma 4.38. *For a matrix G_1 over the ring R, if the DNA code $\phi(< G_1 >_R)$ is closed under reverse constraint then the DNA code $\phi(< G_k >_R)$ will be closed under reverse constraint, where $G_k = (G_{k-1}\ G_1)$ for $k > 1$.*

Proof. For each $\mathbf{x} \in < G_1 >_R$, $\phi^{-1}(\phi(\mathbf{x})^r) \in < G_1 >_R$. Note that, the matrix G_k is k times block repetition of the matrix G_1, so $(\mathbf{x}\ \mathbf{x}\ \ldots\ \mathbf{x}) \in G_k$ and $(\phi^{-1}(\phi(\mathbf{x})^r)$

$\phi^{-1}(\phi(\mathbf{x})^r) \ldots \phi^{-1}(\phi(\mathbf{x}))^r) \in G_k$, for every $\mathbf{x} \in G_1$. But, $(\phi^{-1}(\phi(\mathbf{x})^r)\ \phi^{-1}(\phi(\mathbf{x})^r) \ldots \phi^{-1}(\phi(\mathbf{x}))^r) = \phi^{-1}(\phi(\mathbf{x}\ \mathbf{x} \ldots \mathbf{x})^r)$ which directs $\phi^{-1}(\phi(\mathbf{x}\ \mathbf{x} \ldots \mathbf{x})^r) \in < G_k >_R$. Using Remark 4.6, the result holds. □

Lemma 4.39. *For a matrix G_1 over the ring R, if* $\mathbf{2+2w} = (2 + 2w\ 2 + 2w \ldots 2 + 2w) \in < G_1 >_R$ *then* $(\mathbf{2+2w}\ \mathbf{2+2w} \ldots \mathbf{2+2w}) \in < G_k >_R$, *where $G_k = (G_{k-1}\ G_1)$ for $k > 1$. Hence, if the DNA code $\phi(< G_1 >_R)$ is closed under complement constraint then the DNA code $\phi(< G_k >_R)$ will be closed under complement constraint.*

Proof. The DNA code $\phi(< G_1 >_R)$ is a complement code (by Lemma 4.21). Hence, for $\mathbf{2+2w} \in < G_k >_R$, it is closed under complement constraints by using the result of Lemma 4.21. □

An example of Lemmas 4.38 and 4.39 is shown in Example 4.40.

Example 4.40. *For $k = 2$, if the reversible and reversible-complement DNA code $\phi(< G_1 >_R) = (n = 4, M = 4, d_H = 4)$ is obtained by the generator matrix*

$$G_1 = \begin{pmatrix} 2 & 2 \\ 2+2w & 2+2w \end{pmatrix} \tag{4.24}$$

then

$$G_2 = \begin{pmatrix} 2 & 2 & 2 & 2 \\ 2+2w & 2+2w & 2+2w & 2+2w \end{pmatrix} \tag{4.25}$$

generates $\phi(< G_2 >_R) = \{AAAAAAAA, TTTTTTTT, GGGGGGGG, CCCCCCCC\}$ is reversible and reversible-complement code with $(n = 8, M = 4, d_H = 8)$.

One can observe the following general result for the r^{th} order Reed-Muller type code.

Lemma 4.41. *Let G and H be two matrices over the ring R such that both the DNA codes $\phi(< G >_R)$ and $\phi(< H >_R)$ are closed under reverse constraint. If each row of H is a row of G then the DNA code $\phi(< T >_R)$ will be closed under reverse, where*

$$T = \begin{pmatrix} G & G \\ \mathbf{0} & H \end{pmatrix}.$$

Proof. Any row of T is the row of either matrix $(G\ G)$ or the matrix $(\mathbf{0}\ \ H)$. Now consider two cases for this:

Case: 1 If $(\mathbf{x}\,\mathbf{x})$ is the row of the matrix $(G\ G)$ then $\mathbf{x}$ will be the row of the matrix G. Using Lemma 4.18, $\phi^{-1}(\phi(\mathbf{x})^r) \in < G >_R$ and therefore $(\phi^{-1}(\phi(\mathbf{x})^r)\ \ \phi^{-1}(\phi(\mathbf{x})^r)) \in < G\ G >_R$ by using Lemma 4.38 for each row of the matrix G.

Case: 2 If $(\mathbf{0}\,\mathbf{y})$ is the row of the matrix $(\mathbf{0}\ H)$ then $\mathbf{y}$ will be the row of H. By using Lemma 4.18, $\phi^{-1}(\phi(\mathbf{y})^r) \in < H >_R$. Therefore, $(\mathbf{0}\ \phi^{-1}(\phi(\mathbf{x})^r)) \in < \mathbf{0}\ H >_R \subseteq < T >_R$. But $\mathbf{y}$ is also the row of G so from the case 1, $(\phi^{-1}(\phi(\mathbf{y})^r)\ \ \phi^{-1}(\phi(\mathbf{y})^r)) \in < G\ G >_R \subseteq < T >_R$. Thus $(\phi^{-1}(\phi(\mathbf{y})^r)\ \ \mathbf{0}) \in < T >_R$ for each row $\mathbf{y}$ of the matrix H. Now by the case 1 and case 2, it is concluded that $\phi^{-1}(\phi(\mathbf{t})^r) \in < T >$ for each row $\mathbf{t}$ of the matrix T. Hence, by using Lemma 4.18, the DNA code $\phi(< T >_R)$ is closed under reverse. □

Example 4.42. *If the matrices G and H over the ring R are given as*

$$G = \begin{pmatrix} w & 1 \\ 2w & 2w \\ 2 & 2 \end{pmatrix} \tag{4.26}$$

and

$$H = \begin{pmatrix} 2 & 2 \\ 2w & 2w \end{pmatrix} \tag{4.27}$$

then

$$T = \begin{pmatrix} w & 1 & w & 1 \\ 2 & 2 & 2 & 2 \\ 2w & 2w & 2w & 2w \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2w & 2w \end{pmatrix}, \tag{4.28}$$

by Lemma 4.18 on closure of reversible code, $\phi(< T >)$ is reversible code.

Remark 4.43. *For any matrix G over the ring R, the DNA code* $\phi\left(\left\langle \begin{pmatrix} G \\ \mathbf{2+2w} \end{pmatrix} \right\rangle_R\right)$ *is closed under complement, where the vector* $\mathbf{2+2w} = (2+2w\ 2+2w \ldots 2+2w)$.

4.8 DNA Codes from Rings R_1, R_2 and R_3

In this section, we propose DNA codes over rings R_1, R_2 and R_3. It is interesting to observe that the relation between the elements of rings and the DNA alphabets over these rings. It is exciting to define the Gau map ϕ over these rings. In order to give a distance preserving map, the Gau distance should be defined on rings. We can also define an isometry (distance preserving map) between the codes over these rings and DNA codes. One can also define Gau distances $d_{Gau}^{R_1}$, $d_{Gau}^{R_2}$, $d_{Gau}^{R_3}$ on rings R_1, R_2 and R_3, respectively.

The Gau distance $d_{Gau}^{R_1}$, $d_{Gau}^{R_2}$, $d_{Gau}^{R_3}$ is a metric on the elements of the ring R_1, R_2, R_3, respectively. One can use the Gau distance for rings R_1, R_2, R_3 as follows.

Let $\Gamma_{DNA} = \{AA, AT, TT, TA, CC, CG, GG, GC\}^n$ denote the 2-mer DNA codewords.

To construct the DNA codes using rings R_1, R_2, R_3, a relation is required between the elements of the rings and the 2-mer DNA codewords Γ_{DNA}. To define a distance on rings, the elements of the ring and the 2-mer Γ_{DNA} can be arranged in a manner (see the matrix $\mathscr{M}_{R_1}$ ($\mathscr{M}_{R_2}$) in Equation 4.29 (4.32) respectively) such that the Hamming distance d_H between any two distinct pair of DNA nucleotides in the same row or same column is 1, otherwise it is 2. This motivates us to define a Gau distance on elements of rings. For $x, y \in R_1$, let $x = m_{i,j} \in \mathscr{M}_{R_1}$, $y = m_{i',j'} \in \mathscr{M}_{R_1}$ for some $(i,j), (i',j') \in \{(0,0), (0,3), (1,1), (1,2), (2,1), (2,2), (3,0), (3,3)\}$ then, the Gau distance $d_{Gau}^{R_1}$ can be defined as in Equation 4.2.

$$\mathscr{M}_{R_1} = \begin{array}{c} \\ A \\ G \\ C \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & G & C & T \end{array} \\ \begin{pmatrix} 1 & - & - & 1+w_1 \\ - & 2w_1 & w_1 & - \\ - & 3w_1 & 0 & - \\ 1+3w_1 & - & - & 1+2w_1 \end{pmatrix} \end{array} \tag{4.29}$$

One can also define the Gau distance as follows:

For the ring R_1, the set of zero divisors $Z = \{0, w_1, 2w_1, 3w_1\}$ and the set of

units $U = \{1, 1+w_1, 1+2w_1, 1+3w_1\}$. For both x and $y \in Z$ or both x and $y \in U$ we have,

$$d_{Gau}^{R_1}(x,y) = \begin{cases} 0 & if\ x = y, \\ 1 & if\ x \neq y\ and\ x+y \in \{w_1, 3w_1\}, \\ 2 & otherwise. \end{cases} \tag{4.30}$$

For any $x \in Z$ and $y \in U$,

$$d_{Gau}^{R_1}(x,y) = \begin{cases} 0 & if\ x = y, \\ 2 & if\ x \neq y\ and\ x+y \in U. \end{cases} \tag{4.31}$$

$$\mathscr{M}_{R_2} = \begin{array}{c} \\ A \\ G \\ C \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & G & C & T \end{array} \\ \begin{pmatrix} 0 & - & - & 1 \\ - & w_2 & 1+w_2 & - \\ - & 3+w_2 & 2+w_2 & - \\ 3 & - & - & 2 \end{pmatrix} \end{array} \tag{4.32}$$

For the ring R_2, the set of zero divisors $Z = \{0, 2, w_2, 2+w_2\}$ and the set of units $U = \{1, 3, 1+w_2, 3+w_2\}$. For both x and $y \in Z$ or both x and $y \in U$ we have,

$$d_{Gau}^{R_2}(x,y) = \begin{cases} 0 & if\ x = y, \\ 2 & if\ x \neq y\ and\ x+y \in Z. \end{cases} \tag{4.33}$$

For any $x \in Z$ and $y \in U$,

$$d_{Gau}^{R_2}(x,y) = \begin{cases} 0 & if\ x = y, \\ 1 & if\ x \neq y\ and\ x + y \in \{1,3\}, \\ 2 & otherwise. \end{cases} \tag{4.34}$$

For the ring R_3, the set of zero divisors $Z = \{0, w_3\}$ and the set of units $U = \{1, 1 + w_3\}$. For any x and y in R_3

$$d_{Gau}^{R_3}(x,y) = \begin{cases} 0 & if\ x = y, \\ 1 & if\ x \neq y. \end{cases} \tag{4.35}$$

Remark 4.44. *It is easy to observe that Gau distance $d_{Gau}^{R_3}$ over the ring R_3 coincides the Hamming distance $d_H^{R_3}$ over the ring R_3.*

The map ϕ_1 (ϕ_2) is defined from the elements of R_1 (R_2) to the DNA strands of length 2. For ϕ_3, the straight forward map is $A \to 0$, $T \to 1$, $G \to w_3$, $C \to 1 + w_3$ such that $x^c = x + 1$. The maps ϕ_1 and ϕ_2 are given in Tables 4.11 and 4.12 respectively.

Ring element x	DNA image $\phi_1(x)$	Ring element x	DNA image $\phi_1(x)$
0	CC	1	AA
w_1	GC	$1 + w_1$	AT
$2w_1$	GG	$1 + 2w_1$	TT
$3w_1$	CG	$1 + 3w_1$	TA

Table 4.11: A bijective mapping ϕ_1: $R_1^n \to \Gamma_{DNA}^n$ is given such that $\phi^{-1}(\phi(x)^c) = x + 2w_1$ and $x + \phi^{-1}(\phi(x)^r) = 0$.

Theorem 4.45. *$\phi_1 : (R_1^n, d_{Gau}^{R_1})$ to $(\Gamma_{DNA}^n, d_H^{R_1})$ is a distance preserving map.*

Proof. We prove it for $n = 1$. The higher case is obvious. Consider the following cases.

- Case 1: For $x, y \in R_1$, if $x = y$ then $d_{Gau}^{R_1}(x,y) = 0 \implies d_H^{R_1}(\phi_1(x), \phi_1(y)) = 0$.

- Case 2: For $x \neq y$, if $x + y \in \{w_1, 3w_1\}$, then $d_{Gau}^{R_1}(x, y) = 1$. Observe that $x + y = w_1$ if

$$\begin{aligned} x &\in \{0, 1, 2w_1, 1 + 2w_1\} \\ y &\in \{w_1, 1 + w_1, 3w_1, 1 + 3w_1\} \text{ then} \\ \phi_1(x) &\in \{CC, AA, GG, TT\} \\ \phi_1(y) &\in \{GC, AT, CG, TA\}. \end{aligned}$$

  Hence $d_H^{R_1}(\phi_1(x), \phi_1(y)) = 1$. Similarly, for $x + y = 3w_1$ if

$$\begin{aligned} x &\in \{0, w_1, 1 + w_1, 1\} \\ y &\in \{3w_1, 2w_1, 1 + 2w_1, 1 + 3w_1\} \text{ then} \\ \phi_1(x) &\in \{CC, GC, AT, AA\} \\ \phi_1(y) &\in \{CG, GG, TT, TA\}. \end{aligned}$$

  Hence, $d_H^{R_1}(\phi_1(x), \phi_1(y)) = 1$.

- Case 3: For $x \in \{0, w_1, 2w_1, 3w_1\}$, $y \in \{1, 1 + w_1, 1 + 2w_1, 1 + 3w_1\}$ if $x + y \in \{1, 1 + w_1, 1 + 2w_1, 1 + 3w_1\}$ then $d_{Gau}^{R_1}(x, y) = 2$. Note that $x + y = 1$ if

$$\begin{aligned} x &\in \{0, w_1, 2w_1, 3w_1\} \\ y &\in \{1, 1 + 3w_1, 1 + 2w_1, 1 + w_1\} \text{ then} \\ \phi_1(x) &\in \{CC, GC, GG, CG\} \\ \phi_1(y) &\in \{AA, AT, TT, TA\}. \end{aligned}$$

  Hence, $d_H^{R_1}(\phi_1(x), \phi_1(y)) = 2$. Similarly, one can proof for $x + y \in \{1 + w_1, 1 + 2w_1, 1 + 3w_1\}$

Thus by observing all these cases, it is verified that $\phi_1 : (R_1^n, d_{Gau}^{R_1})$ to $(\Gamma_{DNA}^n, d_H^{R_1})$ is an isometry. □

Lemma 4.46. *For any row $\boldsymbol{x}$ of G_{R_1} over the ring R_1, the DNA code $\phi(< G_{R_1} >)$ is closed under reverse if and only if $\phi^{-1}(\phi(\boldsymbol{x})^r) \in < G_{R_1} >$, the row span of G_{R_1} over R_1.*

Proof. The proof is similar to the proof of Lemma 4.18. □

Lemma 4.47. *For a matrix G_{R_1} over the ring R_1, the DNA code $\phi(< G_{R_1} >)$ is closed under complement if and only if $2w_1 \in < G_{R_1} >$, where $2w_1$ is a string with each element $2w_1$.*

Proof. The proof is similar to the proof of Lemma 4.21. □

The map ϕ_2 from the elements of the ring R_2 to the DNA nucleotides is described in Table 4.12. The proof of Theorems 4.48 and 4.51 is similar to the proof of Theorem 4.45.

Ring element x	DNA image $\phi_2(x)$	Ring element x	DNA image $\phi_2(x)$
0	AA	w_2	GG
1	AT	$1 + w_2$	GC
2	TT	$2 + w_2$	CC
3	TA	$3 + w_2$	CG

Table 4.12: A bijective mapping ϕ_2: $R_2^n \to \Gamma_{DNA}^n$ is given such that $\phi^{-1}(\phi(x)^c) = x + 2$ and $x + \phi^{-1}(\phi(x)^r) = 0$

Theorem 4.48. *$\phi_2 : (R_2^n, d_{Gau}^{R_2})$ to $(\Gamma_{DNA}^n, d_H^{R_2})$ is a distance preserving map.*

Lemma 4.49. *For any row $\boldsymbol{x}$ of G_{R_2} over the ring R_2, the DNA code $\phi(< G_{R_2} >)$ is closed under reverse if and only if $\phi^{-1}(\phi(\boldsymbol{x})^r) \in < G_{R_2} >$, the row span of G_{R_2} over R_2.*

Proof. The proof is similar to the proof of Lemma 4.18. □

Lemma 4.50. *For a matrix G_{R_2} over the ring R_2, the DNA code $\phi(< G_{R_2} >)$ is closed under complement if and only if $\boldsymbol{2} \in < G_{R_2} >$, where $\boldsymbol{2}$ is a string with each element 2.*

Proof. The proof is similar to the proof of Lemma 4.21. □

Theorem 4.51. *$\phi_3 : (R_3^n, d_{Gau}^{R_3})$ to $(\Sigma_{DNA}^n, d_H^{R_3})$ is a distance preserving map.*

Example 4.52. *A DNA code* $\mathscr{C}_{DNA}(n = 16, M = 512, d_H^{R_2} = 8)$ *is obtained by the generator matrix*

$$G_{R_2} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & w_2 & w_2 & w_2 & w_2 \\ 0 & 0 & w_2 & w_2 & 0 & 0 & w_2 & w_2 \\ 0 & w_2 & 0 & w_2 & 0 & w_2 & 0 & w_2 \end{pmatrix} \tag{4.36}$$

from the ring $R_2 = \mathbb{Z}_4 + w_2\mathbb{Z}_2$ *using the map* ϕ_2.

This chapter gives an interesting construction of the DNA codes using the ring $R = \mathbb{Z}_4 + w\mathbb{Z}_4$, where $w^2 = 2 + 2w$. A new distance called the Gau distance on the ring R is introduced. We have also proposed a new distance preserving Gau map ϕ from the elements of the ring R to all the DNA 2-mers. Some of them are optimal with respect to the bounds and are better than the DNA codes from rings obtained in the literature. Also, different rings R_1, R_2 and R_3 and their corresponding Gau maps are discussed.

In the next chapter, the literature on DNA data storage systems is discussed.

CHAPTER 5

On DNA based Data Storage Systems

Our own genomes carry the story of evolution, written in DNA, the language of molecular genetics, and the narrative is unmistakable.

-Kenneth R. Miller [18]

Data storage is an ancient practice performed by humans to pass the information from one generation to another. Stemming from the early day's storage medium to the modern days distributed cloud data storage [32] and Graphene-quantum-dot data storage [67]; there is a drastic advancement in the data storage devices. With the extensive use of social networking and cloud computing, there is a paradigm shift in the volume of data produced. It is predicted that in the era of an internet of things (IoT), the future unit of the big data will be Geopbyte (10^{30} bytes), which highlights a big concern of storing and maintaining a rapid growth of the data that enforces the data storage experts to design a new architecture to store the data [77]. Digital data storage devices are expensive, consume a tremendous amount of energy and releases much heat that is harmful to the environment. The existing data storage media needs to be maintained regularly and are prone to decay. Scientists are trying to miniaturize the size of silicon chips up to many folds, but this makes it more expensive. Alternatively, researchers instigated the use of source from nature to preserve the data which gave rise to the field of biomolecular storage. The biomolecular storage system [10] is an art of storing and retrieving the information to and from the natural medium using bio-

molecules. Many researchers for data storage [81, 89] have explored DNA, RNA and proteins. Looking at the success stories of the biomolecular data storage, in particular, the DNA based data storage systems is considered to be a future data storage technology.

The concept of data storage includes the representation of the data such that it allows the mapping of binary $0's$ and $1's$ to individual states. For instance, $0's$ and $1's$ are represented as pits and lands respectively on CD. Specific properties as storage capacity, data access rate, read/write speed, portability, durability and reliability characterize data storage devices. In order to use bio-molecules as the storage medium, it should have the following properties.

- Monomer units that can encode bits is essential for data storage. For DNA (Deoxyribonucleic acid) based storage systems, nucleotides A, T, G, C serve as monomer units to encode 0's and 1's. For protein, amino acids act as monomer units that can be mapped to binary digits. For bacterial storage, two states of genes as on and off condition of genes can be considered as the flipping of 0 to 1 and vice-versa.

- Coded sequence of biomolecular units used to encode the data must have writing and reading technology. For instance, data stored in DNA can be written and read by using DNA synthesis and sequencing technology respectively. For protein, peptide synthesis and sequencing technology are available. To use any bio-molecule as information storage, synthesis and sequencing must be practically well developed. Bacterial cloning and Recombinant DNA technology may be used for bacterial storage.

These properties of biomolecules such as DNA, RNA and protein enables them to perform as the storage medium. DNA in each cell of human encodes the information which is processed via genes and proteins using molecular machinery. DNA being the oldest data storage medium, it is obvious to think of using DNA as a digital data storage medium.

5.1 DNA as Storage Device

Different channel models for DNA storage has been proposed in [15, 63, 72, 76, 139]. These DNA data storage systems have different properties as described in Figure 5.2. An archival DNA data storage model is presented in Figure 5.1. There is an encoder which converts binary data to DNA nucleotides using error correcting codes. Data is converted to DNA units A, T, G and C. For instance; one can represent each base pair by using 2 bits, ($00 \rightarrow$ AT, $01 \rightarrow$ GC, $10 \rightarrow$ TA and $11 \rightarrow$ CG). Coding potential is the maximum amount of bits encoded per nucleotide. The theoretical limit of coding potential is 2 bits per nucleotide. Different kinds of encoding schemes [43, 44, 107, 119] are used as differential coding [47], Reed-Solomon codes [48], XOR encoding [17], fountain codes [37] for encoding the data. Next, DNA synthesis and sequencing channel include DNA synthesis, amplification, storage and DNA sequencing. Net information density is the ratio of input data to a total number of DNA bases used to encode the data (excluding the adapter sequences). In the final step, the decoder retrieves the original data from DNA by decoding methods. The physical amount of information stored in DNA is measured by DNA data storage capacity that indicates the number of bytes divided by DNA bases used to decode the stored data. There are chances of errors during DNA synthesis and sequencing channel. Most common errors like substitution, insertion and deletion of the base occur. From these, substitution has the higher probability than others. In this thesis, we propose codes for DNA storage that can correct substitution errors (described in Chapter 6).

In this chapter, we summarize DNA data storage methods proposed in the literature.

DNA bases data storage has few pieces of evidence in the history. Microvenus project was initiated by Joe Davis to convert an image in DNA that alludes to the idea of storing abiotic data in DNA. Microvenus [27], a small organism comprises a short piece of synthetic DNA used to encode visual icon in the bacteria *E.coli*. In Clelland encoding models [9, 26], microdots were used to cipher the data in human genomic DNA. By using simple DNA coding schemes many researchers

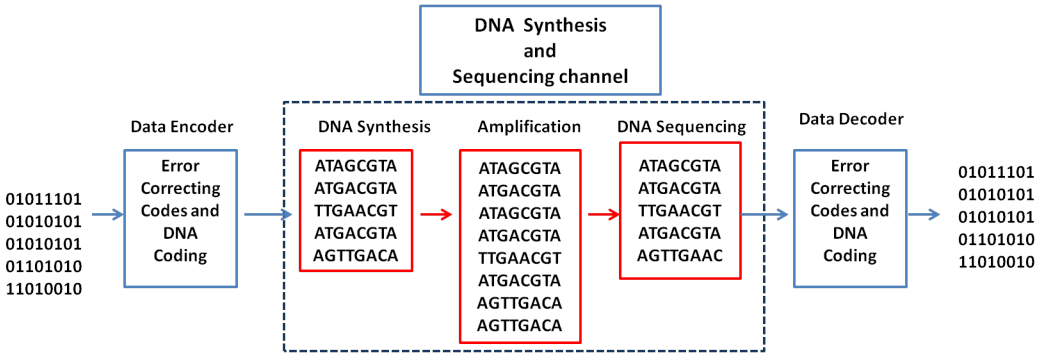

Figure 5.1: Model for Archival DNA storage channel. Archival DNA storage channel is divided into two parts. First is the data encoding channel that includes encoding the data into DNA sequences using encoding methods with error correction which help to detect and correct errors in the data encoded DNA sequences. Second is the storage channel that includes the reading and writing of the data on DNA using DNA synthesis and sequencing technologies.

have encoded English alphabets, mathematical expressions [136], Latin text and simple musical notations [4] to DNA [8].

Although the pioneering work laid the cornerstone for storing data to DNA, each one was successful on a small scale by encoding small bits of data. Later, the large-scale DNA data storage systems were developed. Some of the primary encoding approaches proposed for DNA-based information storage systems are described in this chapter.

The G. Church *et al.* in 2012 at Harvard University did the first successful work using a next-generation synthesis and sequencing technology. Their team proposed an efficient data encoding algorithm (1 bit per base) into a fixed length of DNA chunks (99 bases). In writing and reading DNA, 10 bits error occurred from 5.27 MB. It has the limitation of lacking error correction scheme that was taken care of by N. Goldman *et al.* [47] in 2013 by including error correction writing and reading the data. Their group used base 3 encoding Huffman coding (trits 0, 1 and 2), where each ternary codeword is of length 5 or 6 trits. In this, ternary Huffman was converted to the DNA code. DNA was divided into chunks. Four fold redundancy was included by overlapping 75 bases for each DNA information chunks that can help to recover the data loss which may occur during synthesis and sequencing DNA. As proof of concept, they used four different file types (739

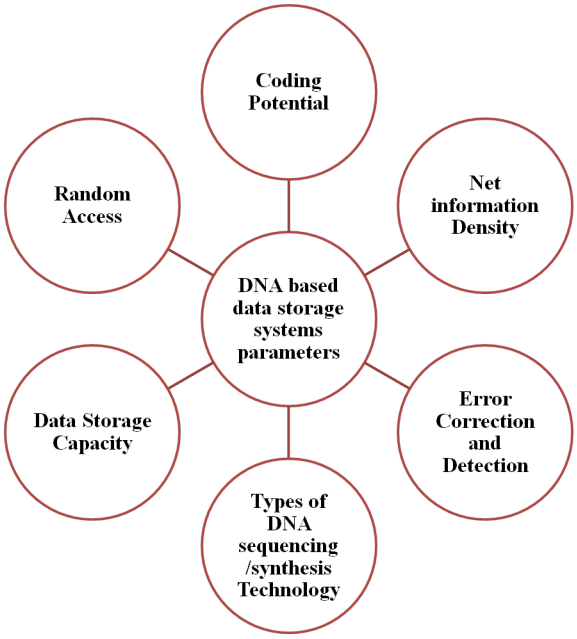

Figure 5.2: DNA Data Storage System Properties

kilobytes file size) and achieved 2.2 PB/g DNA storage capacity.

Over the extended period, data encoded DNA may get damaged in an inappropriate condition, for that matter, a long-term data storage in DNA was demonstrated by Grass et al. [48] that showcased the storage of DNA for long-term in a sphere. To witness long-term storage of DNA and the DNA stability, researchers have developed chemical based [48] method to encapsulate DNA into glass sphere and preserve it from environmental damage for long-term archival. They used very prominent error correcting code Reed Solomon (RS) codes, which are used in digital storage devices like CD and DVD.

All the above methods are archival DNA storage systems and do not allow random access of data from the file. Also, all the described methods are only write-once read-once methods. To overcome these issues, Yazdi *et al.* [128] and J. Bornolt *et al.* [17] proposed re-writable and random access DNA based data storage systems using prefix-synchronized and XOR encoding respectively, in which they used unique mutually uncorrelated addresses by which data can be randomly accessed. Recently, large scale random access method was proposed by L. Organick *et al.* in [96].

I. Holmes introduced the modular encoding for DNA storage in which non-

repeating DNA codes applicable to DNA storage is designed [61]. Also, an efficient and two-dimensional interleaved error-correcting code (ECC) scheme which can correct all the types of error is developed by M. Blawat *et al.* in [15]. W. L. Hughes *et al.* in [139] demonstrated a DNA data storage system terms as Nucleic Acid Memory (NAM). A compelling article on DNA as memory device is elucidated by introducing the properties like Read/write latency, a retention and volumetric density for DNA. E. Marcotte *et al.* in [5] introduced method to encode the data in DNA by defining a Levenshtein distance less than equal to 3 between the DNA codewords. Advancement in the DNA sequencing encouraged the researchers to develop the error-free portable DNA data storage system [137], which employs the most advanced nanopore DNA sequencing for reading the DNA encoded data.

In the past year, researchers have introduced the concept of DNA fountain [37] which use fountain codes to encode the data. Fountain Code is symbolic to a fountain which supplies endless data packets from senders end to a receiver just as water droplets (encoded data) which fill the empty bucket. The encoding method achieved highest net information density per nucleotide as 1.57 bits/nucleotide. This method recorded the highest data storage capacity of 214 Petabyte/gram of DNA. A fundamental theoretical limits have been discussed by Kanan *et al*. in [110].

Very recently, R. Heckel *et al.*, studied the distribution of error probabilities at the different levels of DNA data storage systems by analyzing the data sets of DNA data storage experiments from different researchers [109].

It is concluded that errors occur at each level of DNA storage and the source of errors are highly subjected to the kind of processes which is used for reading/writing the data on DNA. A high physical redundancy of the data encoded in DNA is required for reliable data storage, but this decreases the information density. Hence to have a better trade-off, it is recommended to use better error correcting schemes that can correct the errors.

This motivated us to develop codes for DNA storage such that it can avoid high GC-weight and long runlengths in a DNA codeword.

CHAPTER 6

Codes for DNA based Data Storage Systems

Human DNA is like a computer program but far, far more advanced than any software we have ever created.

Bill Gates

The Road Ahead, page 228 (Viking, Penguin Group, 1996, Revised Edition [18]

In this chapter, codes for archival DNA data storage systems are introduced. The objective of the DNA data storage system is to design the encoding methods to convert the digital data (binary codes) to DNA sequences (quaternary codes) such that it can correct the maximum number of errors (increasing the data resilience). Two methods for encoding the data in the DNA are proposed in this dissertation. The first method uses a constrained codes which is described in section 6.1. The constrained codes proposed for DNA data storage is published in [82] [1]. The second method is a DNA Golay subcode for DNA data storage which is discussed in the Section 6.2. The section 6.2 is published in [79] [2].

[1] © [2018] IEEE. Reprinted, with permission, from Dixita Limbachiya, Manish K. Gupta, and Vaneet Aggarwal, *Family of Constrained Codes for Archival DNA Data Storage, accepted in IEEE Communication Letters, July 2018, Early Access, doi:10.1109/LCOMM.2018.2861867.*

[2] © [2018] IEEE. Reprinted, with permission, from Dixita Limbachiya, Vijay Dhameliya, Madhav Khakhar, and Manish K. Gupta, *On Optimal Family of Codes for Archival DNA storage*, In Proceedings of IEEE Seventh International Workshop on Signal Design and its Applications in Communications (IWSDA), pp. 123-127. 2015.

6.1 Constraint Codes Method

Many researchers have proposed different encoding schemes for DNA based data storage such that it satisfies either of no-runlength or GC-weight constraints. Specifically, DNA codes with the no-runlength constraint for DNA based data storage have been studied in [16, 47, 48, 64, 79]. Although the DNA codes proposed in [73] have included GC-weight for DNA codewords but it did not consider the no-runlength constraint for DNA codes. A fountain code based methods for DNA data storage were proposed by Y. Erlich and D. Zielinski that developed DNA codes with runlength ≤ 3 and GC-weight constraints [37]. However, there is no evidence in the literature for such theoretical bounds on DNA codes for both these constraints.

Therefore in this work, we develop DNA codewords with both these constraints. We present a lower bound on number of codewords by considering DNA codewords with these constraints and a particular distance. The proposed altruistic algorithm first forms a list of all codewords of fixed GC-weight with the no-runlength constraint for which an exact characterization has been given.

6.1.1 Constrained Coding for the DNA Storage

The constrained coding method is considered to generate DNA codes with both no-runlength and fixed GC-weight u constraints. In [47] and [79], the ternary codewords were used to encode the data in the DNA. The idea of an altruistic algorithm from [51] is used, which results in a different set of codes with length n and minimum Hamming distance d. The modified algorithm generates the DNA codes with the no-runlength constraint and a fixed GC-weight using the quaternary encoding. Once the DNA codes are generated, the codebook can be used to encode the data.

Algorithm 1 first lists all the codewords to develop families of constrained DNA codes with different length n and minimum Hamming distance d satisfying no-run lengths and a fixed GC -weight $u = \lfloor n/2 \rfloor$. Next, until the codebook has codewords with a minimum Hamming distance d, an altruistic algorithm greedily

Algorithm 1 Altruistic Algorithm to generate DNA codes

Input: DNA code length n, GC-weight u, minimum Hamming distance d
Output: Altruistic DNA codebook

1 Enumerate $4 \times 3^{n-1}$ codewords in which there are no consecutive runlengths.
2 Screen the above list of codewords and remove the codewords that do not have GC-weight u. Theorem 6.2 gives the number of such codewords with fixed GC-weight u.
3 Count all the codewords at the distance $d-1$ in a sphere for each codeword in the generated list.
4 Delete the codeword with a maximum number of codewords in radius $d-1$. Reduce the number of codewords at distance $d-1$ by 1 for all codewords that were within the distance $d-1$ of this deleted codeword.
5 Repeat the process in the Step **4** till the maximum number of codewords within distance $d-1$ is at most 1 for each element of the list.
6 Generate DNA codewords by mapping each quaternary code in the list to DNA alphabets $\{0,1,2,3\} \rightarrow \{A,T,G,C\}$ respectively.

remove the codewords with the maximum number of codewords within a radius $d-1$ iteratively. Table 6.1 shows results obtained for $4 \leq n \leq 13$ and $1 \leq d \leq 10$. For instance, 289 codewords are obtained for $n = 8$ and $d = 3$.

The compressed data is indexed from $\{1,2,\ldots,N\}$ and map coded sequence of length n to index n of the compressed data. The coded sequences are converted to DNA sequence by mapping $A \rightarrow 0, T \rightarrow 1, G \rightarrow 2, C \rightarrow 3$. The codebook satisfies the no-repetition, GC-weight u, and minimum distance d constraints. A distance of d implies that $\lfloor (d-1)/2 \rfloor$ substitution errors can be corrected.

6.1.2 Bounds on DNA Codes

In this section, the number of codewords with no-runlength and fixed GC-weight constraints is given. Let us denote the number of such codewords by $B(n,u)$, where n is the length and u is GC-weight of the codeword. Then, with an additional distance constraints is added to this number to give a lower bound on such number of codewords. We will first ignore the distance, and find the number of codewords of GC-weight u with no-runlength constraints.

The following lemma will help us with the derivation of the result on the number of codewords with no-runlength and fixed GC-weight constraints.

Lemma 6.1. *The number of possibilities for d positive integers to sum to u is the same*

as the number of possibilities for d non-negative integers to sum to $u-d$, and is given as $\binom{u-1}{d-1}$.

Proof. The standard results of combinatorics can be used to prove the statement and can be referred from [102]. □

Theorem 6.2. *The number of codewords of GC-weight u with no-runlength constraint is given by*

$$\begin{aligned} B(n,u) &= \sum_{y=0}^{v-1} 2^{2v+1-2y} \binom{v-1}{y} \binom{n-v}{v-y} \\ &\quad + \sum_{y=0}^{v-2} 2^{2v-1-2y} \binom{v-1}{y} \binom{n-v-1}{v-y-2}, \end{aligned} \tag{6.1}$$

for $v>0$, where $v=\min(u,n-u)$. Further, $B(n,u)=2$ for $\min(u,n-u)=0$.

Proof. The proof for $v=0$ is straightforward and is thus omitted. For $v>0$, to derive the results for GC-weight u of the codeword, we need to consider scenario the following cases. For this, divide the proof into three cases based on the ranges of n and u which are

1. Case 1: $n>2u$
2. Case 2: $n<2u$
3. Case 3: $n=2u$

By proving the result of the Theorem for all these cases, the result follows. For notations, let 1_A be equal to one if the condition A is satisfied, and is zero otherwise.

Case 1: $n>2u$

For any location of G/C at u positions, there occur $u+1$ A/T runs between every of the G/C locations (including the start and the end). We denote these run lengths be $x_1,\ldots,x_{u+1}$. For instance ACTAGCATAG has $n=10$, $u=4$, and the runs are $x_1=1$ (A), $x_2=2$ (TA), $x_3=0$ (C follows G), $x_4=3$ (ATA), and $x_5=0$ (ending in G/C).

For any codeword with given $x_1,\ldots,x_{u+1}$, the number of codewordss with no-runlength constraint are $2^{2u+1-1_{x_1=0}-1_{x_{u+1}=0}-2\sum_{i=2}^{u}1_{x_i=0}}$. To understand this, first

note that if all $x_i > 0$, then in each of the $u+1$ runs, we can have any possibility of ATAT$\cdots$ or TATA$\cdots$ giving two possibilities in each of the $u+1$ runs. Further, each location of C or G has two possibilities. As they could be either C or G giving additional 2^u possibilities. Hence, the total number of codewords are 2^{2u+1}. Next, consider that if certain runs are zero, the number of codeword decreases. If the first or last run is zero, the only change that happens is that the additional possibility of 2 choices that can happen in that run are missed thus reducing the possibilities to $2^{2u+1-1_{x_1=0}-1_{x_{u+1}=0}}$. However, if any of $x_i = 0$ for $2 \leq i \leq u$, then the possibilities of A/T run cannot be added and in addition, the extra flexibility of having C or G for the next element is removed due to no-runlength constraint. Thus, we see that the total number of codewords for a given $x_1, \cdots, x_{u+1}$ are $2^{2u+1-1_{x_1=0}-1_{x_{u+1}=0}-2\sum_{i=2}^{u} 1_{x_i=0}}$. In order to obtain the total number of codewords, we can sum this expression over all possible choices of $x_1, \cdots, x_{u+1}$.

Thus the overall number of codewords are given as

$$B(n,u) = \sum_{(x_1,\cdots,x_{u+1})\in G} (2^{2u+1-1_{x_1=0}-1_{x_{u+1}=0}-2\sum_{i=2}^{u} 1_{x_i=0}}), \tag{6.2}$$

where $G = \{(x_1, \cdots, x_{u+1}) : x_i \geq 0, \sum_{i=1}^{u+1} x_i = n-u\}$, which denotes the runs of A/T's and the total number of A/Ts in the codeword is $n-u$. It is also known that $|G| = \binom{n}{u}$. We split G into three parts - the first is when both x_1 and x_{u+1} are non-zero, which we call G_1. The second is when exactly one of x_1 or x_{u+1} is zero, which we call as G_2. The third part of G is when $x_1 = x_{u+1} = 0$, which we denote by G_3. We note that G_1, G_2, and G_3 are disjoint and their union is G. Thus,

$$\begin{aligned} B(n,u) \;=\; & \sum_{(x_1,\cdots,x_{u+1})\in G_1} (2^{2u+1-1_{x_1=0}-1_{x_{u+1}=0}-2\sum_{i=2}^{u} 1_{x_i=0}}) \\ & + \sum_{(x_1,\cdots,x_{u+1})\in G_2} (2^{2u+1-1_{x_1=0}-1_{x_{u+1}=0}-2\sum_{i=2}^{u} 1_{x_i=0}}) \\ & + \sum_{(x_1,\cdots,x_{u+1})\in G_3} (2^{2u+1-1_{x_1=0}-1_{x_{u+1}=0}-2\sum_{i=2}^{u} 1_{x_i=0}}) \end{aligned}$$

We will now evaluate each of these three terms one by one. We label the three

terms from left to right as R_1, R_2, and R_3, respectively. The first term is given as

$$\begin{aligned} R_1 &= \sum_{(x_1,\cdots,x_{u+1})\in G_1} (2^{2u+1-2\sum_{i=2}^{u} 1_{x_i=0}}) \\ &= \sum_{y=0}^{u-1} \sum_{(x_1,\cdots,x_{u+1})\in G_1} (2^{2u+1-2\sum_{i=2}^{u} 1_{x_i=0}}) \times \\ &\qquad 1_{\text{exactly } y \text{ out of } (x_2,\cdots,x_u) \text{ are zero}} \end{aligned} \tag{6.3}$$

In the last step, we note that some of $(x_2, \cdots, x_u)$ could be zero. We let y of them be zero, where all possibilities of y varying from 0 till $u-1$ are accounted. Using this way of summation, we will obtain $\sum_{i=2}^{u} 1_{x_i=0} = y$ and would simplify the expression, as seen below.

$$\begin{aligned} R_1 &= \sum_{y=0}^{u-1} \sum_{(x_1,\cdots,x_{u+1})\in G_1} (2^{2u+1-2y}) \times \\ &\qquad 1_{\text{exactly } y \text{ out of } (x_2,\cdots,x_u) \text{ are zero}} \end{aligned} \tag{6.4}$$

$$\begin{aligned} &= \sum_{y=0}^{u-1} 2^{2u+1-2y} \sum_{(x_1,\cdots,x_{u+1})\in G_1} 1_{\text{exactly } y \text{ out of } (x_2,\cdots,x_u) \text{ are zero}} \\ &= \sum_{y=0}^{u-1} 2^{2u+1-2y} \binom{u-1}{y} \times \\ &\qquad (\text{Number of possibilities for } u+1-y \text{ positive} \\ &\qquad \text{integers to sum to } n-u) \end{aligned} \tag{6.5}$$

The last step follows since the sum can be split as all the sequences that have exactly y out of $u-1$ terms zero, and the $u+1-y$ non-zero integers sum to $n-u$. Then, using Lemma 6.1, we can simplify the expression as follows.

$$R_1 = \sum_{y=0}^{u-1} 2^{2u+1-2y} \binom{u-1}{y} \binom{n-u-1}{u-y} \tag{6.6}$$

We now consider the second part, which is sum over G_2, and is given as follows.

The steps are similar and thus the detailed explanations are skipped.

$$R_2 = \sum_{(x_1,\cdots,x_{u+1})\in G_2} (2^{2u-2\sum_{i=2}^{u} 1_{x_i=0}}) \tag{6.7}$$

$$= \sum_{y=0}^{u-1} \sum_{(x_1,\cdots,x_{u+1})\in G_2} (2^{2u-2\sum_{i=2}^{u} 1_{x_i=0}}) \times$$

$$1_{\text{exactly } y \text{ out of } (x_2,\cdots,x_u) \text{ are zero}} \tag{6.8}$$

$$= \sum_{y=0}^{u-1} \sum_{(x_1,\cdots,x_{u+1})\in G_2} (2^{2u-2y}) \times$$

$$1_{\text{exactly } y \text{ out of } (x_2,\cdots,x_u) \text{ are zero}} \tag{6.9}$$

$$= \sum_{y=0}^{u-1} 2^{2u-2y} \times$$

$$\sum_{(x_1,\cdots,x_{u+1})\in G_2} 1_{\text{exactly } y \text{ out of } (x_2,\cdots,x_u) \text{ are zero}} \tag{6.10}$$

$$= \sum_{y=0}^{u-1} 2^{2u-2y} 2 \binom{u-1}{y} (\text{Number of possibilities for}$$

$$u-y \text{ positive integers to sum to } n-u) \tag{6.11}$$

$$= \sum_{y=0}^{u-1} 2^{2u+1-2y} \binom{u-1}{y} \binom{n-u-1}{u-y-1} \tag{6.12}$$

Now consider the third part, which is sum over G_3, and is given as follows.

$$R_3 = \sum_{(x_1,\cdots,x_{u+1})\in G_3} (2^{2u-1-2\sum_{i=2}^{u} 1_{x_i=0}}) \tag{6.13}$$

$$= \sum_{y=0}^{u-1} \sum_{(x_1,\cdots,x_{u+1})\in G_3} (2^{2u-1-2\sum_{i=2}^{u} 1_{x_i=0}}) \times$$

$$1_{\text{exactly } y \text{ out of } (x_2,\cdots,x_u) \text{ are zero}} \tag{6.14}$$

$$= \sum_{y=0}^{u-1} \sum_{(x_1,\cdots,x_{u+1})\in G_3} (2^{2u-1-2y}) \times \tag{6.15}$$

$$1_{\text{exactly } y \text{ out of } (x_2,\cdots,x_u) \text{ are zero}} \tag{6.16}$$

$$= \sum_{y=0}^{u-1} 2^{2u-1-2y} \times \sum_{(x_1,\cdots,x_{u+1})\in G_3} 1_{\text{exactly } y \text{ out of } (x_2,\cdots,x_u) \text{ are zero}} \tag{6.17}$$

$$= \sum_{y=0}^{u-1} 2^{2u-1-2y} \binom{u-1}{y} (\text{Number of possibilities for } u-y-1 \text{ positive integers to sum to } n-u) \tag{6.18}$$

$$= \sum_{y=0}^{u-2} 2^{2u-1-2y} \binom{u-1}{y} \binom{n-u-1}{u-y-2} \tag{6.19}$$

Adding R_1, R_2, and R_3, we have

$$B(n,u) = \sum_{y=0}^{u-1} 2^{2u+1-2y} \binom{u-1}{y} \binom{n-u}{u-y}$$

$$+ \sum_{y=0}^{u-2} 2^{2u-1-2y} \binom{u-1}{y} \binom{n-u-1}{u-y-2}, \tag{6.20}$$

where we have used the combinatorial identity $\binom{n}{k} + \binom{n}{k-1} = \binom{n+1}{k}$ for simplifications.

Case 2: $2u > n$

In this case, we change the proof technique by fixing A/C and considering the runs of C/G. This way, all the expressions work as in Case 1, by replacing u by $n-u$. Thus, we can replace u by $\min\{n,u\}$ in the Equation (6.20) for all $u \neq n/2$ to combine the expressions of Case 1 and Case 2.

Case 3: $2u = n$

Case 3 differs from Case 1 in the first calculation of G_1, $y = 0$ does not occur in this case. Since the $\binom{l}{l-1} = 0$, does not effect the result. Next, using the combinatorial identity even alleviates the issue of having the term of the form $\binom{l}{l-1}$ in the overall expression. Hence the statement of the Theorem holds. □

Theorem 6.2 gives an exact expression for the number of codewords without any distance constraints. So in the next result, the minimum distance d of the

codewords is incorporated along with the two constraints. Recall that $A_4^{GC}(n,d,u)$ the maximum number of codewords that satisfies with the length n, a minimum distance d and GC-weight u. Here, the function $A_4^{GC}(n,d,u)$ also satisfies no-runlength constraint.

The next theorem gives a lower bound on $A_4^{GC}(n,d,u)$ in the following theorem.

Theorem 6.3. *The maximum number of codewords of length n with GC-weight u with a minimum distance d satisfying no-runlength constraint is lower bounded by*

$$A_4{}^{GC}(n,d,u) \geq \frac{B(n,u)}{\sum_{r=0}^{d-1}\sum_{i=0}^{min\{\lfloor r/2\rfloor,u,n-u\}}\binom{u}{i}\binom{n-u}{i}\binom{n-2i}{r-2i}2^{2i}}. \tag{6.21}$$

Proof. To obtain the lower bound on the number of codewords, we give an upper bound on the number of the codewords that have distance at most $d-1$ from any fixed codeword x by ignoring the no-runlength constraint, and this is given as

$$\sum_{r=0}^{d-1}\sum_{i=0}^{min\lfloor r/2\rfloor,u,n-u}\binom{u}{i}\binom{n-u}{i}\binom{n-2i}{r-2i}2^{2i}$$

[73]. Hence, the result of the theorem holds. □

We compare the number of codewords in the lower bound of Theorem 6.3 with the codes obtained by the proposed altruistic code in Table 6.1. A remarkably higher number of codewords satisfying both these constraints are obtained as compared to the derived lower bound on the DNA constraints. Theorem 6.2 gives an exact expression for $d=1$, thus it is not compared in Table 6.1.

To summarize this section, constraint based codes for archival data storage in DNA are proposed, which uses an altruistic approach for generating DNA codes with both the constraints.

In the next section, another method for constructing codes satisfying the constraints for DNA data storage is discussed. The seven families of non linear ternary codes with the parameters $(9,256,3)_3$, $(11,256,5)_3$, $(15,256,7)_3$, $(18,256,9)_3$,

	$d=2$		$d=3$		$d=4$		$d=5$		$d=6$		$d=7$		$d=8$		$d=9$		$d=10$	
n	c	l	c	l	c	l	c	l	c	l	c	l	c	l	c	l	c	l
4	32	11	11	2	4	0	-	-	-	-	-	-	-	-	-	-	-	-
5	68	21	17	3	7	1	2	0	-	-	-	-	-	-	-	-	-	-
6	216	60	44	7	16	1	6	0	4	0	-	-	-	-	-	-	-	-
7	528	130	110	13	36	2	11	0	4	0	2	0	-	-	-	-	-	-
8	1704	372	289	33	86	6	29	1	9	0	4	0	4	0	-	-	-	-
9	4336	857	662	68	199	11	59	2	15	0	8	0	4	0	0	0	-	-
10	13688	2473	1810	174	525	25	141	4	43	1	7	0	5	0	4	0	4	0
11	35936	5964	4320	382	1235	49	284	8	82	1	29	0	9	0	4	0	4	0
12	112712	17289	12068	1007	3326	119	662	18	190	3	58	1	22	1	8	0	4	0
13	302064	43062	41867	2318	7578	251	1432	34	1201	6	123	1	39	1	13	1	6	1

Table 6.1: The derived lower bounds are compared with the codes obtained using altruistic coding. c indicates codewords generated using altruistic method. l denotes the lower bounds on the codes for n and d obtained from Theorem 6.3. © [2018] IEEE. Reprinted, with permission, from Dixita Limbachiya, Manish K. Gupta, and Vaneet Aggarwal, *Family of Constrained Codes for Archival DNA Data Storage, accepted in IEEE Communication Letters, July 2018, Early Access, doi:10.1109/LCOMM.2018.2861867.*

$(21, 256, 11)_3$, $(24, 256, 13)_3$ and $(26, 256, 15)_3$ respectively [49] are constructed. Among these, $(11, 256, 5)_3$ is a subcode of ternary Golay code [46].

6.2 DNA Golay Subcode Method

Let $\mathscr{C}_{Z_3}(n, M, d_H)$ be a ternary code with a length n, the number of codewords M and the minimum Hamming distance d_H. In this section, we use a non-linear ternary error correction code for encoding the data, instead of Huffman codes that was used in [47]. We discuss encoding of data in DNA using the ternary Golay subcode $(11, 256, 5)_3$ with the length $n = 11$ and the minimum Hamming distance $d_H = 5$ that can correct 2 bit-flips (substitution) errors.

The input file is encoded byte-wise the ternary Golay subcode by the method presented in Figure 6.2. In particular, we give a method to encode the ternary codewords into DNA for archival storage. First, any arbitrary digital file is converted into the list of ASCII values. Therefore, to encode any such file into DNA string, we need a set of 256 codewords such that each can be mapped to one of the ASCII values.

The list of designed codewords of the code $(11, 256, 5)_3$ for data storage are listed in Table 6.7. It consists of 243 codewords from 729 Golay codewords such that the minimum Hamming distance between any two codewords is 5. These 243

Table 6.2: Conversion of ternary codewords to DNA codewords developed by N. Goldman *et al.* [47] which avoids runlengths.

$$\psi = \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline A & C & G & T \\ C & G & T & A \\ G & T & A & C \\ T & A & C & G \end{array}$$

codewords were assigned ASCII values with the higher probability of occurrences of the alphabet. The rest of 13 ASCII values were assigned ternary Golay codewords randomly *i.e.*, these 13 codewords will have the minimum Hamming distance $d_H = 5$ with other 243 codewords. It assures that by a maximum likelihood decoding, two substitution errors can be corrected. It is challenging to construct a code with size at least 256 codewords such that the length of each codeword is 11 and the minimum Hamming distance $d_H > 5$. Because it is only possible to have 243 codewords in the set of 3^{11} codewords such that the minimum Hamming distance is $d_H = 6$. Hence, we use the code $(11, 256, 5)_3$ to encode the data. Next, the ternary codewords is converted to DNA such that runlengths of DNA bases can be prevented. It is desirable to have DNA with length 100-250 bases because the long DNA bases is not feasible for DNA sequencing. Thus, the resulting DNA codeword was divided into chunks each of length 99 bases as shown in Figure 6.1. We have modified the chunk architecture given in [47] for the DNA data storage by adding flexibility to the length of the chunk. In [47], only the fix length of chunk ($l = 99$) is possible while in this method, we can vary the chunk length based on the size of input file.

In Figure 6.1, $(i = 99)$ denotes the number of bases used for encoding the original file. λ is the number of bases required to store file index (number of file index trits = 2 which allow the maximum of 9 files to be differentiated). The μ indicates the DNA bases required for chunk index (no of segment index trits $\mu = \lceil \log_3 (\text{ total no of segments}) \rceil$) and an odd parity-check is appended at the end of each segment. This parity is obtained by summing odd bits of a file identifier and a chunk index.

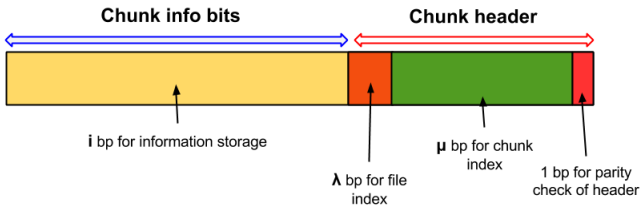

Figure 6.1: DNA Storage Chunk Architecture: Given chunk architecture has two parts. It has information (i) bits (Yellow color) and the chunk header. A chunk information bits contains original data to be encoded and a chunk header. The chunk header includes a file index for file identification and index of the chunk to identify a particular chunk. An odd parity check bit is appended at the end. © [2018] IEEE. Reprinted, with permission, from Dixita Limbachiya, Vijay Dhameliya, Madhav Khakhar, and Manish K. Gupta, *On Optimal Family of Codes for Archival DNA storage*, In Proceedings of IEEE Seventh International Workshop on Signal Design and its Applications in Communications (IWSDA), pp. 123-127. 2015.

6.2.1 Algorithm for Encoding and Decoding Data Files

The sequential procedure for encoding and decoding the data from a file into the DNA and vice versa is described in the respective Algorithm 2 and 3 respectively. For encoding and decoding, it assumes that base A is a previous base at the start position.

Example 6.4. *A simple example is demonstrated to understand the encoding and decoding procedure. The message data is "DA". The corresponding ASCII values for 'D' and 'A' are 68 and 65 respectively. The codes for the letter 'D' is* 02221221120 *and for 'A' is* 10111000101 *referred from Table 6.7. DNA codeword corresponding to "DA" is CATGATGCTGAGTCTCGTAGTC. This DNA is divided into two chunks C_1 is CATGATGCTGA and C_2 is GTCTCGTAGTC (here each chunk is of length 11). Let the chunk index (i_1 and i_2) for each DNA chunk be* 0 *and* 1 *respectively. Let the file identifier ID for the text be* 00*. A parity bit P is added which is calculated by summation of odd positions in ID and i. Now each DNA chunk is appended by with chunk identifier bases (concatenating ID.i.P). A chunk index for each DNA chunk C_1 and C_2 is CGTA and CGAG respectively. So the final DNA chunks are CATGATGCTGACGTA and GTCTCGTAGTCCGAG.*

Some results on error correction for the proposed codes for DNA data storage

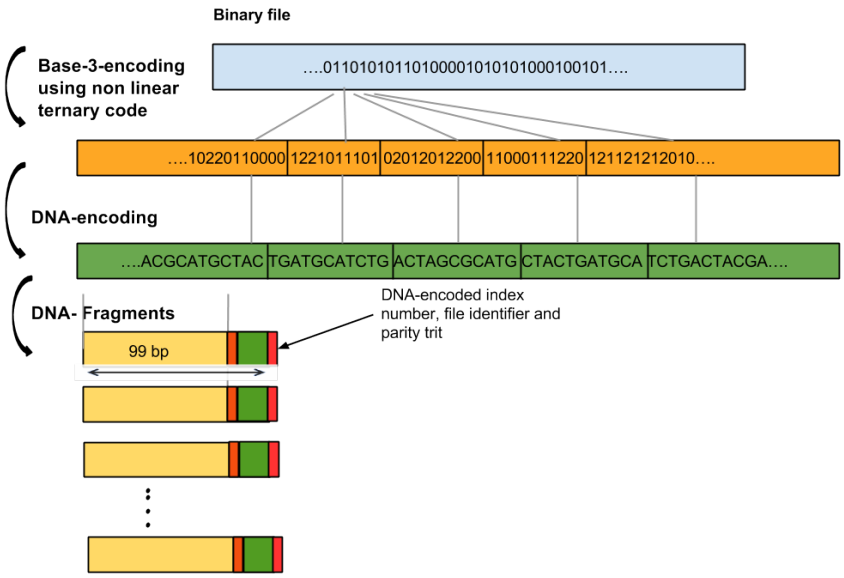

Figure 6.2: Schematic representation of the proposed DNA Golay subcode is presented. The steps of converting input files into DNA codewords by using the ternary DNA Golay subcode Table 6.7. First step in blue color is to convert the binary code were mapped to a base 3 non-linear ternary code (indicated in orange color). Next, these ternary codewords were converted to DNA codewords (green color) using the conversion Table 6.2. This helps in avoiding runlengths. DNA was divided into DNA chunks, each of length 99. © [2018] IEEE. Reprinted, with permission, from Dixita Limbachiya, Vijay Dhameliya, Madhav Khakhar, and Manish K. Gupta, *On Optimal Family of Codes for Archival DNA storage*, In Proceedings of IEEE Seventh International Workshop on Signal Design and its Applications in Communications (IWSDA), pp. 123-127. 2015.

Algorithm 2 Algorithm for Encoding

Input: Any arbitrary computer file
Output: DNA sequences in which data is encoded

1.1 Any computer file is read , and each bit is converted to base 3 using ternary Golay subcode (mentioned in Table 6.7). It gives a ternary codeword of length 11 for each bit in the string which is denoted S_1.
1.2 The file extension of a file is converted to the ternary code (using Table 6.7), and it is denoted as S_3.
1.3 To mark the input file and its extension, the separators S_2 and S_4 *comma* (,) and *colon* (:) respectively is used. String N is encoded into code and stored in S_5.
1.4 In order to have chunks of the same length, the string S_1 is padded with *zeros*. The number of zeros is denoted as S_6.
1.5 Strings $S_7 = S_1.S_2.S_3.S_4.S_6.S_5$ is concatenated in string S_7. Size of the string S_7 is denoted as n.
1.6 The string S_7 is encoded to a DNA string using the ψ map (see Table 6.2).
1.7 The chunk index is added to each chunk. A file identifier and parity check bit are appended at the ends of each chunk.
1.8 The data encoded DNA sequence can be synthesized and stored.

Algorithm 3 Algorithm for Decoding

Input: DNA sequences in which data is encoded.
Output: Original computer file

1.1 The DNA sequence is decoded by extracting the input file from the chunk.
1.2 Data encoded DNA sequence is converted to ternary codewords using the ψ map in Table 6.2.
1.3 Each string is decoded to length 11 (the Table 6.7).
1.4 Data is retrieved by corresponding the ASCII values corresponding to each ternary codeword. The original file is segregated from the separators.

are obtained in the next section.

6.2.2 Analysis on Error Correction

Remark 6.5. *Let ψ be the map used for the conversion (using Table 6.2) of $\mathscr{C}_{Z_3}$ ternary code to $\mathscr{C}_{DNA}$ DNA code i.e, $\psi(\mathscr{C}_{Z_3}) = \mathscr{C}_{DNA}$.*

The alternative possible mapping is given in the Tables 6.3 and 6.4.

$$\psi_1 = \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline A & T & C & G \\ C & A & G & T \\ G & C & T & A \\ T & G & A & C \end{array}$$

Table 6.3: Alternative mapping ψ_1 to covert the ternary Golay subcode to the DNA nucleotides avoiding runlengths.

$$\psi_2 = \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline A & G & T & C \\ C & T & A & G \\ G & A & C & T \\ T & C & G & A \end{array}$$

Table 6.4: Alternative mapping ψ_2 to covert the ternary Golay subcode to the DNA nucleotides avoiding runlengths.

In order to investigate the error correction possible in DNA codewords, the following results holds.

Lemma 6.6. *Let $\psi(\boldsymbol{x}) \in \mathscr{C}_{DNA}$ send through a noisy channel and $\psi(\boldsymbol{y})$ is received. If $d_H(\psi(\boldsymbol{x}), \psi(\boldsymbol{y})) = t$ and corresponding ternary code obtained for $\psi(\boldsymbol{x})$ and $\psi(\boldsymbol{y})$ are $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively then $t < d_H(\boldsymbol{x}, \boldsymbol{y}) \leq 2t$ where $\boldsymbol{x}, \boldsymbol{y} \in \mathscr{C}_{Z_3}$.*

Proof. As per the lemma, assume that $\psi(\mathbf{x})$ is sent and $\psi(\mathbf{y})$ is received. Their corresponding ternary codes are $\mathbf{x}$ and $\mathbf{y}$ respectively (using Table 6.2). By using Lemma 6.5, one can observe the following. Let $\mathbf{x}[i]$ be the i^{th} element of the strings$\mathbf{x}$. Occurrence of substitution errors can be characterized as follows

1. Bit Flip error: For $1 \leq i < j < 11$, if $\psi(\mathbf{x})[i] \neq \psi(\mathbf{y})[i]$ and $\psi(\mathbf{x})[j] \neq \psi(\mathbf{y})[j]$ such that $j - i > 1$ then $\mathbf{x}[i] \neq \mathbf{y}[i]$, $\mathbf{x}[i+1] \neq \mathbf{y}[i+1]$ and $\mathbf{x}[j] \neq \mathbf{y}[j]$, $\mathbf{x}[j+1] \neq \mathbf{y}[j+1]$. Hence for the bit flip error, $d_H\ (\ \psi(\mathbf{x}), \psi(\mathbf{y})\ ) = 2t$.

2. Burst error in consecutive b positions: For $1 \leq i < 11$, if $\psi(\mathbf{x})[i] \neq \psi(\mathbf{y})[i]$, $\psi(\mathbf{x})[i+1] \neq \psi(\mathbf{y})[i+1], \ldots, \psi(\mathbf{x})[i+b] \neq \psi(\mathbf{y})[i+b]$ then $\mathbf{x}[i] \neq \mathbf{y}[i]$, $\mathbf{x}[i+1] \neq \mathbf{y}[i+1], \ldots, \mathbf{x}[i+b] \neq \mathbf{y}[i+b]\ \forall\ b < d_H$. Hence for the burst error, $d_H\ (\ \psi(\mathbf{x})\ ,\ \psi(\mathbf{y})\ ) = t + 1$.

3. For random bit flips and burst error in consecutive positions, from above cases one can observe that $d_H\ (\ \psi(\mathbf{x})\ ,\ \psi(\mathbf{y})\ ) \leq t$.

If the number of bit flips or burst error at end positions in DNA code is t, then the Hamming distance between respective ternary codes will be t because the number of errors in ternary codes cannot be less than the number of error in DNA codes. Hence in general, $t < d_H(\ \mathbf{x}\ ,\ \mathbf{y}\ ) \leq 2t$. □

Example 6.7. *Let $\psi(\boldsymbol{x}) = GTCTCGTAGTC$ and $\psi(\boldsymbol{y}) = GAGTCGTAGTC$ then $\boldsymbol{x} =$ 10111000101 and $\boldsymbol{y} =$ 11101000101. The minimum Hamming distance $d_H(\psi(\boldsymbol{x}),\psi(\boldsymbol{y})) =$ 2 and $d_H(\boldsymbol{x}, \boldsymbol{y}) = 4$.*

Corollary 6.8 is an observation for results of Lemma 6.6.

Corollary 6.8. *For $\psi(\boldsymbol{x})$, $\psi(\boldsymbol{y}) \in \mathscr{C}_{DNA}$ and $\boldsymbol{x}$, $\boldsymbol{y} \in \mathscr{C}_{Z_3}$, if the minimum Hamming distance $d_H(\psi(\boldsymbol{x}), \psi(\boldsymbol{y})) = t$ then $d_H(\boldsymbol{x}, \boldsymbol{y}) = 2t$. But this is not applicable in reverse direction, that is if we know $d_H(\boldsymbol{x}, \boldsymbol{y})$, we cannot say anything about the distance between $\psi(\boldsymbol{x})$ and $\psi(\boldsymbol{y})$.*

Example 6.9. *Let $\psi(\boldsymbol{x}) = GTCTCGTAGTC$ and $\psi(\boldsymbol{y}) = GACTCGTAGTC$ then $\boldsymbol{x} =$ 10111000101 and $\boldsymbol{y} =$ 11011000101. The minimum Hamming distance $d_H(\psi(\boldsymbol{x}),$ $\psi(\boldsymbol{y})) = 1$ and $d_H(\boldsymbol{x}, \boldsymbol{y}) = 2$.*

Lemma 6.10. *Using the double layer error correcting scheme, one can correct any 2 bit flips in DNA.*

Proof. If there is 1 bit flip in a DNA codeword, it implies 2 bit flips in ternary codeword using Lemma 6.5. This can be easily corrected using the ternary code

with the distance $d_H = 5$. In case of 2 bit flips, one can prove it by contradiction and Lemmas 6.5 and 6.6.

Let $\mathbf{x}$ and $\mathbf{y}$ be the ternary codeword corresponding the DNA codeword $\psi(\mathbf{x})$ and $\psi(\mathbf{y})$ respectively. Now, $2t < d_H(\mathbf{x},\mathbf{y}) \leq 4t$ by using Lemma 6.6. Now, let L be the set of codewords from $\mathscr{C}_{Z_3}$ that are at distance $d_H = 4$ from $\mathbf{y}$.

Select $\mathbf{w} \in \mathscr{C}_{Z_3}, \mathbf{w} \neq \mathbf{x}$ such that $d_H(\mathbf{y},\mathbf{w}) \leq 4t$. Let L_{DNA} be the set of corresponding codewords $\mathbf{w}_{DNA} \in \mathscr{C}_{DNA}$ obtained by converting the ternary code to the DNA code. Trivially, $\mathbf{x} \in L$. Let L_{DNA} be the set of corresponding codewords $\psi(\mathbf{w}_{DNA}) \in \mathscr{C}_{DNA}$ obtained.

Now, $\forall\ \psi(\mathbf{w}) \neq \psi(\mathbf{x})$, we claim that $d_H(\psi(\mathbf{y}),\psi(\mathbf{w})) > 2t \implies d_H(\mathbf{y},\mathbf{w}) \leq 4$ $\Rightarrow d_H(\psi(\mathbf{y}),\psi(\mathbf{w})) > 2$. Let us prove it by contradiction. Suppose $d_H(\psi(\mathbf{y}),\psi(\mathbf{w}))$ ≤ 2. This can be divided into two cases. In the first case, $d_H(\psi(\mathbf{y}),\psi(\mathbf{w})) \leq t$. But as per Lemma 6.6, this cannot occur and hence such $\psi(\mathbf{w})$ can't exist in the set L while solving for $2t$ errors. In the second case, $t+1 \leq d_H(\psi(\mathbf{y}),\psi(\mathbf{w})) \leq 2t$. This will result in required codeword $\psi(\mathbf{x})$.

□

Example 6.11. *Let* $\psi(\boldsymbol{x}) = C\boldsymbol{A}TGA\boldsymbol{T}GAGCG$ *and* $\psi(\boldsymbol{y}) = C\boldsymbol{G}TGA\boldsymbol{C}GAGCG$ *then* $\boldsymbol{x} = 02221221120$ *and* $\boldsymbol{y} = 0\boldsymbol{00}21\boldsymbol{00}1120$. *The minimum Hamming distance* $d_H(\psi(\boldsymbol{x}),$ $\psi(\boldsymbol{y})) = 2$ *and* $d_H(\boldsymbol{x},\boldsymbol{y}) = 4$. *Let* $L \subset \mathscr{C}_{Z_3}$ *such that* $d_H(\boldsymbol{y},\boldsymbol{w}) \leq 4\ \forall\ \boldsymbol{w} \in L$. *See Table 6.5.*

6.2.3 Results and Simulation

We select five computer files used by N. Goldman *et al.* [47] as a proof of concept for the refurbished algorithm for data encoding in DNA codewords. Simulation and analysis for the input data encoded in DNA sequences were performed by the software DNA Cloud [121]. First, selected input files were converted into DNA sequences. These files constituted the total of 757051 bytes, and each byte was encoded into the ternary Golay codeword of length 11 using the software. In a nutshell, the five files were stored in 84126 DNA chunks. The size of each chunk varied from 109 to 112 nucleotides (nt) (depending on the number of chunks required for the input file). This results in encoding of 757,051 bytes of input data

No	w	$d_H(\mathbf{y}, w)$	$\psi(w)$	$d_H(\psi(w), \psi(\mathbf{y}))$
w_1	20021020110	4	TACAGTGTCTA	10
w_2	02221221120	4	CATGATGAGCG	2
w_3	00020121100	4	CGTGTCAGACG	4
w_4	00021112020	4	CGTGAGATATA	6
w_5	01011011100	4	CTAGACTCTAC	7
w_6	20211001020	4	TATCTACTATA	10
w_7	12001002120	4	GCGTCGTGATA	11
w_8	00001021220	4	CGTAGTGATGT	6
w_9	00022001121	4	CGTGCGTCTGA	7
w_{10}	01121000220	4	CTCAGTACATA	10
w_{11}	00101101110	4	CGACTCGAGAC	5
w_{12}	10021201200	4	GTATCACTGTA	10

Table 6.5: Observe that the highlighted codeword w_2 is at only at distance 4 ie. $d_H(\psi(w), \psi(\mathbf{y})) = 2$. Hence, $d_H(\mathbf{y}, w) = 4$ which implies w_2 is the sent codeword that is highlighted in red color.

Original File name	File size	Bytes	Chunk Size	No. of Chunks	No. of nucleotides	Chunk Size	No. of chunks	No. of nucleotides
			N.Goldman Result			Golay Codes Result		
EBI.jp2	179.9 KB	184264	117	37423	4378491	112	20476	2293312
MLK_excerpt_VBR_45 − 85.mp3	164.6 KB	168539	117	34164	3997188	111	18728	2078808
View_huff3.cd.new	15.3 KB	15646	117	3163	370071	109	1740	189660
watsoncrick.pdf	274.3 KB	280864	117	56911	6658587	112	31209	3495408
wssnt10.txt	105.2 KB	107738	117	21650	2533050	111	11973	1329003
Total	739.3 KB	757051		153335	17937387		84126	9386191

Table 6.6: Executive Summary of data encoded using N. Goldman *et al.* approach and proposed DNA Golay subcode approach

in 9,386,191 nt (see Table 6.6). This evident the improved results compared to N. Goldman's encoding scheme, that encoded the same files of 757,051 bytes in 153,335 DNA chunks, each of 117 nt. Note that we have not performed synthesis and sequencing protocols for the encoded data in DNA. However, we observed theoretical improvements in DNA data net information density and associated cost.

DNA Net Information Density

We analyzed DNA net information density for the proposed ternary Golay subcodes for DNA data storage.

Definition 6.1. *For a given DNA based information storage system, DNA information density is the total amount of data in bits that can be stored in the unit gram of DNA.*

At the theoretical maximum, one gram of a single-stranded genetic code can

store 455 EB (exabytes) of information [24]. Work proposed by N. Goldman *et al.* achieved practical information density 2.2 PB (petabytes) per gram of DNA while using the DNA Golay subcodes, theoretically we have achieved net information density for DNA data storage system for the proposed chunk architecture as $1.15 \times 10^{20} = 115$ EB (Exabytes) per gram DNA.

Proposition 6.12. *DNA Net Information Density using our chunk architecture for one gram of DNA is calculated by solving the following non-linear equation*

$$\left[(182 \times 10^{19} \times l) \div \left((l+3) + log_3\left(\frac{N\,(x+22)}{l}\right) \times N\right)\right] - 22 = x, \tag{6.22}$$

where, x= number of bytes per one gram of DNA, l = Length of chunk without chunk index, N = Length of the error correcting code.

Proof. Consider total information that can be encoded in one gram of DNA is x bytes. Let I be number of nucleotides required to store file such that

$$I = N \times x + N \times 2 + N \times log_{10}(x) \tag{6.23}$$

where $x \times N$ are nucleotides for x bytes, $2 \times N$ are nucleotides for 2 separators and $N \times log_{10}(x)$ are nucleotides for storing file size on DNA. Since maximum storage capacity of DNA is 455 exabytes, we can consider $log_{10}(x) = 20$. Therefore I can be generalized as

$$I = N\,(x+22) \tag{6.24}$$

Information encoded in DNA is divided into chunks of particular length. Let chunk length without chunk index be l, length of chunk with chunk index be L and number of chunks be C.
$L = (l+3) + (log_3\,(C))$ where Chunk number $C = \lceil \frac{I}{l} \rceil$. To calculate DNA net information density, we need to estimate bytes which can be stored in one gram of DNA which has 182×10^{19} nucleotides. Therefore, DNA net data density is

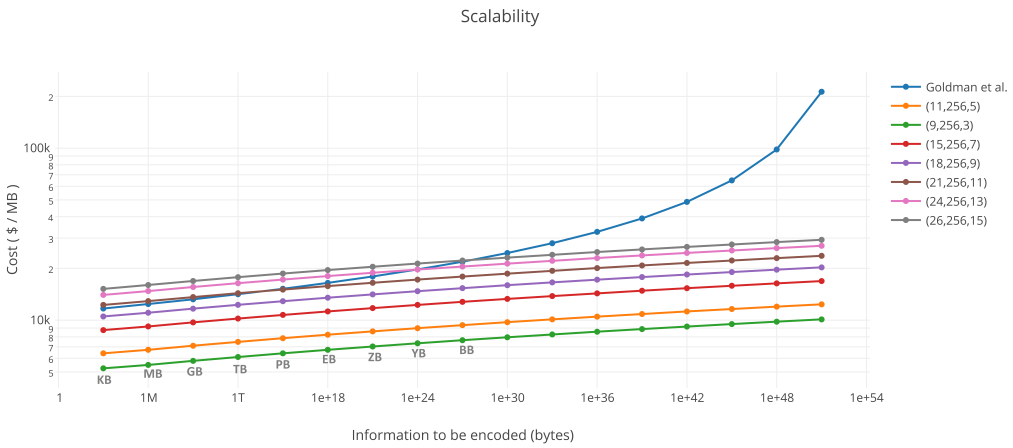

Figure 6.3: Comparison between cost of DNA synthesis and sequencing using N. Goldman approach and families of DNA Golay subcodes used in the DNA information storage. The graph shows that cost using N. Goldman's approach is significantly higher than the DNA Golay subcodes approach.

obtained by following

$$x = \left[\left(182 \times 10^{19} \times l \right) \div \left((l+3) + log_3 \left(\frac{N\,(x+22)}{l} \right) \times N \right) \right] - 22 \qquad (6.25)$$

□

Theoretical DNA net information density for the proposed method obtained by solving Equation 6.25 is 1.15×10^{20} (115 Exabytes) bytes per gram of DNA for the chunk length $l = 99$ and $N = 11$.

DNA Storage Cost Simulation

We plot the DNA data storage cost for various file size (MB) by assuming cost \$0.05 per DNA base (see Figure 6.3). Using the DNA Golay code, there was drop in amount of the DNA required which points to decrease in the cost of storage. We have also plotted the cost of non-linear families of ternary codes. The proposed codes are cost-effective as there is a trivial increase in cost with the increment in the amount of the data encoded while there is significant increment in the cost of the method proposed by N. Goldman *et al.* [47] .

Trade off

The coding potential of the proposed encoding scheme is 0.73 bits per base by encoding 8 bits into 11 DNA bases per byte (*i.e.*, $8/11 = 0.73$ bits per base). This can be improved to 0.89 by employing 9 base per byte (*i.e.*, $8/9 = 0.89$ bits per base) with $(9, 256, 3)_3$ code. However, this will reduce error detection to two bits and error correction to one bit for the code.

Figure 6.4 shows the trade off for each family of codewords developed along with error correction capacity t. Note that as the length n of the code increases, there is a linear increase in the error correction capacity t. But code rate decreases as there is an increment in both the length n and error correction t. Hence, for the robust data storage, $(11, 256, 5)_3$ code is better than the code $(9, 256, 3)_3$ because with the code $(9, 256, 3)_3$, the code rate and coding potential decreases. Error correction capacity t for each code increases linearly with the increase in the length of the code n. However, with the increase in these two parameters, there is declination of the code rate.

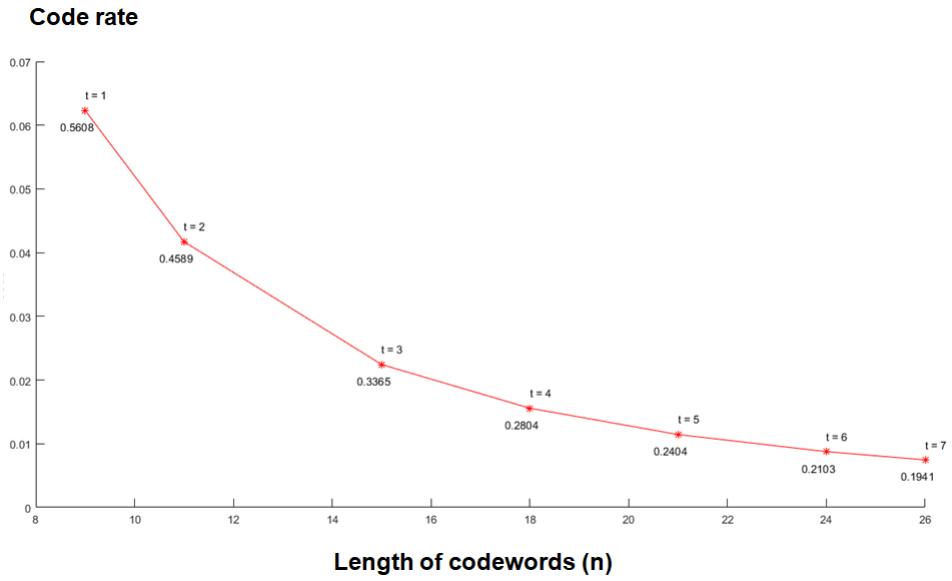

Figure 6.4: A curve plots the trade off between code rate of a code and length of codeword. One can observe that the code with length $n = 12$ and error correction $t = 2$ is a reliable code that has code rate 0.5

In summary, codes for DNA data storage systems are proposed by using constrained coding and Golay subcode method. These DNA codes satisfy the two in-

evitable constraints which are fixed GC-weight and no run-lengths for DNA storage. The presented DNA codes with the DNA constraints is necessary to avoid errors that may occur during DNA synthesis and sequencing.

ASCII Values	Golay codes	Weight	ASCII values	Golay codes	Weight	ASCII values	Golay codes	Weight	ASCII values	Golay codes	Weight
86	00002111202	6	170	00001222101	6	127	00020220222	6	253	00022001121	6
52	00021112020	6	138	00010110111	6	41	00012221010	6	86	00011002212	6
42	00201010122	6	100	00200121021	6	44	00202202220	6	250	00221200011	6
132	00220011210	6	161	00222122112	9	98	00211120200	6	8	00210201102	6
34	00212012001	6	10	00102020211	6	149	00101101110	6	87	00100212012	6
21	00122210100	6	74	00121021002	6	36	00120102201	6	69	00112100022	6
177	00111211221	9	20	00110022120	6	213	02012212122	9	163	02011020021	6
229	02010101220	6	255	02002102011	6	197	02001210210	6	133	02000021112	6
252	02022022200	6	26	02021100102	6	173	02020211001	6	151	02210222211	9
82	02212000110	6	75	02211111012	9	37	02200112100	6	166	02202220002	6
191	02201001201	6	88	02220002022	6	63	02222110221	9	68	02221221120	9
150	02111202000	6	76	02110010202	5	4	02112121101	9	154	02101122222	9
234	02100200121	6	22	02102011020	6	162	02121012111	9	105	02120120010	6
102	02122201212	9	171	01021121211	9	104	01020202110	6	169	01022010012	6
196	01011011100	6	208	01010122002	6	84	01012200201	6	130	01001201022	6
146	01000012221	6	72	01002120120	6	16	01222101000	6	66	01221212202	9
24	01220020101	6	106	01212021222	9	223	01211102121	9	58	01210210020	6
137	01202211111	9	73	01201022010	6	101	01200100212	6	168	01120111122	9
181	01122222021	9	175	01121000220	6	251	01110001011	6	40	01112112210	9
140	01111220112	9	17	01100221200	6	83	01102002102	6	254	01101110001	6
240	20121202122	9	214	20120010021	6	53	20122121220	9	202	20111122011	9
25	20110200210	6	18	20112011112	9	247	20101012200	6	174	20100120102	6
112	20102201001	6	89	20022212211	9	210	20021020110	6	217	20020101012	6
248	20012102100	6	194	20011210002	6	182	20010021201	6	80	20002022022	6
79	20001100221	6	195	20000211120	6	12	20220222000	6	209	20222000202	6
165	20221111101	9	245	20210112222	9	2	20212220121	9	81	20211001020	6
38	20200002111	6	141	20202110010	6	211	20201221212	9	239	22100111211	9
95	22102222110	9	43	22101000012	6	224	22120001100	6	203	22122112002	9
145	22121220201	9	147	22110221022	9	19	22112002221	9	50	22111110120	9
136	22001121000	6	107	22000202202	6	134	22002010101	6	109	22021011222	9
153	22020122121	9	148	22022200020	6	205	22011201111	9	212	22010012010	6
54	22012120212	9	241	22202101122	9	156	22201212021	9	115	22200020220	6
116	22222021011	9	78	22221102210	9	67	22220210112	9	70	22212211200	9
178	22211022102	9	159	22210100001	6	142	21112020000	6	92	21111101202	9
48	21110212101	9	90	21102210222	9	218	21101021121	9	126	21100102020	6
39	21122100111	9	219	21121211010	9	167	21120022212	9	114	21010000122	6
172	21012111021	9	14	21011222220	9	120	21000220011	6	139	21002001210	6
160	21001112112	9	33	21020110200	6	179	21022221102	9	117	21021002001	6
225	21211010211	9	129	21210121110	9	183	21212202012	9	230	21201200100	6
35	21200011002	6	93	21202122201	9	6	21221120022	9	32	21220201221	9
56	21222012120	9	158	10212101211	9	185	10211212110	9	47	10210020012	6
143	10202021100	6	123	10201102002	6	204	10200210201	6	242	10222211022	9
111	10221022221	9	103	10220100120	6	108	10110111000	6	9	10112222202	9
65	10111000101	6	249	10100001222	6	13	10102112121	9	180	10101220020	6
226	10120221111	9	144	10122002010	6	15	10121110212	9	57	10011121122	9
128	10010202021	6	135	10012010220	6	243	10001011011	6	190	10000122210	6
207	10002200112	6	77	10021201200	6	45	10020012102	6	91	10022120001	6
192	12221010000	6	186	12220121202	9	216	12222202101	9	97	12211200222	9
118	12210011121	9	246	12212122020	9	215	12201120111	9	51	12200201010	6
206	12202012212	9	184	12122020122	9	227	12121101021	9	233	12120212220	9
237	12112210011	9	188	12111021210	9	113	12110102112	9	49	12102100200	6
201	12101211102	9	155	12100022001	6	222	12020000211	6	231	12022111110	9
5	12021222012	9	27	12010220100	6	131	12012001002	6	164	12011112201	9
3	12000110022	6	46	12002221221	9	119	12001002120	6	28	11200222122	9
176	11202000021	6	23	11201111220	9	64	11220112011	9	157	11222220210	9
187	11221001112	9	244	11210002200	6	238	11212110102	9	96	11211221001	9
235	11101202211	9	60	11100010110	6	1	11102121012	9	110	11121122100	9
200	11120200002	6	221	11122011201	9	99	11111012022	9	31	11110120221	9
198	11112201120	9	193	11002212000	6	125	11001020202	6	124	11000101101	6
152	11022102222	9	122	11021210121	9	71	11020021020	6	94	11012022111	9
220	11011100010	6	29	11010211212	9	199	00000201211	5	61	00000102122	5
11	00002012110	5	228	00002210021	5	62	00001021220	5	55	00001120012	5
121	00020121100	5	7	00020022011	5	30	00022100210	5	232	00022202002	5
189	00021010201	5	59	00010212200	5	236	00010011022	5	0	00000000000	0

Table 6.7: Codewords from subcode of Ternary Golay code *i.e.* $(11, 6, 5)_3$ assigned to 256 ASCII values is given in the Table. © [2018] IEEE. Reprinted, with permission, from Dixita Limbachiya, Vijay Dhameliya, Madhav Khakhar, and Manish K. Gupta, *On Optimal Family of Codes for Archival DNA storage*, In Proceedings of IEEE Seventh International Workshop on Signal Design and its Applications in Communications (IWSDA), pp. 123-127. 2015.

CHAPTER 7

Conclusion and Future Scope

Biology has at least 50 more interesting years.

James Watson

News summaries 31 Dec 1984. Quoted in James Beasley Simpson, Simpson's Contemporary Quotations (1988), [36]

DNA computing is a fascinating area of research which have expanded its role in different applications. For such computation, DNA codes are designed to perform desirable computation. There are different approaches in the literature to design the DNA codes. In this work, theoretical approaches of algebraic coding are described. DNA codes using the ring $\mathbb{Z}_4 + w\mathbb{Z}_4$, where $w^2 = 2 + 2w$ is proposed. A new distance (called the Gau distance) on the ring R is introduced. We have also proposed a new distance preserving the Gau map ϕ from the elements of the ring R to all the DNA codewords of length 2. Different properties as linearity and closure of the Gau map for DNA codes is presented. Several new families of the DNA codes which satisfies Hamming distance, reverse and reverse complement constraints are obtained. Some of them are optimal with respect to the bounds and are better than the DNA codes obtained in the literature. Some of general results on rings are also described. Further, results of the Gau map is extended to different rings.

Later part of the thesis consist of codes with no-runlength and fixed GC-weight constraints for DNA data storage. We provide constrained coding to generate

DNA codewords with the minimum distance between the codewords satisfying these constraints. An exact number of DNA codewords with both the constraints is enumerated. Further, bounds on such number of DNA codewords are provided.

We have also developed another new approach which satisfies both these constraints using Golay subcodes. A non-linear families of DNA codes is developed for archival DNA information encoding systems. By using proposed ternary Golay subcode, two bit-flips errors can be corrected. We have simulated the method and analyzed code rate and estimated cost for the method.

DNA codes have been constructed on some finite fields, but there are many other finite fields from which DNA codes can be constructed. For the higher values of n and large M, fields of higher order can be used. Significant work has been done on constructing DNA codes from rings by considering all the theoretical approaches. It gives a more substantial number of DNA codewords for larger values of n. Mapping plays a vital role in DNA code construction over rings and field. Different mapping on the same fields or rings can generate different DNA codes.

Though theoretical construction methods help to achieve bounds on the DNA codes for larger values of length n and distance d, it is difficult to modify the theoretically constructed DNA codewords for the specific application of DNA computing. The interesting problem is to classify the DNA codes with respect to algebraic properties of DNA. Endeavoring the portability of DNA codes from one application to another that can help in the building common background for different applications is challenging. For the future study, it would be an exciting task to investigate the algebraic structure of the cyclic codes over the ring R and their correspondence to the DNA codes using the map ϕ. Using algebraic coding, constructing the optimal DNA codes meeting the bounds on the reverse, reverse complement, GC-weight constraints is also an exciting future work.

Using the DNA codes for data storage is widely studied in recent years and has achieved handful success. Still, there are challenges associated with DNA reading and writing technology. As the cost of reading and writing DNA are

high, storing the data in DNA is still expensive. Nevertheless, looking at the recent technological advancements in DNA synthesis and sequencing methods, it is anticipated that DNA storage may become an extremely competitive technology for archival data storage. Designing the DNA storage architecture with capacity achieving codes with error handling ability, including reading, writing and storage errors are a spellbinding research problem. Designing the optimal coding schemes universal to any data type is still a challenge. To develop data extracting method to retrieve the information the randomly from the pools of the millions of DNA sequences in the well is an interesting research problem. As the interaction of computers and programmable DNA is leading to significant discoveries, one needs to develop the methods that are feasible to future DNA computers. Researchers need to understand better the way nature reads the information from the DNA over the extended period. Investigating the approaches to optimize the DNA data storage space such that data is stored in a compact form in a single cell is challenging. Challenges to build the DNA based data storage with low cost and high reading writing speed, technologies of DNA writing and reading should be improved such that large-scale DNA synthesis and sequencing can be done by using massive parallel next-generation DNA synthesis and sequencing [122]. Looking at the technical requirements of the DNA manipulation machines, it will be interesting to develop robust and portable technology for the DNA data storage.

Extending the DNA data storage to other biomolecules and the living model organism is anticipated in the near future.

Publications

1. **Dixita Limbachiya**, Manish K. Gupta, and Vaneet Aggarwal, *Family of Constrained Codes for Archival DNA Data Storage*, IEEE Communications Letters 22.10 (2018): 1972-1975.

2. **Dixita Limbachiya**, Krishna Gopal Benerjee, Bansari Rao and Manish K Gupta, *On DNA Codes using the Ring* $\mathbb{Z}_4 + w\mathbb{Z}_4$, In Proceedings of IEEE International Symposium on Information Theory (ISIT), pp. 2401-2405, 2018.

3. **Dixita Limbachiya**, Vijay Dhameliya, Madhav Khakhar, and Manish K. Gupta, *On Optimal Family of Codes for Archival DNA storage*, In Proceedings of IEEE Seventh International Workshop on Signal Design and its Applications in Communications (IWSDA), pp. 123-127, 2015.

4. **Dixita Limbachiya**, Krishna Gopal Benerjee, Bansari Rao, Manish K Gupta, *Computational Algebraic Perspectives of DNA Codes*, International School on Computer Algebra (COCOA 2016) IIT Gandhinagar, India, 2016 (poster).

5. **Dixita Limbachiya**, *Mathematical techniques for DNA based information storage systems*, Symposium on Mathematical and computational biology, IIT Gandhinagar, 2015 (poster).

6. Amay Agrawal, Birva Patel, **Dixita Limbachiya** and Manish K Gupta, *3DNA Printer : A Tool For Automated DNA Origami*, In Proceedings of Foundations of Nanoscience Self-Assembled Architectures and Devices (FNANO 17), SnowBird, Utah, USA, pp.125-125. [online] Available at arXiv: 1702.04343 [cs.ET].

7. **Dixita Limbachiya**, Dhaval Trivedi and Manish K. Gupta, *DNA Image Pro - A Tool for Generating Pixel Patterns using DNA Tile Assembly*, In Proceedings of Foundations of Nanoscience Self-Assembled Architectures and Devices (FNANO 17), SnowBird, Utah, USA, pp.126-126.[online] Available at arXiv:1607.03434 [cs.ET].

8. Shalin Shah, **Dixita Limbachiya** and Manish K. Gupta, *DNACloud: A Potential Tool for storing Big Data on DNA*, In Proceedings of Foundations of Nanoscience: Self-Assembled Architectures and Devices (FNANO 14), SnowBird, Utah, USA, 2014, pp. 204-205.

9. Shikhar Kumar Gupta, Foram Joshi, **Dixita Limbachiya** and Manish K. Gupta, *3DNA: A Tool for Lego Modeling*, In Proceedings of Foundations of Nanoscience: Self-Assembled Architectures and Devices (FNANO 14), SnowBird, Utah, USA, 2014 , pp. 190.

10. Arnav Goyal, **Dixita Limbachiya**, Shikhar Kumar Gupta, Foram Joshi, Sushant Pritmani, Akshita Sahai and Manish K. Gupta, *DNA Pen: A Tool for Drawing on a Molecular Canvas*, DNA 19 Conference 2013, Arizona, USA, 2013. (poster).

11. **Dixita Limbachiya**, Bansari Rao, Manish K Gupta, *The Art of DNA Codes: Sixteen Years of DNA Codes*, arXiv:1607.00266 [cs.IT], 2016, [online] Available at https://arxiv.org/abs/1607.00266 .

12. **Dixita Limbachiya** and Manish K Gupta, *Natural Data Storage: A Review on sending Information from now to then via Nature*, arXiv:1505.04890 [cs.IT], 2015, [online] Available at https://arxiv.org/pdf/1505.04890.pdf.

References

[1] N. Aboluion, D. H. Smith, and S. Perkins. Linear and nonlinear constructions of DNA codes with Hamming distance d, constant GC-content and a reverse-complement constraint. *Discrete Mathematics*, 312(5):1062–1075, 2012.

[2] T. Abualrub, A. Ghrayeb, and X. N. Zeng. Construction of cyclic codes over GF(4) for DNA computing. *Journal of the Franklin Institute*, 343(4):448–457, 2006.

[3] L. M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, 1994.

[4] M. Ailenberg and O. D. Rotstein. An improved Huffman coding method for archiving text, images, and music characters in DNA. *Biotechniques*, 47(3):747–754, 2009.

[5] A. Akhmetov, A. Ellington, and E. Marcotte. A highly parallel strategy for storage of digital information in living cells. *bioRxiv, [online] Available at https://www.biorxiv.org/content/early/2016/12/26/096792.full.pdf*, 2016.

[6] I. Akyildiz, M. Pierobon, S. Balasubramaniam, and Y. Koucheryavy. The internet of bio-nano things. *IEEE Communications Magazine*, 53(3):32–40, 2015.

[7] M. M. Al-Ashker. Simplex codes over the ring $\mathbb{F}_2 + u\mathbb{F}_2$. *Arabian Journal for Science and Engineering*, 30(2):277–286, 2005.

[8] M. Arita and Y. Ohashi. Secret signatures inside genomic DNA. *Biotechnology progress*, 20(5):1605–1607, 2004.

[9] C. Bancroft, T. Bowler, B. Bloom, and C. T. Clelland. Long-term storage of information in DNA. *Science*, 293(5536):1763–1765, 2001.

[10] E. B. Baum. Building an associative memory vastly larger than the brain. *Science*, 268(5210):583–585, 1995.

[11] A. Bayram, E. S. Oztas, and I. Siap. Codes over $\mathbb{F}_4 + v\mathbb{F}_4$ and some DNA applications. *Designs, Codes and Cryptography*, 80(2):379–393, 2016.

[12] N. Bennenni, K. Guenda, and A. Gulliver. Construction of codes for DNA computing by the greedy algorithm. *ACM Commun. Comput. Algebra*, 49(1):14–19, 2015.

[13] N. Bennenni, K. Guenda, and S. Mesnager. DNA cyclic codes over rings. *Advances in Mathematics of Communications*, 11(1):83–98, 2017.

[14] M. C. Bhandari, M. K. Gupta, and A. K. Lal. On $\mathbb{Z}_4$-simplex codes and their gray images. In *Proceedings of Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pages 170–179. Springer, 1999.

[15] M. Blawat, K. Gaedke, I. Huetter, X.-M. Chen, B. Turczyk, S. Inverso, B. Pruitt, and G. Church. Forward error correction for DNA data storage. *Procedia Computer Science*, 80:1011–1022, 2016.

[16] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss. A DNA-based archival storage system. *ACM SIGOPS Operating Systems Review*, 50(2):637–649, 2016.

[17] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss. Toward a DNA-based archival storage system. *IEEE Micro*, 37(3):98–104, 2017.

[18] BrainyQuote.com. Available at https://www.brainyquote.com/quotes/. In *Xplore Inc.*, 2001.

[19] I. Braslavsky, B. Hebert, E. Kartalov, and S. R. Quake. Sequence information can be obtained from single DNA molecules. *Proceedings of the National Academy of Sciences*, 100(7):3960–3964, 2003.

[20] S. Brenner and R. A. Lerner. Encoded combinatorial chemistry. *Proceedings of the National Academy of Sciences*, 89(12):5381–5383, 1992.

[21] K. Chatouh, K. Guenda, T. A. Gulliver, and L. Noui. Simplex and macdonald codes over R_q. *Journal of Applied Mathematics and Computing*, 55(1-2):455–478, 2017.

[22] Y. M. Chee and S. Ling. Improved lower bounds for constant GC-content DNA codes. *IEEE Transactions on Information Theory*, 54(1):391–394, 2008.

[23] Y. Choie and S. T. Dougherty. Codes over rings, complex lattices and hermitian modular forms. *European Journal of Combinatorics*, 26(2):145–165, 2005.

[24] G. M. Church, Y. Gao, and S. Kosuri. Next-generation digital information storage in DNA. *Science*, 337(6102):1628–1628, 2012.

[25] J. Clarke, H.-C. Wu, L. Jayasinghe, A. Patel, S. Reid, and H. Bayley. Continuous base identification for single-molecule nanopore DNA sequencing. *Nature nanotechnology*, 4(4):265–270, 2009.

[26] C. T. Clelland, V. Risca, and C. Bancroft. Hiding messages in DNA microdots. *Nature*, 399(6736):533–534, 1999.

[27] J. Davis. Microvenus. *Art Journal*, 55(1):70–74, 1996.

[28] D. Deamer, M. Akeson, and D. Branton. Three decades of nanopore sequencing. *Nature biotechnology*, 34(5):518–524, 2016.

[29] D. W. Deamer and M. Akeson. Nanopores and nucleic acids: prospects for ultrarapid sequencing. *Trends in biotechnology*, 18(4):147–151, 2000.

[30] R. Deaton, M. Garzon, R. Murphy, J. Rose, D. Franceschetti, and S. E. Stevens Jr. Reliability and efficiency of a DNA-based computation. *Physical Review Letters*, 80(2):417–420, 1998.

[31] A. Dertli and Y. Cengellenmis. On cyclic DNA codes over the rings $\mathbb{Z}_4 + w\mathbb{Z}_4$ and $\mathbb{Z}_4 + w\mathbb{Z}_4 + v\mathbb{Z}_4 + wv\mathbb{Z}_4$. *BIOMATH*, 6(2):1712167, 2017.

[32] A. G. Dimakis, P. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE Transactions on Information Theory*, 56(9):4539–4551, 2010.

[33] H. Q. Dinh, A. K. Singh, S. Pattanayak, and S. Sriboonchitta. Cyclic DNA codes over the ring $\mathbb{F}_2 + u\mathbb{F}_2 + v\mathbb{F}_2 + uv\mathbb{F}_2 + v^2\mathbb{F}_2 + uv^2\mathbb{F}_2$. *Designs, Codes and Cryptography*, 86(7):1451–1467, 2018.

[34] S. T. Dougherty. *Algebraic Coding Theory Over Finite Commutative Rings*. Springer, 2017.

[35] J. Eid, A. Fehr, J. Gray, K. Luong, J. Lyle, G. Otto, P. Peluso, D. Rank, P. Baybayan, B. Bettman, et al. Real-time DNA sequencing from single polymerase molecules. *Science*, 323(5910):133–138, 2009.

[36] I. G. Ellis. Today in science. *[online] Available at https://todayinsci.com/*, 1999.

[37] Y. Erlich and D. Zielinski. DNA fountain enables a robust and efficient storage architecture. *Science*, 355(6328):950–954, 2017.

[38] L. Faria, A. Rocha, J. Kleinschmidt, R. Palazzo, and M. Silva-Filho. DNA sequences generated by BCH codes over GF (4). *Electronics letters*, 46(3):202–203, 2010.

[39] R. A. Fisher. A system of confounding for factors with more than two alternatives, giving completely orthogonal cubes and higher powers. *Annals of Human Genetics*, 12(1):283–290, 1943.

[40] G. D. Forney Jr, N. J. Sloane, and M. D. Trott. The Nordstrom-Robinson code is the binary image of the octacode. In *Proceedings of DIMACS/IEEE workshop on Coding and Quantization*, pages 19–26, 1992.

[41] C. W. Fuller, L. R. Middendorf, S. A. Benner, G. M. Church, T. Harris, X. Huang, S. B. Jovanovich, J. R. Nelson, J. A. Schloss, D. C. Schwartz, et al. The challenges of sequencing by synthesis. *Nature biotechnology*, 27(11):1013–1023, 2009.

[42] P. Gaborit and O. D. King. Linear constructions for DNA codes. *Theoretical Computer Science*, 334(1):99–113, 2005.

[43] R. Gabrys, H. M. Kiah, and O. Milenkovic. Asymmetric Lee distance codes for DNA-based storage. *IEEE Transactions on Information Theory*, 63(8):4982–4995, 2017.

[44] R. Gabrys, E. Yaakobi, and O. Milenkovic. Codes in the damerau distance for DNA storage. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 2644–2648, 2016.

[45] A. L. Ghindilis, M. W. Smith, K. R. Schwarzkopf, K. M. Roth, K. Peyvan, S. B. Munro, M. J. Lodes, A. G. Stöver, K. Bernards, K. Dill, et al. Combimatrix oligonucleotide arrays: genotyping and gene expression assays employing electrochemical detection. *Biosensors and Bioelectronics*, 22(9-10):1853–1860, 2007.

[46] M. J. Golay. Notes on digital coding. *Proc. IEEE*, 37(6):657–657, 1949.

[47] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, and E. Birney. Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nature*, 494(7435):77–80, 2013.

[48] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark. Robust chemical preservation of digital information on DNA in silica with error-correcting codes. *Angewandte Chemie International Edition*, 54(8):2552–2555, 2015.

[49] M. Grassl. Bounds on the minimum distance of linear codes and quantum codes, 2007, [online] Available: http://www.codetables.de.

[50] E. D. Green, E. M. Rubin, and M. V. Olson. The future of DNA sequencing. *Nature News*, 550(7675):179–181, 2017.

[51] J. Gu and T. Fuja. A generalized Gilbert-Varshamov bound derived via analysis of a code-search algorithm. *IEEE Transactions on Information Theory*, 39(3):1089–1093, 1993.

[52] K. Guenda and T. A. Gulliver. Construction of cyclic codes over $\mathbb{F}_2 + u\mathbb{F}_2$ for DNA computing. *Applicable Algebra in Engineering, Communication and Computing*, 24(6):445–459, 2013.

[53] K. Guenda, T. A. Gulliver, and P. Solé. On cyclic DNA codes. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 121–125, 2013.

[54] M. K. Gupta. *On some linear codes over $\mathbb{Z}_{2^s}$*. PhD thesis, Indian Institute of Technology, Kanpur, 1999.

[55] M. K. Gupta. The quest for error correction in biology. *IEEE, Engineering in Medicine and Biology Magazine*, 25(1):46–53, 2006.

[56] F. Gursoy, E. S. Oztas, and I. Siap. Reversible DNA codes over $\mathbb{F}_{16} + u\mathbb{F}_{16} + v\mathbb{F}_{16} + uv\mathbb{F}_{16}$. *Adv. in Math. of Comm.*, 11(2):307–312, 2017.

[57] F. Gursoy, E. S. Oztas, and I. Siap. Reversible DNA codes using skew polynomial rings. *Applicable Algebra in Engineering, Communication and Computing*, 28(4):311–320, 2017.

[58] A. R. Hammons, P. V. Kumar, A. R. Calderbank, N. J. Sloane, and P. Solé. The $\mathbb{Z}_4$ -linearity of Kerdock, Preparata, Goethals, and related codes. *IEEE Transactions on Information Theory*, 40(2):301–319, 1994.

[59] T. D. Harris, P. R. Buzby, H. Babcock, E. Beer, J. Bowers, I. Braslavsky, M. Causey, J. Colonell, J. DiMeo, J. W. Efcavitch, et al. Single-molecule DNA sequencing of a viral genome. *Science*, 320(5872):106–109, 2008.

[60] T. Head. Formal language theory and DNA: An analysis of the generative capacity of specific recombinant behaviors. *Bulletin of Mathematical Biology*, 49(6):737 – 759, 1987.

[61] I. Holmes. Modular non-repeating codes for DNA storage. *arXiv preprint arXiv:1606.01799, [online] Available at https://arxiv.org/abs/1606.01799*, 2016.

[62] H. Hong, L. Wang, H. Ahmad, J. Li, Y. Yang, and C. Wu. Construction of DNA codes by using algebraic number theory. *Finite Fields and Their Applications*, 37:328–343, 2016.

[63] F. H. Hunt, S. Perkins, and D. H. Smith. Channel models and error correction codes for DNA information storage. *International Journal of Information and Coding Theory*, 3(2):120–136, 2015.

[64] K. A. S. Immink and K. Cai. Design of capacity-approaching constrained codes for DNA-based storage systems. *IEEE Communications Letters*, 22(2):224–227, 2018.

[65] G. Jacob and A. Murugan. DNA based cryptography: An overview and analysis. *Int J Emerg Sci*, 3(1):36–42, 2013.

[66] M. Jain, I. T. Fiddes, K. H. Miga, H. E. Olsen, B. Paten, and M. Akeson. Improved data analysis for the minION nanopore sequencer. *Nature methods*, 12(4):351–356, 2015.

[67] S. S. Joo, J. Kim, S. S. Kang, S. Kim, S.-H. Choi, and S. W. Hwang. Graphene-quantum-dot nonvolatile charge-trap flash memories. *Nanotechnology*, 25(25):255203, 2014.

[68] L. Kari. DNA computing: arrival of biological mathematics. *The mathematical intelligencer*, 19(2):9–22, 1997.

[69] L. Kari, E. Losseva, and P. Sosik. DNA computing and errors: a computer science perspective. *Molecular Computational Models: Unconventional Approaches*, pages 56–77, 2005.

[70] T. Kasami, S. Lin, and W. Peterson. New generalizations of the Reed-Muller codes–i: Primitive codes. *IEEE Transactions on Information Theory*, 14(2):189–199, 1968.

[71] P. Kaski and P. R. Östergård. *Classification algorithms for codes and designs*. Springer, 2006.

[72] H. M. Kiah, G. J. Puleo, and O. Milenkovic. Codes for DNA storage channels. In *Proceedings of IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015.

[73] O. D. King. Bounds for DNA codes with constant GC-content. *Electron. J. Combin*, 10(1):33, 2003.

[74] S. Kosuri and G. M. Church. Large-scale de novo DNA synthesis: technologies and applications. *Nature methods*, 11(5):499–507, 2014.

[75] Y. Krishnan and F. C. Simmel. Nucleic acid based molecular devices. *Angewandte Chemie International Edition*, 50(14):3124–3156, 2011.

[76] A. Lenz, P. H. Siegel, A. Wachter-Zeh, and E. Yaakobi. Coding over sets for DNA storage. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 2411–2415, 2018.

[77] Y. K. Leong, K. M. M. Aung, and P. S. Alexopoulos. Storage system architecture for data centers of the future. *International Journal of Advancements in Computing Technology*, 4(9):184 –192, 2012.

[78] J. Liang and L. Wang. On cyclic DNA codes over $\mathbb{F}_2 + u\mathbb{F}_2$. *Journal of Applied Mathematics and Computing*, 51(1-2):81–91, 2016.

[79] D. Limbachiya, V. Dhameliya, M. Khakhar, and M. K. Gupta. On optimal family of codes for archival DNA storage. In *Proceedings of IEEE Seventh International Workshop on Signal Design and its Applications in Communications (IWSDA)*, pages 123–127, 2015.

[80] D. Limbachiya, K. Gopal, B. Rao, and M. K. Gupta. On DNA codes using the ring $\mathbb{Z}_4 + w\mathbb{Z}_4$. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 2401–2405, 2018.

[81] D. Limbachiya and M. K. Gupta. Natural data storage: A review on sending information from now to then via nature. *arXiv:1505.04890, [online], Available at https://arxiv.org/pdf/1505.04890.pdf*, 2015.

[82] D. Limbachiya, M. K. Gupta, and V. Aggarwal. Family of constrained codes for archival DNA data storage. *IEEE Communications Letters*, 22(10):1972–1975, 2018.

[83] D. Limbachiya, B. Rao, and M. K. Gupta. The art of DNA strings: Sixteen years of DNA coding theory. *arXiv:1607.00266 [online], Available at https://arxiv.org/abs/1607.00266*, 2016.

[84] N. J. Loman and A. R. Quinlan. Poretools: a toolkit for analyzing nanopore sequence data. *Bioinformatics*, 30(23):3399–3401, 2014.

[85] F. Ma, Y. Cao, and J. Gao. On cyclic DNA codes over $\mathbb{F}_4[u]/(u^2+1)$. *International Journal of Research and Reviews in Applied Sciences*, 24(3):101, 2015.

[86] F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*. Elsevier, 1977.

[87] A. Marathe, A. E. Condon, and R. M. Corn. On combinatorial DNA word design. *Journal of Computational Biology*, 8(3):201–219, 2001.

[88] J. Messing, R. Crea, and P. H. Seeburg. A system for shotgun DNA sequencing. *Nucleic acids research*, 9(2):309–321, 1981.

[89] O. Milenkovic, R. Gabrys, H. M. Kiah, and S. M. H. Tabatabaei Yazdi. Exabytes in a test tube. *IEEE Spectrum*, 55:40–45, 05 2018.

[90] O. Milenkovic and N. Kashyap. DNA codes that avoid secondary structures. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 288–292, 2005.

[91] R. Montemanni and D. H. Smith. Construction of constant GC-content DNA codes via a variable neighbourhood search algorithm. *Journal of Mathematical Modelling and Algorithms*, 7(3):311–326, 2008.

[92] H. Mostafanasab and A. Y. Darani. On cyclic DNA codes over $\mathbb{F}_2 + u\mathbb{F}_2 + u^2\mathbb{F}_2$. In *Procceeding of the 4th Seminar on Algebra and its Applications*, pages 155–155, 2016.

[93] D. E. Muller. Application of boolean algebra to switching circuit design and to error detection. *IEEE Transactions of the IRE Professional Group on Electronic Computers*, (3):6–12, 1954.

[94] A. Niema. *The construction of DNA codes using a computer algebra system*. PhD thesis, University of Glamorgan, 2011.

[95] P. Nyrén and A. Lundin. Enzymatic method for continuous monitoring of inorganic pyrophosphate synthesis. *Analytical biochemistry*, 151(2):504–509, 1985.

[96] L. Organick, S. D. Ang, Y.-J. Chen, R. Lopez, S. Yekhanin, K. Makarychev, M. Z. Racz, G. Kamath, P. Gopalan, B. Nguyen, et al. Random access in large-scale DNA data storage. *Nature biotechnology*, 36(3):242–248, 2018.

[97] E. S. Oztas, B. Yildiz, and I. Siap. A novel approach for constructing reversible codes and applications to DNA codes over the ring $\mathbb{F}_2[u]/(u^{2k}-1)$. *Finite Fields and Their Applications*, 46:217–234, 2017.

[98] S. E. Oztas and I. Siap. Lifted polynomials over $\mathbb{F}_{16}$ and their applications to DNA codes. *Filomat*, 27(3):459–466, 2013.

[99] P. Pancoska, Z. Moravek, and U. M. Moll. Rational design of DNA sequences for nanotechnology, microarrays and molecular computers using eulerian graphs. *Nucleic acids research*, 32(15):4630–4645, 2004.

[100] S. Pattanayak and A. K. Singh. On cyclic DNA codes over the ring $\mathbb{Z}_4 + u\mathbb{Z}_4$. *arXiv:1508.02015, [online] Available at https://arxiv.org/abs/1508.02015*, 2015.

[101] E. Pettersson, J. Lundeberg, and A. Ahmadian. Generations of sequencing technologies. *Genomics*, 93(2):105–111, 2009.

[102] A. Pott. *Finite geometry and character theory*. vol. 160, Lecture Notes in Mathematics, Springer, 1995.

[103] L. Qian and E. Winfree. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332(6034):1196–1201, 2011.

[104] G. Qingji, W. Bin, Z. Changjun, W. Xiaopeng, and Z. Qiang. DNA code design based on the bloch quantum chaos algorithm. *IEEE Access*, 5:22453–22461, 2017.

[105] Q. Qiu, D. Burns, Q. Wu, and P. Mukre. Hybrid architecture for accelerating DNA codeword library searching. In *IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology, 2007. CIBCB'07.*, pages 323–330. IEEE, 2007.

[106] R. Rapley and D. Whitehouse. *Molecular biology and biotechnology*. Royal Society of Chemistry, 2015.

[107] N. Raviv, M. Schwartz, and E. Yaakobi. Rank modulation codes for DNA storage. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 3125–3129, 2017.

[108] J. Reif, H. Chandran, N. Gopalkrishnan, and T. LaBean. Self-assembled DNA nanostructures and DNA devices. *Nanofabrication Handbook*, pages 299–328, 2012.

[109] H. Reinhard, M. Gediminas, N. G. Robert, and T. T. Nguyen. A characterization of the DNA data storage channel. *arXiv:1803.03322, [online] Available at https://arxiv.org/abs/1803.03322*, 2018.

[110] H. Reinhard, S. Ilan, R. Kannan, and N. C. T. David. Fundamental limits of DNA storage systems. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 3130–3134, 2017.

[111] J. A. Reuter, D. V. Spacek, and M. P. Snyder. High-throughput sequencing technologies. *Molecular cell*, 58(4):586–597, 2015.

[112] M. Ronaghi, M. Uhlén, and P. Nyrén. A sequencing method based on real-time pyrophosphate. *Science*, 281(5375):363–365, 1998.

[113] G. L. Rosen. Examining coding structure and redundancy in DNA. *IEEE engineering in medicine and biology magazine*, 25(1):62–68, 2006.

[114] M. G. Ross, C. Russ, M. Costello, A. Hollinger, N. J. Lennon, R. Hegarty, C. Nusbaum, and D. B. Jaffe. Characterizing and measuring bias in sequence data. *Genome biology*, 14(5):R51, 2013.

[115] P. W. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 2006.

[116] G. Rozenberg and A. Salomaa. DNA computing: new ideas and paradigms. In *Proceedings of Automata, Languages and Programming*, pages 106–118. Springer, 1999.

[117] V. V. Rykov, A. J. Macula, D. C. Torney, and P. S. White. DNA sequences and quaternary cyclic codes. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 248–248, 2001.

[118] F. Sanger, S. Nicklen, and A. R. Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the national academy of sciences*, 74(12):5463–5467, 1977.

[119] C. Schoeny, F. Sala, and L. Dolecek. Novel combinatorial coding results for DNA sequencing and data storage. In *Proceedings of IEEE Asilomar Conference on Signals, Systems, and Computers*, pages 511–515, 2017.

[120] R. Selvakumar. Unconventional construction of DNA codes: group homomorphism. *Journal of Discrete Mathematical Sciences and Cryptography*, 17(3):227–237, 2014.

[121] S. Shah, D. Limbachiya, and M. K. Gupta. DNAcloud: A tool for storing big data on DNA. In *Proceedings of Foundations of Nanoscience: Self-Assembled Architectures and Devices (FNANO14), SnowBird, Utah, USA*, pages 204–205, 2014.

[122] J. Shendure, S. Balasubramanian, G. M. Church, W. Gilbert, J. Rogers, J. A. Schloss, and R. H. Waterston. DNA sequencing at 40: past, present and future. *Nature*, 550(7676):345–353, 2017.

[123] J. Shendure and H. Ji. Next-generation DNA sequencing. *Nature biotechnology*, 26(10):1135–1145, 2008.

[124] I. Siap, T. Abualrub, and A. Ghrayeb. Similarity cyclic DNA codes over rings. In *Proceedings of IEEE International Conference on Bioinformatics and Biomedical Engineering (ICBBE)*, pages 612–615, 2008.

[125] I. Siap, T. Abualrub, and A. Ghrayeb. Cyclic DNA codes over the ring $\mathbb{F}_2[u]/(u^2-1)$ based on the deletion distance. *Journal of the Franklin Institute*, 346(8):731–740, 2009.

[126] B. E. Slatko, A. F. Gardner, and F. M. Ausubel. Overview of next-generation sequencing technologies. *Current protocols in molecular biology*, 122(1):e59, 2018.

[127] B. Srinivasulu and M. Bhaintwal. Reversible cyclic codes over $\mathbb{F}_4 + u\mathbb{F}_4$ and their applications to DNA codes. In *Proceedings of IEEE International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 101–105, 2015.

[128] Y. S. Tabatabaei, Y. Yuan, J. Ma, H. Zhao, and O. Milenkovic. A rewritable, random-access DNA-based storage system. *Scientific reports*, 5:14138, 2015.

[129] S. Torgasin. *Graph-based Methods for the Design of DNA Computations*. PhD thesis, Technische Universität Hamburg, 2012.

[130] D. C. Tulpan. *Effective heuristic methods for DNA strand design*. PhD thesis, The University Of British Columbia, 2006.

[131] D. C. Tulpan, H. H. Hoos, and A. E. Condon. Stochastic local search algorithms for DNA word design. In *Proceedings of International Workshop on DNA-Based Computers*, pages 229–241. Springer, 2003.

[132] Z. Varbanov, T. Todorov, and M. Hristova. A method for constructing DNA codes from additive self-dual codes over $GF(4)$. *ROMAI Journal*, 10(2):203–211, 2014.

[133] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, et al. The sequence of the human genome. *Science*, 291(5507):1304–1351, 2001.

[134] M. Wanunu. Nanopores: A journey towards DNA sequencing. *Physics of life reviews*, 9(2):125–158, 2012.

[135] E. Winfree. *Algorithmic self-assembly of DNA*. PhD thesis, California Institute of Technology, 1998.

[136] N. Yachie, K. Sekiyama, J. Sugahara, Y. Ohashi, and M. Tomita. Alignment-based approach for durable data storage into living organisms. *Biotechnology progress*, 23(2):501–505, 2007.

[137] S. H. T. Yazdi, R. Gabrys, and O. Milenkovic. Portable and error-free DNA-based data storage. *Scientific Reports*, 7(1):5011, 2017.

[138] B. Yildiz and S. Karadeniz. Linear codes over $\mathbb{Z}_4 + u\mathbb{Z}_4$: Macwilliams identities, projections, and formally self-dual codes. *Finite Fields and Their Applications*, 27:24–40, 2014.

[139] V. Zhirnov, R. M. Zadegan, G. S. Sandhu, G. M. Church, and W. L. Hughes. Nucleic acid memory. *Nature Materials*, 15(4):366–370, 2016.