Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 2s (2024)

Applications of Nondeterministic Zerodivisor Graph

P. Shakila Banu[a], S. Naveena[b] and A. Jansy Isabella Rani[c]

[a]Assistant Professor, Department of Mathematics, Vellalar College for Women,

Erode-638012, Tamilnadu, India

[b]Research Scholar, Department of Mathematics, Vellalar College for Women,

Erode-638012, Tamilnadu, India

[c]Assistant Professor, Department of Biochemistry, Vellalar College for Women,

Erode-638012, Tamilnadu, India

E-Mail: [a] shakimeeran10@gmail.com, [b] naveena94430@gmail.com, [c] isabellarani@gmail.com

Article History:

Received: *18-03-2024*

Revised: *02-05-2024*

Accepted: *22-05-2024*

Abstract:

This research integrates concepts from zerodivisor graph Z64 to examine a multifaceted approach to mRNA odd and even codons assessments. furthermore, correction of errors has been rendered attainable with the implementation of parity codes, which increase the resilience of mRNA sequence representation. The Huffman approach promotes the encryption and decryption process of data retention reliability. The work advances mRNA sequence analysis and creates novel possibilities for assessments and conserving biological information.

Keywords: Non deterministic zerodivisor graph, mRNA codons, Huffman coding, Parity codes.

2020 subject classifications:94B05,05C50, 81P73,92D20

1. Introduction

In the field of computing, systems with a limited amount of states and changes among these states are depicted through the use of predictable and unpredictable networks. These charts are employed in numerous applications, which involves visual computing, artificial intelligence, and natural language processing. In contrast to predictable graphs, unpredictable graphs tend to be greater evocative. Kurtz[10] authored an article headlined "predictable and Unpredictable Networks" in 1973, outlining a common paradigm for both types of structures. The dissertation emphasised the significance of predictable and unpredictable networks in the field of computation and laid the groundwork for future research on the subject. Beck first pitched the idea of a zerodivisorgraph in 1998 [4]. The first researchers who streamlined Beck's zerodivisor graph seemed Anderson and Livingston [3]. Redmond modified the conceptual framework of a zerodivisor graph by integrating a ring in 2002 [16]. P. Shakila Banu and S. Naveena demonstrated in 2023 [17] that every instance of zerodivisor graphs is unpredictable zerodivisor graphs, however the contrary need not holds.

Tom Head initially proposed the concept of implementing DNA for computing around 1987, although Adleman accomplished an initial viable computer powered by DNA experiment in 1994.

Genetic hybridization is the fundamental component of DNA computing, however it may additionally contribute to inaccuracies. Consequently, error mitigation techniques are vital for the effective implementation of DNA computation.

In 2005, Yachkov [5] created the premise of proximity processes, which have proven beneficial to estimating statistical affinities throughout genomes. The correlation between persistent cyclic codes and persistent supplement cyclic codes, particularly has been shown to be crucial for DNA processing, was investigated in 2013 by K. Guenda and T. A. Gulliver [6]. DNA codes are perceived as terms over the alphabet set

$\sum = \{A, C, G, T\}$, accomplishing precise algebraic criteria. In 2016, Limbachiya [14] laid out the definitive framework for DNA, comprising of four bases, among which are Thymine (T), Adenine (A), Cytosine (C), and Guanine (G). In [12], the author created a distance-preserving Gaussian map π that generated a one-to-one connection between every one of the genome codewords of length two and the constituent components of the ring R. Researchers introduce multiple novel categories of DNA codes that comply with reverse complement limitations by utilising this representation. In [13], author explored the algebraic properties of the ring R and defined a characterised as the Gau proximity of DNA and the other components of the ring R. In 2021, Alahmadi [2] proposed that reverse complement constraints the particular codes in the ring R. In 2022, Kim, Jon-Lark, and Dong Eun Ohk [9] described the genetic codes featuring static value dispersion subject to GC-content limitations and least dispersion subject to retroactive reinforce limitations. The Huffman encoding methodology was further refined in 2009 by M. Ailenberg and O. D. Rotstein [1] for the preservation of written content, visual content, and acoustic characters in DNA. Using a Huffman encoder and receiver process by IJulia [19]. In 2022, Sultana, Nahar, Tasnim, Hossain and Andersson [18] researched about An Effective Technique for Compressing and Decoding to Reduce the Dimensions of Huffman Networks. In 2003, the article [8] dealt about the replacement polymorphism system (SPN) symmetric block crypts' parity code based concurrent error detection (CED) mechanism versus such assaults. In 2021, Rankin, David [15] implemented solely one parity verification programme for recognising inconsistencies

In this research article, we examined Non Deterministic zerodivisor graph on DNA codes for error detection.

In section 2, crucial definitions are mentioned.

In section 3, we defined a genetic code algebra as odd and even codons.

In section 4, we looked at the odd and even parity check codes for error correction. In section 5, we provide an analysis of DNA sequences using the Huffman Coding technique for message encoding and decoding.

2. Preliminaries

Hereby, stepped over over a few essential definitions which have relevance to our substantial concepts.

Definition 2.1. [12] The nucleotide is the genetic order in ribonucleic acid (RNA) and deoxyribonucleic acid (DNA) that establishes the protein amino acid chain. Antibodies are not

directly produced from DNA, despite the fact that the knowledge for the amino acid sequences is encoded in the linear sequence of nucleotides in DNA. Rather, the genetic material is utilised to generate a molecule of messenger RNA to regulate the synthesis of the amino acids. The four bases that composed into RNA are adenine (A), guanine (G), cytosine (C), and uracil (U).

Definition 2.2. [3] In the event that there is a non zero element $b \in R$ such that $ab = 0$ in R, then a not equal to zero variable $a \in R$ is termed to as a zero divisor graph.

Definition 2.3. [17] Consider a zero-divisor graph, zn. The zero-divisor graph zn is referred to as a non-deterministic zero-divisor graph if it is nondeterministic in the sense that the results from particular actions or occurrences are not precisely determined (i.e., there exists a certain number of possibilities for every vertex set).

Definition 2.4. [15] An additional bit added to a binary message that renders the total number of 1s remain odd or even. This is called a parity bit. The total number of 1s in a string of binary characters is indicated by a parity word. Even and odd parity checks are the two types of parity systems.

3. Graphical Representation of Genetic Code Algebra

In this section, the exact amino acid sequence that corresponds to an amino acid of the genetic code. As a consequence, the nucleotide sequence perceived in either messenger ribonucleic acid (mRNA). In mRNA, there are two long chains of nucleotides that complement each other: Adenine (A), Cytosine (C), Guanine (G), Uracil(U). A codon is made up of three consecutive mRNA nucleotides. Every codon designates a specific amino acid. It is possible to give the set of 64 codons a compatible ring topology to the ring of integers modulo 64$\{Z_{64}\}$. We have demonstrated how certain sorts of mutations on the bases of codons divide the entire codon set into disjoint graphs, which in turn create the entire genetic code graph. The set of non deterministic zero divisor graph in the ring Z_{64} is represented as $\{Z_{64}\}$ = {AAG, CAA, GAA, ACA, AGA}, in this set are hydrophilic codons, or codes for hydrophilic amino acids.

Theorem 3.1. The cartesian product of the set of mRNA codons forms a non-determinitic graph.

Proof. Let us denote the cartesian product of the set of mRNA codons with itself $G_\alpha \times G_\alpha$ and $G_\beta \times G_\beta$. The set of mRNA codons can be represented as, $G_\alpha = \{c_1, c_3, \ldots\ldots\ldots\ldots, c_{63}\}$ and

$G_\alpha = \{c_2, c_4, \ldots\ldots\ldots\ldots\ldots\ldots, c_{64}\}$. The cartesian product of $G_\alpha \times G_\alpha$ is defined as,

$G_\alpha \times G_\alpha = \{(c_i, c_j) | c_i, c_j \in G_\alpha\}$ and $G_\beta \times G_\beta = \{(c_i, c_j) | c_i, c_j \in G_\beta\}$.

The outcome of the event is not uniquely determined is clearly from Fig 3.1 ($G_\alpha \times G_\alpha$) & Fig 3.2 ($G_\beta \times G_\beta$).

□

The total graph of the codon set can be segregated into odd codons and even codons. Odd codons G_α = {AAC, AAU, CAC, CAU, GAC, GAU, UAC, UAU, ACC, ACU, CCC, CCU, GCC, GCU, UCC, UCU, AGC, AGU, CGC, CGU, GGC, GGU, UGC, UGU, AUC, AUU, CUC, CUU, GUC, GUU, UUC, UUU}

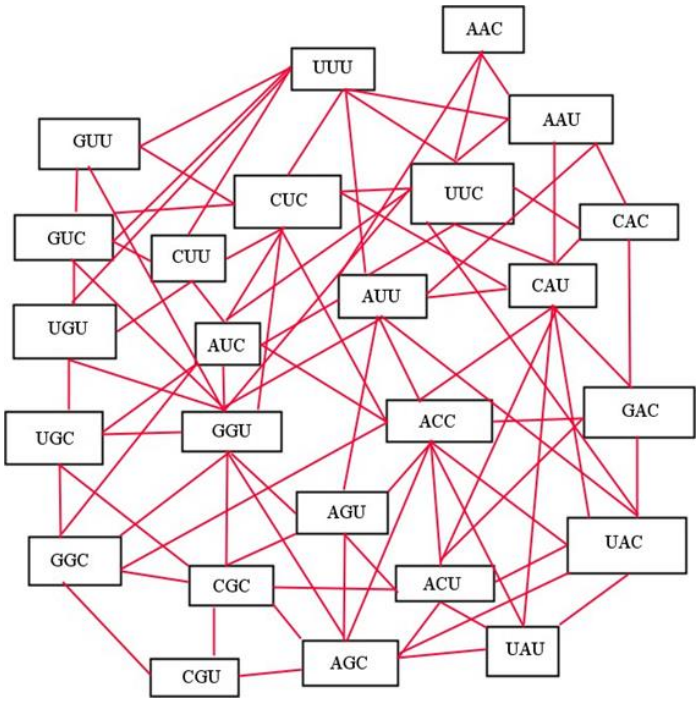


Fig 3.1 : $G_\alpha \times G_\alpha$

Even codons G_β = {AAA, AAG, CAA, CAG, GAA, GAG, UAA, UAG, ACA, ACG, CCA, CCG, GCA, GCG, UCA, UCG, AGA, AGG, CGA, CGG, GGA, GGG, UGA, UGG, AUA, AUG, CUA, CUG, GUA, GUG, UUA, UUG.}

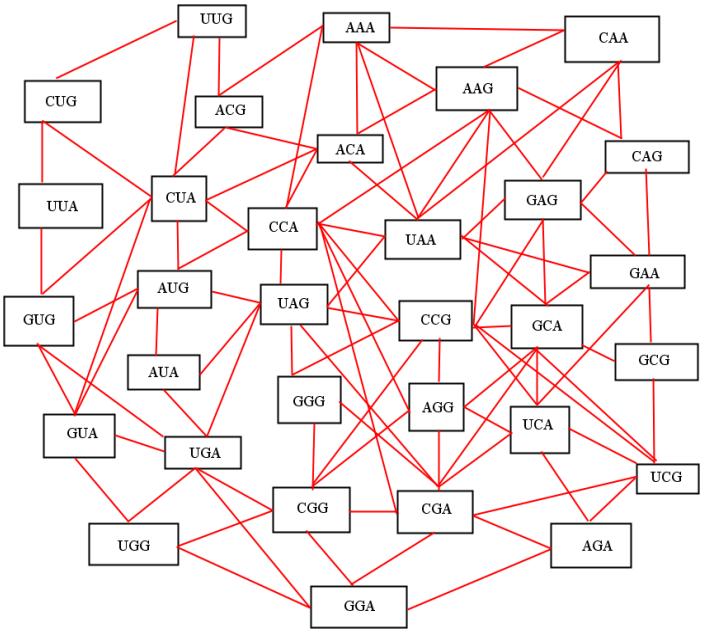


Fig 3.2: $G_\beta \times G_\beta$

We can obtain the total graph of $\{z_4 \times z_4 \times z_4\}$ as eight disjoint graphs such as,

- The amino acids of eight set of codons of G_1 are Asparagine and Tyrosine.

  $G_1 = \{AAC, AAU, CAC, CAU, GAC, GAU, UAC, UAU\}$

- The amino acids of eight set of codons of G_2 are Threonine and Proline.

  $G_2 = \{ACC, ACU, CCC, CCU, GCC, GCU, UCC, UCU\}$

- The amino acids of eight set of codons of G_3 are Serine and Cysteine.

  $G_3 = \{AGU, AGC, CGU, CGC, GGC, GGU, UGU, UGC\}$

- The amino acids of eight set of codons of G_4 are Isoleucine and Leucine.

  $G_4 = \{AUU, AUC, CUU, CUC, GUC, GUU, UUU, UUC\}$

- The amino acids of eight set of codons of G_5 are Lysine and Glutamine.

  $G_5 = \{AAG, AAA, CAG, CAA, GAG, GAA, UAG, UAA\}$

- The amino acids of eight set of codons of G_6 are Threonine and Alanine.

  $G_6 = \{ACG, ACA, CCG, CCA, GCG, GCA, UCG, UCA\}$

- The amino acids of eight set of codons of G_7 are Arginine and Glycine.

  $G_7 = \{AGG, AGA, CGG, CGA, GGG, GGA, UGG, UGA\}$

- The amino acids of eight set of codons of G_8 are Isoleucine and Methionine.

  $G_8 = \{AUG, AUA, CUG, CUA, GUG, GUA, UUG, UUA\}$

Addition modulo is defined in the set of four bases of the mRNA such as

$(x + y) \bmod 4 = z$. Therefore, addition table is as follows:

+	A	C	G	U
A	A	C	G	U
C	C	G	U	A
G	G	U	A	C
U	U	A	C	G

Table 3.1

Theorem 3.2. If two odd mRNA codons are added together, then their sum results in an even codon.

Proof. It can be readily through the following; We define a function $\phi : G_\alpha \rightarrow G_\beta$ such that,

$$\emptyset(xyz) = \begin{cases} xyA, & \text{if } z = C \\ xyG, & \text{if } z = U \end{cases} \qquad \forall\, xyz \in G_\alpha$$

Example: Assume an odd codon AAC which is an element of G_α, $\emptyset(AAC) = AAC + CCC = CCG$. Then the even nucleotide CCG is an element of G_β . □

Theorem 3.3. If two odd mRNA codons are added together, then their sum results in an odd codon.

Proof. It becomes clear when considering the following: we defined a function $\emptyset_\beta = G_\beta \to G_\alpha$ such that,

$$\emptyset(xyz) = \begin{cases} xyC, & \text{if } z = A \\ xyU, & \text{if } z = G \end{cases} \qquad \forall\, xyz \in G_\beta$$

Example: Assume even codon CCG which is an element of G_β, $\phi(CCG) = CCG + GGG = UUU$. Then the odd nucleotide UUU is an element of G_α.

The genome chart reveals the order prompted {A, G, C, U}

$\{z_{64}\}$	Binary number	Codon	Amino acid	z_{64}	Binary number	Codon	Amino acid
0	0	AAA	K	33	100001	AGC	S
1	1	AAC	N	34	100010	AGG	R
2	10	AAG	K	35	100011	AGU	S
3	11	AAU	N	36	100100	CGA	R
4	100	CAA	Q	37	100101	CGC	R
5	101	CAC	H	38	100110	CGG	R
6	110	CAG	Q	39	100111	CGU	R
7	111	CAU	H	40	101000	GGA	G
8	1000	GAA	E	41	101001	GGC	G
9	1001	GAC	D	42	101010	GGG	G
10	1010	GAG	E	43	101011	GGU	G
11	1011	GAU	D	44	101100	UGA	—
12	1100	UAA	—	45	101101	UGC	C
13	1101	UAC	Y	46	101110	UGG	W
14	1110	UAG	—	47	101111	UGU	C
15	1111	UAU	Y	48	110000	AUA	I
16	10000	ACA	T	49	110001	AUC	I
17	10001	ACC	T	50	110001	AUG	M
18	10010	ACG	T	51	110010	AUU	I
19	10011	ACU	T	52	110011	CUA	L
20	10100	CCA	P	53	110100	CUC	L
21	10101	CCC	P	54	110101	CUG	L
22	10110	CCG	P	55	110110	CUU	L
23	10111	CCU	P	56	110111	GUA	V
24	11000	GCA	A	57	111000	GUC	V
25	11001	GCC	A	58	111001	GUG	V

26	11010	GCG	A	59	111010	GUU	V
27	11011	GCU	A	60	111011	UAA	L
28	11100	UCA	S	61	111100	UUC	F
29	11101	UCC	S	62	111101	UUG	L
30	11110	UCG	S	63	111111	UUU	F
31	11111	UCU	S				
32	100000	AGA	R				

Table 3.2: DNA Codons for Amino acid

4. Odd and even parity generator and parity checker in mRNA codes

In this section, during the transmission and processing of binary data by digital systems, the introduction of noise can lead to alterations, flipping zero values towards ones. This computation is feasible, and among the least prevalent approaches to communicating information to fix errors is the parity generating technique using p⊕ q ⊕ r ⊕……. ⊕ n ⊕ P.

$\{z_{64}\}$	Binary number	0dd Parity	Even Parity	z_{64}	Binary number	Odd Parity	Even Parity
0	0	1	0	33	100001	1	0
1	1	0	1	34	100010	1	0
2	10	0	1	35	100011	0	1
3	11	1	0	36	100100	1	0
4	100	0	1	37	100101	0	1
5	101	1	0	38	100110	0	1
6	110	1	0	39	100111	1	0
7	111	0	1	40	101000	1	0
8	1000	0	1	41	101001	0	1
9	1001	1	0	42	101010	0	1
10	1010	1	0	43	101011	1	0
11	1011	0	1	44	101100	0	1
12	1100	1	0	45	101101	1	0
13	1101	0	1	46	101110	1	0
14	1110	0	1	47	101111	0	1
15	1111	1	0	48	110000	1	0
16	10000	0	1	49	110001	0	1
17	10001	1	0	50	110001	0	1
18	10010	1	0	51	110010	0	1
19	10011	0	1	52	110011	1	0
20	10100	1	0	53	110100	0	1
21	10101	0	1	54	110101	1	0
22	10110	0	1	55	110110	1	0
23	10111	1	0	56	110111	0	1
24	11000	1	0	57	111000	0	1

25	11001	0	1	58	111001	1	0
26	11010	0	1	59	111010	1	0
27	11011	1	0	60	111011	0	1
28	11100	0	1	61	111100	1	0
29	11101	1	0	62	111101	0	1
30	11110	1	0	63	111111	1	0
31	11111	0	1				
32	100000	0	1				

Table 3.3: Odd and Even Parity Generator

Therefore, a parity bit is added to the word containing the data. There is a data error when a message contains the count reaches ones at the receiving end is counted and it differs from the one that was broadcast. The entirety of the quantity of ones will be even parity bit. when the additional parity bit is used, and odd parity when the added parity bit is used. The fundamental idea behind parity network implementation is that the total of an odd number of 1′s is always 1, and the sum of an even number of ones is always zero.

Parity Checker: Imagine that the sender point receives three input messages and an even parity bit. The parity detector circuit utilises these bits as input to determine whether the data comprises imperfections. Considering the even parity of the data dissemination, the four bits intercepted at the circuit must include a pair of one's.

$$[E] = \begin{cases} 1, & if\ the\ error\ occurs \\ 0, & Otherwise \end{cases}$$

The received message involves a small number of one's if there is an oversight. PEC (Parity Error Check) is the return value of the parity detector.

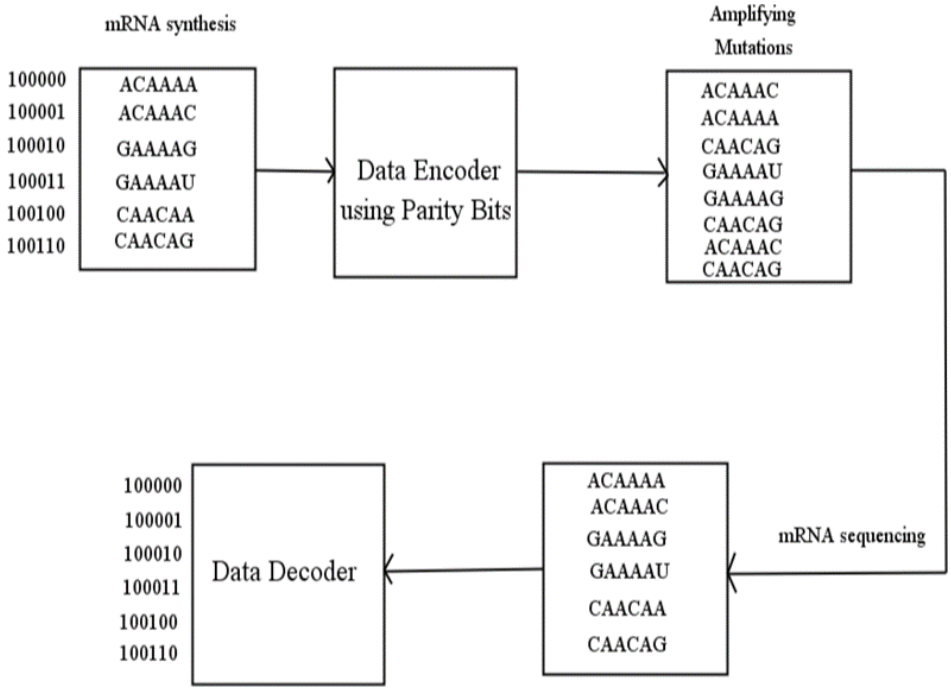


Fig 3.3: Encoding and Decoding with error correcting sequence

5. Efficient Data Compression and Decoding using Huffman Coding Algorithm in mRNA Sequence Analysis

In this segment, we propose the formula to enhance the efficiency of information storage during both the encoding and decoding processes.

Algorithm:

Step 1: Scan the mRNA sequence to tally the occurrences of each base and convert them into binary data. Subsequently, generate nodes for each base containing its mRNA counterpart along with its binary representation.

Step 2: Create a priority queue using a max-heap structure, prioritizing nodes based on the abundance of binary data associated with each mRNA base. The proportion at which the mRNA bases appear establishes the priority.

Step 3: Whenever there is just one node remains in the priority queue, keep doing the following actions that begin extract binary data.

Step 4: Create a new internal node whose binary data is the aggregate of the two derived nodes data in binary. The two previously abolished nodes have been assigned this novel node as the underlying node. The new internal node ought to be reinstated into the priority queue.

Step 5: This tree is constructed in a manner where the mRNA bases are positioned as leaves, and the route from the root to each base symbolizes its code of varying lengths.

Step 6: Explore the Huffman tree to allocate binary codes to individual mRNA bases. Utilize ’1’ to represent a left branch and ’0’ for a right branch. These codes are formulated according to the route taken from the root to each leaf node.

Step 7: Create a table or dictionary that maps each mRNA base to its corresponding Huffman code.

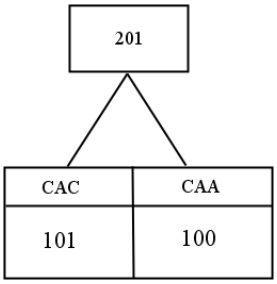
Step 8: Replace each mRNA base in the original sequence with its Huffman code to generate the compressed data.

Step 9: Decode the compressed data back to its original mRNA sequence.

The subsequent methodology evaluates the calculation method.

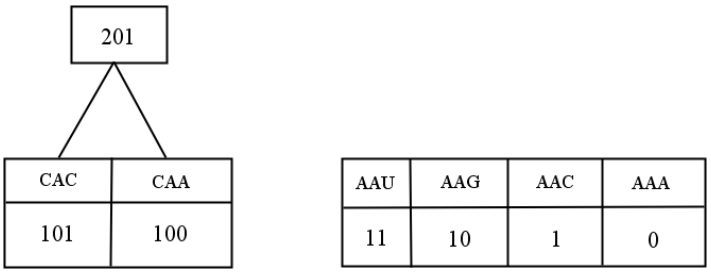
Codons	AAA	AAC	AAG	AAU	CAA	CAC
Binary number	0	1	10	11	100	101

Step 1: Determine the binary data of each mRNA base in the input data by scanning the sequence and counting occurrences, then create nodes for each base holding its mRNA representation along with its binary data.

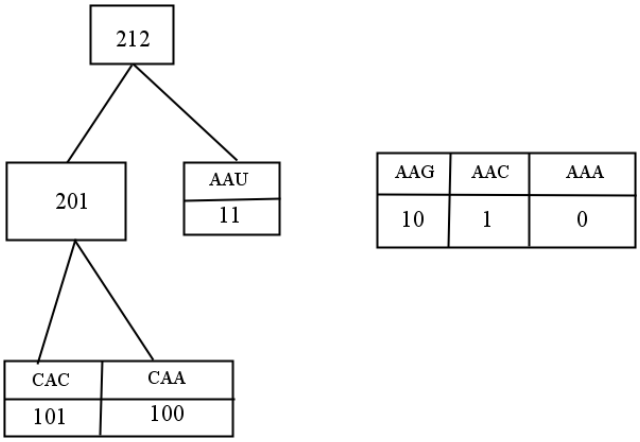


Step 2: For each mRNA base in the input with a binary data, create a node containing that base DNA.

Step 3: for (max-heap) using the initial nodes. The priority queue is



based on the frequencies of the mRNA bases, with nodes having large binary data having higher priority.

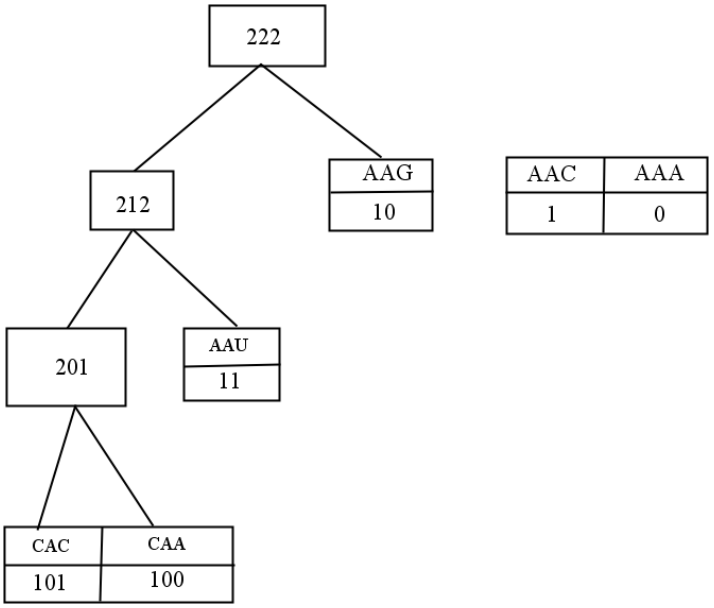


Step 4: Whenever there is just a single node in the priority queue, keep doing the following:

i. Take the top two nodes in the priority queue for binary data extraction.

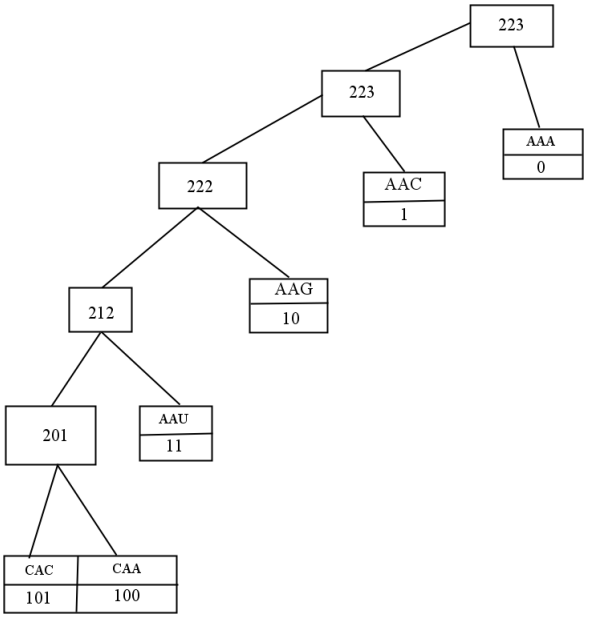
ii. With binary data equal to the total of the binary data from the two extracted nodes, create a new internal node. Assign the two extracted nodes to this new node as their parents.

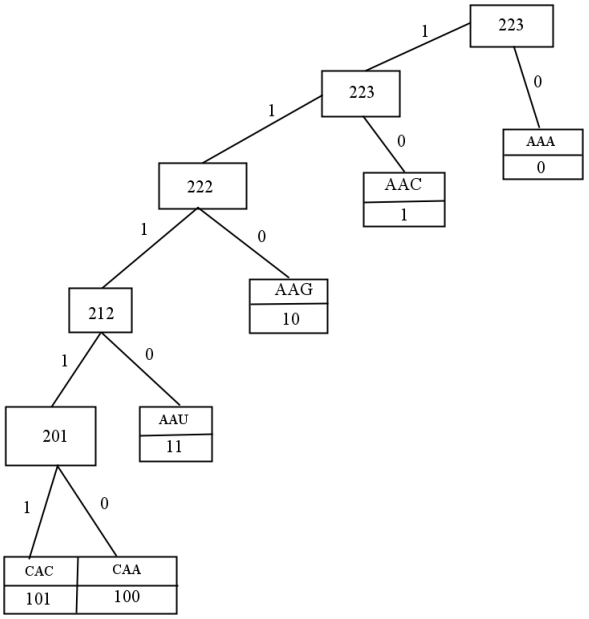
iii. Re-add the newly created internal node to the priority queue.



Step 5: The tree is built such that the mRNA bases are leaves, and the path from the root to each base represents its variable-length code.

Step 6: The codes are constructed based on the path from the root to each leaf.





Encoded mRNA sequence "AACAAG" decoded as "10110," corresponds to the nucleotide triplets CAC (11111), CAA (11110), AAU (1110), AAG (10), and AAA (0), resulting in the amino acid sequence NK.

6. Conclusion

We have bridged the gap between mathematics and biology, uncovering hidden patterns within mRNA sequences using zero divisor graph Z_{64}. By utilizing power of algorithms like Huffman and parity codes and turbocharged the efficiency of DNA sequence representation and analysis. With the implementation of error correction techniques, we are paving the way for error-free decoding of complex genetic codes, ensuring accuracy and reliability in genetic research. Our exploration of nondeterministic zero divisor graphs has unveiled exciting prospects for understanding and manipulating mRNA, offering a glimpse into the future of genetic research and technology.

References

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

[2] Alahmadi, Adel DNA Codes over the Ring $Z_4 + wZ_4$, 2021.
arXiv preprint arXiv:2110.09089.

[3] Anderson, D. F. and Livingston, P. S., The Zero-divisor graph of a commutative ring, InternationalJournal of commutative Ring, vol.2, 2002, 203-211.

[4] Beck.I, Coloring of Commutative Rings, journal of Algebra, vol.1, 1988, 56-57.

[5] D'yachkov, A.G., Vilenkin, P.A., Ismagilov, I.N.K.E., Sarbaev, R.S., Macula, A., Torney, D. and White, S On DNA codes, Problems of Information Transmission, vol.41, 2005, pp.349-367.

[6] K. Guenda and T. A. Gulliver Construction of cyclic codes over F2 + uF2 for DNA computing, Applicable Algebra in Engineering, Communication and Computing, vol.24(6), 2013, pp.445–459.

[7] 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, vol.37, 2016, pp.328–343.

[8] Karri, Ramesh, Grigori Kuznetsov, and Michael Goessel Parity-based concurrent error detection of substitution-permutation network block ciphers, proceedings Springer Berlin Heidelberg, 2003.

[9] Kim, Jon-Lark, and Dong Eun Ohk DNA codes over two noncommutative rings of order four, Journal of Applied Mathematics and Computing, vol.68(3), 2022, pp.2015-2038.

[10] Kurtz.A, Deterministic and Nondeterministic Graphs, journal of Information and Control, vol.22, 1973, 205-226.

[11] 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), 2018, pp. 2411–2415.

[12] Limbachiya, D., Benerjee, K. G., Rao, B., Gupta, M. K. On DNA Codes using the Ring $Z_4 + wZ_4$. In IEEE International Symposium on Information Theory, 2018, pp. 2401-2405.

[13] Limbachiya, Dixita On designing DNA codes and their applications, Institute of Information and Communication Technology, 2019.

[14] Limbachiya, D., Rao, B. and Gupta, M.K., The art of DNA strings Sixteen years of DNA coding theory, 2016.
arXiv preprint arXiv:1607.00266.

[15] Rankin, David M., and T. Aaron Gulliver Single parity check product codes, IEEE Transactions on Communications vol.49(8), 2021, pp.1354-1362.

[16] Redmond S.P, The zerodivisor graph of a noncommutative ring, Interna- tional journal of commutative ring, vol.1, 2002, 203-211.

[17] Shakila Banu .P and Naveena .S Non Deterministic Zero Divisor Graph, Ratio Mathematica, vol.48, 2023, pp.222 – 230.

[18] Sultana, Z., Nahar, L., Tasnim, F., Hossain, M. S., Andersson .K. An Encoding and Decoding Technique to Compress Huffman Tree Size in an Efficient Manner In International Conference on Intelligent Computing Optimization, 2022, pp. 863-873.

[19] V. G. Vidhyaa, S. A. Rajalakshmi, R. Raghavan, G. S. V. Venu Gopal and R. Gandhiraj Huffman encoding and decoding algorithm using IJulia, International Conference on Communication and Signal Processing (ICCSP), 2016, pp. 0587-0591.
doi: 10.1109/ICCSP.2016.7754207.