UNIVERSIDADE FEDERAL DE ALFENAS

ALICE NORONHA DE OLIVEIRA

CÓDIGOS BCH APLICADOS NO PROCESSO DE ANÁLISE DE FENÔMENOS MUTACIONAIS

Alfenas/MG
2020

ALICE NORONHA DE OLIVEIRA

CÓDIGOS BCH APLICADOS NO PROCESSO DE ANÁLISE DE FENÔMENOS MUTACIONAIS

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Estatística Aplicada e Biometria pelo Programa de Pós-graduação em Estatística Aplicada e Biometria. Área de concentração: Estatística Aplicada e Biometria.

Orientador: Prof. Dr. Anderson José de Oliveira.

Alfenas/MG
2020

Dados Internacionais de Catalogação-na-Publicação (CIP)
Sistema de Bibliotecas da Universidade Federal de Alfenas

O48c

Oliveira, Alice Noronha de.
Códigos BCH aplicados no processo de análise de fenômenos mutacionais / Alice Noronha de Oliveira. - Alfenas/MG, 2020.
87f.: il. --

Orientador: Anderson José Pereira.
Dissertação (Mestrado em Estatística Aplicada e Biometria) - Universidade Federal de Alfenas, 2020.
Bibliografia.

1. Álgebra. 2. Códigos BCH. 3. DNA. 4. Enzima ATP6. I. Pereira Anderson José. II. Título.

CDD-512

Ficha Catalográfica elaborada por Fátima dos Reis Goiatá
Bibliotecária-Documentalista CRB/6-425

Alice Noronha de Oliveira

Códigos BCH Aplicados no Processo de Análise de Fenômenos Mutacionais

A Banca examinadora abaixo-assinada aprova a Dissertação apresentada como parte dos requisitos para a obtenção do título de Mestre em Estatística Aplicada e Biometria pela Universidade Federal de Alfenas. Área de concentração: Estatística Aplicada e Biometria.

Aprovada em: 19 de junho de 2020.

Prof. Dr. Anderson José de Oliveira
Instituição: Universidade Federal de Alfenas - UNIFAL-MG

Profa. Dra. Cátia Regina de Oliveira Quilles Queiroz
Instituição: Universidade Federal de Alfenas - UNIFAL-MG

Prof. Dr. Evandro Monteiro
Instituição: Universidade Federal de Alfenas - UNIFAL-MG

Documento assinado eletronicamente por **Anderson Jose de Oliveira**, **Professor do Magistério Superior**, em 23/06/2020, às 13:46, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Evandro Monteiro**, **Professor do Magistério Superior**, em 23/06/2020, às 14:08, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Cátia Regina de Oliveira Quilles Queiroz**, **Professor do Magistério Superior**, em 23/06/2020, às 15:30, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.unifal-mg.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0317034** e o código CRC **6BCF2B10**.

Dedico todo o esforço que depositei neste trabalho ao meu filho, Francisco José de Oliveira Gonçalves.

AGRADECIMENTOS

Primeiramente a Deus, que foi um verdadeiro guia nessa jornada. Agradeço por ter permitido que eu tivesse determinação e saúde para não desanimar durante a realização desse trabalho. O Senhor foi minha maior força nos momentos difíceis.

Ao meu orientador, Anderson José, que foi incumbido da árdua tarefa de me conduzir nesse trabalho, a quem devo muito pela paciência, compreensão e disponibilidade em me orientar. Obrigada por ser esse pai acadêmico tão amigo. Deixo um agradecimento especial ao colaborador Diogo Guilherme, agora amigo, pela sua grande contribuição nessa pesquisa, além de me socorrer no LaTex.

Ao meu marido, Gustavo, pelo apoio, amor, amizade e paciência. Sem a sua ajuda, cuidando do nosso Cicico, eu não teria realizado o sonho de concluir o mestrado. Ao meu filho Francisco, o maior amor da minha vida, que me fez perceber que eu sou mais forte do que imaginava. Esse trabalho é uma conquista nossa.

Aos aos meus pais Givany e Francisco Ozanan, que todos os dias me deram forças para superar as dificuldades e me ajudaram a cuidar do Francisco no período que eu estava em Alfenas. Obrigada por tudo, por orarem e chorarem comigo, por acreditarem e se orgulharem com cada uma das minhas conquistas. Amo vocês.

Às minhas irmãs, sobrinhos e cunhados por estarem tão presentes em todos os momentos da vida. Vocês me mostram o quanto sou especial. A vocês minha eterna gratidão.

A todos os meus amigos, em especial Nayara, Roberta, Larissa e Thainá, com que pude compartilhar alegrias e tristezas. Tudo seria bem mais díficil se vocês não estivessem aqui.

A todos do programa de Pós-graduação em Estatística Aplicada e Biometria, por todo aprendizado e apoio durante o mestrado. À Universidad Federal de Alfenas por proporcionar essa oportunidade, a Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) pelo apoio financeiro e a CAPES, pois o presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

RESUMO

A teoria dos códigos corretores de erros tem como objetivo desenvolver métodos capazes de detectar e corrigir erros que possam surgir durante a transmissão ou armazenamento de dados. Embora não apresente relação aparente, estudos recentes evidenciam a utilização dos códigos corretores de erros também na transmissão e armazenamento de informações genéticas, verificando a existência de uma estrutura matemática relacionada com a estrutura do DNA. Desse modo, sequências de DNA podem ser identificadas e reproduzidas como palavras código de códigos corretores de erros sobre a extensão de um anel de Galois. Para a reprodução dessas sequências de DNA utiliza-se os códigos BCH, os quais possuem características simples, porém com alto poder de detecção de erros, tornando esses códigos eficientes para serem aplicados no contexto biológico. Este trabalho tem como objetivo um estudo da estrutura algébrica de um código BCH, além da reprodução de uma sequência de DNA relacionada à proteína mitocondrial ATP6, por meio dos códigos BCH, a fim de identificar onde ocorre a troca de nucleotídeo na sequência gerada. Na reprodução dessa sequência foi possível observar que ocorre a troca de um nucleotídeo na posição da trinca 17, acarretando em uma mutação não silenciosa. Assim, será analisado como essa alteração pode modificar a arquitetura biológica da sequência gerada. Os resultados dessas análises poderão ser úteis no estudo de mutações genéticas, pois a troca de um aminoácido na enzima ATP6 pode acarretar em algumas doenças genéticas.

Palavras-chave: Álgebra. Códigos BCH. DNA. Enzima ATP6.

ABSTRACT

The error-correcting codes theory aims to develop methods capable of to detect and correct errors that may arise during the transmission or storage of data. Although there is no apparent relationship, recent studies has shown the use of error-correcting codes also in the transmission and storage of genetic information verifying the existence of a mathematical structure related to the DNA structure. In this way, DNA sequences can be identified and reproduced as codewords for error-correcting codes about the Galois field extension. For the reproduction of these DNA sequences, the BCH codes are used, which have simple features, but with high error detection power, making these codes efficient to be applied in the biological context. This work aims to study the algebraic structure of a BCH code, in addition to the reproduction of a DNA sequence related to the mitochondrial protein ATP6, through the BCH codes, in order to identify where the nucleotide exchange occurs in the generated sequence. In the reproduction of this sequence, it was observed that a nucleotide is exchanged at the position of crack 17, resulting in a non-silent mutation. Thus, it will be analyzed how this change can modify the biological architecture of the generated sequence. The results of these analyzes may be useful in the study of genetic mutations, since the exchange of an aminoacid in the enzyme ATP6 can lead some genetic diseases.

Keywords: Algebra. BCH codes. ADN. ATP6 enzyme.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A necessidade de se garantir que uma mensagem seja armazenada ou transmitida de forma confiável pelos diversos meios existentes, exige o uso da teoria dos códigos corretores de erros. Propostos inicialmente por R. W. Hamming, no final da década de 40, os códigos corretores de erros são atualmente utilizados desde o processo de comunicação via satélite até o processo de comunicação celular em um sistema biológico.

Embora o processo de transmissão de informação seja considerado, na comunicação digital, como um processo que realiza a passagem de sinais através dos meios físicos de comunicação, a transmissão de informação também pode ser considerada no contexto genético. Nesse contexto, a transmissão de informação é realizada por meio do processo de transcrição, em que a informação obtida no DNA é transformada em proteínas. Desse modo, é natural questionar se existe uma estrutura intrínseca de códigos corretores de erros na transmissão de informação genética, isto é, se sequências de DNA podem ser reproduzidas mediante a utilização de códigos corretores de erros.

Rocha (2010) e Faria (2010) apresentam uma resposta afirmativa para esse questionamento. Eles apontaram a existência de um código matemático que transcreve a sequência de DNA, bem como formas de reproduzir e classificar matematicamente essas sequências, utilizando uma analogia entre um modelo de um sistema de comunicação digital e um modelo de sistema de comunicação de informação genética. Essa correspondência possibilitou o desenvolvimento de um algoritmo de geração de proteínas sobre as estruturas algébricas de corpos e anéis, que auxilia no processo de análise de fenômenos mutacionais.

Desde então, alguns pesquisadores têm aprofundado seus conhecimentos no estudo da identificação de sequências de DNA via códigos corretores de erros, a fim de demonstrar propriedades e fatos que foram obtidos em Rocha (2010) e Faria (2011) e que não foram provados; bem como o de otimizar o algoritmo de geração de proteínas, para que esse programa seja capaz de reproduzir sequências de comprimentos maiores.

Utilizando o modelo de sistema de comunicação genética, proposto por Rocha (2010) e Faria (2011), Gonzalez (2017) validou as hipóteses que o ribossomo age como o modulador do sistema biológico e que as proteínas possuem uma estrutura similar à estrutura de um código corretor de erros. Para validar as hipóteses, foram mostrados que as sequências mRNA podem ser codificadas por um código BCH sobre $\mathbb{Z}_4$ ou $\mathbb{F}_4$ e que o ribossomo age como o mapa sobrejetor de códons para os aminoácidos, além de verificar que as proteínas podem ser codificadas por um código cíclico sobre $\mathbb{Z}_{20}$ ou $\mathbb{F}_4 \times \mathbb{Z}_5$, construído através de códigos BCH.

O desenvolvimento do algoritmo de geração de proteínas por Rocha (2010) e Faria (2011) mostrou os benefícios de se associar tecnologias computacionais com as análises da biotecnologia para o estudo de análises mutacionais, no entanto, devido à complexidade

computacional do algoritmo, sequências de comprimentos extensos eram inviáveis de serem geradas. Assim, Pereira (2014) propôs a criação de um banco de dados que calcula diversos polinômios geradores que serão usados por um outro programa que se utiliza destes polinômios geradores para realizar análises em sequências de DNA e identificar palavras código na forma de novas sequências de DNA, possibilitando o aprimoramento computacional do algoritmo de geração de proteínas sobre a estrutura algébrica de anel.

Diferentes sequências de DNA, com funcionalidades biológicas diversas, podem ser reproduzidas pelo algoritmo. No entanto, elas devem atender a duas restrições: 1) a sequência de DNA deve ter comprimento $n = 2^r - 1$, onde r denota o grau da extensão e 2) é preciso obter uma sequência de DNA original postada no NCBI (National Center for Biotechnology Information).

A enzima mitocondrial ATP sintase, também conhecida como complexo V, está localizada na parte interna da mitocôndria e a sua principal função é sintetizar a maior parte do ATP (adenosina trifosfato) que é a principal forma de energia química, viabilizando a maioria das reações metabólicas que ocorrem no interior das células. Essa enzima é composta por diferentes subunidades proteicas, dentre elas a subunidade a da membrana de ATP sintase, ATP6, que possui comprimento de $2^6 - 1 = 63$ nucleotídeos, e atende as restrições do algoritmo. A importância da análise dessa sequência, via códigos corretores de erros, reside no fato que alterações na sequência ATP6 acarretam em doenças mitocondriais graves como, por exemplo, a doença de NARP, o diabetes mellius e o hipogonadismo hipergonadotrópico.

Por meio deste trabalho, temos como propósito apresentar um estudo da estrutura algébrica de um código BCH e gerar a sequência de DNA associada ao ATP6, por meio do algoritmo de geração de proteínas, a fim de localizar onde ocorreu uma mutação, de efeito maléfico, nessa molécula de DNA e analisar como essa alteração pode interferir na produção de ATP, pela proteína ATP6. Por meio dessa análise é possível mostrar algebricamente a mutação que ocorre nessa sequência e as consequências bioquímicas acerca dessa mutação.

Este trabalho está estruturado da seguinte forma: no Capítulo 2 serão apresentados, de forma sucinta, conceitos básicos dos quais depende este trabalho como um todo. São conceitos elementares relacionados a álgebra abstrata, códigos corretores de erros e biologia. No Capítulo 3 será apresentado o algoritmo de geração de proteínas que auxilia na identificação de possíveis mutações em sequências de DNA, por meio dos códigos corretores de erros. É mostrada ainda a aplicação desse algoritmo em um exemplo da construção do código BCH sobre anéis e a análise das sequências reproduzidas pelo algoritmo de geração de proteínas. No Capítulo 4 será apresentada a geração da sequência de DNA relacionada à enzima mitocondrial ATP6 via códigos corretores de erros, bem como a análise dos resultados obtidos por meio da simulação. No Capítulo 5 serão apresentadas as conclusões deste trabalho e indicações de alguns tópicos para estudos futuros.

2 REFERENCIAL TEÓRICO

Em virtude da interdisciplinaridade desta pesquisa, este capítulo introduz conceitos relacionados à álgebra abstrata, códigos corretores de erros e biologia molecular.

O presente capítulo está organizado da seguinte forma: Na Seção 2.1 são apresentados os principais conceitos e definições das estruturas algébricas de grupos, anéis, corpos e extensões de Galois. Essas estruturas são fundamentais na teoria dos códigos corretores de erros, pois facilitam o processo de codificação e decodificação. Na Seção 2.2 é apresentada uma breve introdução à teoria dos códigos corretores de erros, códigos cíclicos, códigos cíclicos sobre anéis residuais e os elementos principais dos códigos BCH sobre anéis. Na Seção 2.3 é realizada uma breve introdução dos principais conceitos referentes ao DNA, com o objetivo de compreender o processo de geração de proteínas, sendo esse processo fundamental para entender como ocorrem as mutações no sistema biológico. Por fim, na Seção 2.4 são apresentadas as conexões existentes entre a teoria de comunicação padrão e a teoria de comunicação da informação genética.

2.1 ÁLGEBRA

Nesta seção serão apresentados os conceitos de algumas estruturas algébricas, tais como grupos, anéis, corpos e corpos de Galois, utilizados ao longo do trabalho. Esses conceitos são fundamentais na teoria de códigos corretores de erros, pois facilitam os processos de codificação e decodificação de uma palavra transmitida, e estão detalhados da seguinte forma: na Subseção 2.1.1 são apresentados a definição, teoremas, conceitos e exemplos de grupos. Na Subseção 2.1.2 são apresentados a definição de anéis, bem como os principais teoremas relacionados a essa estrutura algébrica. Na Subseção 2.1.3 é apresentada a estrutura de corpos e suas propriedades, com exemplos, e finalmente nas Subseções 2.1.4, 2.1.5 e 2.1.6 é apresentada a estrutura de corpos de Galois e suas propriedades, com foco nos corpos de Galois binários.

Os conceitos, definições e exemplos podem ser encontrados em Costelo e Lin (1983) e Domingues e Iezzi (2003).

2.1.1 Grupos

Definição 2.1.1. *Um conjunto não vazio G e uma operação $(x, y) \mapsto x * y$ sobre G é chamado de grupo, se essa operação satisfazer as seguintes propriedades:*

a) associatividade: $a * (b * c) = (a * b) * c$, quaisquer que sejam $a, b, c \in G$;

b) existência de elemento neutro: existe um elemento $e \in G$ tal que $e*a = a*e = a$, qualquer que seja $a \in G$;

c) existência de elemento inverso: para todo $a \in G$ existe um elemento $a' \in G$, tal que $a * a' = a' * a = e$.

Se, além disso, a propriedade da comutatividade for válida para o grupo G, isto é, $a * b = b * a$ para quaisquer $a, b \in G$, então esse grupo é chamado de *grupo abeliano ou comutativo.*

Os teoremas a seguir decorrem da definição de grupo.

Teorema 2.1.1. *O elemento neutro de um grupo é único.*

Demonstração. *Suponhamos que existem dois elementos neutros $e, e' \in G$, assim temos que,*

$$e' = e' * e = e,$$

ou seja, $e' = e$.

Portanto, existe um único elemento neutro em G. ■

Teorema 2.1.2. *O elemento inverso de um grupo é único.*

Demonstração. *Suponhamos que para cada $a \in G$ que existem dois elementos inversos $a', a'' \in G$. Então,*

$$a' = a' * e = a' * (a * a'') = (a' * a) * a'' = e * a'' = a'',$$

ou seja, $a' = a''$.

Isto implica que a' e a'' são iguais e existe um único inverso para o elemento a. ■

Exemplo 2.1.1. *O conjunto $\mathbb{Z}$ dos inteiros é um grupo comutativo sob a adição. Assim, o elemento neutro é o 0 e o inteiro $-a$ é o oposto de a, para todo $a \in \mathbb{Z}$. Porém, o conjunto $\mathbb{Z}$ não é um grupo sob a multiplicação, pois nem todos os elementos pertencentes a esse conjunto possuem elemento inverso.*

Definição 2.1.2. *O número de elementos de um grupo é chamado de ordem de um grupo. Um grupo de ordem finita é chamado grupo finito. Caso contrário, é denominado de grupo infinito.*

Definição 2.1.3. *As operações entre elementos de um grupo finito podem ser representadas em uma tabela de informações, conhecida como Tábua de Cayley.*

Exemplo 2.1.2. *Considere $G = \{e, a\}$ um grupo sob a operação $*$, de ordem 2. Para satisfazer as propriedades de grupo, necessariamente um elemento de G tem que ser um elemento neutro, além disso, o elemento a tem que ser o seu próprio elemento inverso, caso contrário, teríamos dois elementos neutros, o que contradiria a sua unicidade. Este grupo é representado pela tábua, ilustrada na Tabela 1:*

Tabela 1 – Tábua de $G = \{e, a\}$

$*$	e	a
e	e	a
a	a	e

Fonte: Da autora.

2.1.2 Anéis

Definição 2.1.4. *Um conjunto não vazio A e um par de operações sobre A, respectivamente uma adição $(x, y) \mapsto x + y$ e uma multiplicação $(x, y) \mapsto x \cdot y$, é chamado de anel se for munido das propriedades a seguir.*
Para a adição:

a) associatividade: $a + (b + c) = (a + b) + c$, quaisquer que sejam $a, b, c \in A$;

b) comutatividade: $a + b = b + a$, quaisquer que sejam $a, b \in A$;

c) existência de elemento neutro: existe um elemento neutro $0_A \in A$ tal que, qualquer que seja $a \in A$, $a + 0_A = a = 0_A + a$;

d) existência de elemento inverso: qualquer que seja $a \in A$, existe um elemento em A, indicado por $-a$, tal que $a + (-a) = 0_A = (-a) + a$.

Para a multiplicação:

e) associatividade: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$, quaisquer que sejam $a, b, c \in A$;

f) a multiplicação é distributiva em relação à adição, à direita e à esquerda: se $a, b, c \in A$, então $a \cdot (b + c) = a \cdot b + a \cdot c$ e $(b + c) \cdot a = b \cdot a + c \cdot a$.

Quando em um anel a propriedade da comutatividade for válida para a multiplicação, dizemos que ele é um *anel comutativo*. Pelas propriedades postas anteriormente, em um anel não é necessário ter o elemento neutro da multiplicação, porém quando um anel A tiver este elemento, denotado por 1_A, dizemos que A é um *anel com unidade*. Os elementos não nulos de um anel não necessitam ter inversos multiplicativos, mas quando um anel A os possuir, estes elementos são chamados de *invertíveis* de A.

Exemplo 2.1.3. *O conjunto $\mathbb{Z}$ dos inteiros é um anel, onde $+$ e $\cdot$ são as operações de adição e multiplicação usuais dos inteiros. E temos ainda, que a operação de multiplicação é comutativa e $1 \in \mathbb{Z}$ é o elemento neutro dessa operação. Sendo assim, o conjunto dos inteiros é um anel comutativo com unidade.*

As propriedades listadas no teorema a seguir são consequências do fato de que A é um grupo comutativo sob adição.

Teorema 2.1.3. *Sejam a, b e c elementos de um anel A. Então:*

1. *Valem as leis do cancelamento para a adição, ou seja, se* $a+b=a+c$*, então* $b=c$ *e se* $b+a=c+a$*, então* $b=c$*;*

2. *O elemento neutro da adição é único;*

3. *Para todo* a*, o inverso aditivo de* a *é único;*

4. $a \cdot 0 = 0 = 0 \cdot a$*;*

5. $a \cdot (-b) = (-a) \cdot b = -(a \cdot b)$*;*

6. *Se* A *é um anel com unidade, então* 1_A *é único.*

Demonstração.

1. *Como* A *é um grupo abeliano sob a adição, segue que para todo elemento* $a \in A$ *existe um inverso* $-a$*. Então,*

$$\begin{aligned} a+b=a+c &\Rightarrow (-a)+a+b=(-a)+a+c \\ &\Rightarrow b=c. \end{aligned}$$

   Portanto, $b=c$*. O mesmo raciocínio é utilizado para mostrar que* $b+a=c+a \Rightarrow b=c$*.*

2. *Suponhamos que existem dois elementos neutros* $0_A, 0'_A \in A$*, tais que,*

$$0_A = 0_A + 0'_A = 0'_A,$$

   ou seja, $0_A = 0'_A$*.*

   Logo, existe somente um elemento neutro em A*.*

3. *Suponhamos que existem dois elementos inversos* $-a, -a' \in A$*, para um elemento* $a \in A$*. Assim,*

$$a+(-a)=0=a+(-a').$$

   Segue então pela lei do cancelamento, provada na propriedade 1, $-a=-a'$*. Portanto, existe apenas um elemento inverso aditivo para cada elemento* a*.*

4. *Para todo* $a \in A$*, temos que*

$$\begin{aligned} a \cdot 0_A + 0_A &= a \cdot 0_A \\ &= a \cdot (0_A + 0_A) \\ &= a \cdot 0_A + a \cdot 0_A. \end{aligned}$$

   Segue pela lei do cancelamento que $0_A = a \cdot 0_A$*.*

5. *Note que,*

$$a \cdot (-b) + a \cdot b = a \cdot (-b + b) = a \cdot 0_A = 0_A. \tag{2.1}$$

Logo, $a \cdot (-b) = -(a \cdot b)$*. Do mesmo modo,*

$$(-a) \cdot b + a \cdot b = (-a + a) \cdot b = 0_A \cdot b = 0_A. \tag{2.2}$$

Por isso, $(-a) \cdot b = -(a \cdot b)$*. Portanto, de (2.1) e (2.2), segue que* $a \cdot (-b) = a \cdot (-b) = -(a \cdot b)$*.*

6. *Suponhamos que existem dois elementos neutros da multiplicação* $1_A, 1'_A \in A$*, então,*

$$1_A = 1_A \cdot 1'_A = 1'_A,$$

Logo, se A *é um anel com unidade, então* 1_A *é único.*

■

Um anel $(A, +, \cdot)$ em que o conjunto A é finito chama-se *anel finito.* Se A é um anel finito, as tábuas da adição e multiplicação podem ser úteis para a visualização de algumas de suas características.

Exemplo 2.1.4. *O subconjunto dos números inteiros no qual todos os elementos possuem o mesmo resto quando divididos por* m *é chamado classe residual módulo* m *do elemento* a *de* $\mathbb{Z}$*, e é denotado por:* $[a] = \{x \in \mathbb{Z}; x \equiv a \pmod{m}\}$*. O conjunto de todas as classes residuais módulo* m *é representada por* $\mathbb{Z}_m$*.*

Os anéis da classe de resto módulo m*,* $\mathbb{Z}_m$ *(*$m > 1$*), são exemplos importantes de anéis finitos. As suas operações são definidas da seguinte forma:*

$$\overline{a} + \overline{b} = \overline{a + b} \text{ e } \overline{a} \cdot \overline{b} = \overline{a \cdot b}.$$

As operações de soma e poduto entre os elementos de $\mathbb{Z}_m$ *estão bem definidas. Pois, se* $\overline{a_1} = \overline{a_2} \in \mathbb{Z}_m$ *e* $\overline{b_1} = \overline{b_2} \in \mathbb{Z}_m$*, então:*

$$a_1 \equiv a_2 (mod \quad m) \text{ e } b_1 \equiv b_2 (mod \quad m).$$

Segue das propriedades de congruência que:

$$a_1 + b_1 \equiv a_2 + b_2 (mod \quad m) \text{ e } a_1 \cdot b_1 \equiv a_2 \cdot b_2 (mod \quad m).$$

Isto é,

$$\overline{a_1 + b_1} = \overline{a_2 + b_2} \text{ e } \overline{a_1 \cdot b_1} = \overline{a_2 \cdot b_2}.$$

Logo, $\overline{a_1} + \overline{b_1} = \overline{a_2} + \overline{b_2}$ *e* $\overline{a_1} \cdot \overline{b_1} = \overline{a_2} \cdot \overline{b_2}$*.*

Agora, considere as tábuas do anel $\mathbb{Z}_4 = \{0, 1, 2, 3\}$ *apresentadas nas Tabelas 2 e 3.*

Tabela 2 – Tábua da adição de $\mathbb{Z}_4$

$+$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Fonte: Da autora.

Tabela 3 – Tábua da multiplicação de $\mathbb{Z}_4$

$\cdot$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Fonte: Da autora.

2.1.3 Corpos

Definição 2.1.5. *Um conjunto não vazio K e um par de operações sobre K, respectivamente uma adição $(x, y) \mapsto x + y$ e uma multiplicação $(x, y) \mapsto x \cdot y$, é chamado de corpo, se verificar as seguintes propriedades:*
Para a adição:

a) associatividade: $a + (b + c) = (a + b) + c$, quaisquer que sejam $a, b, c \in K$;

b) comutatividade: $a + b = b + a$, quaisquer que sejam $a, b \in K$;

c) existência de elemento neutro: existe um elemento neutro $0 \in K$ tal que, qualquer que seja $a \in K$, $a + 0 = a = 0 + a$;

d) existência de elemento inverso: qualquer que seja $a \in K$, existe um elemento em K, indicado por $-a$, tal que $a + (-a) = 0 = (-a) + a$.

Para a multiplicação:

e) associatividade: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$, quaisquer que sejam $a, b, c \in K$;

f) comutatividade: $a \cdot b = b \cdot a$, quaisquer que sejam $a, b \in K$;

g) existência de elemento neutro: existe um elemento neutro $1 \in K$ tal que, qualquer que seja $a \in K$, $a \cdot 1 = a = 1 \cdot a$;

h) existência de elemento inverso: qualquer que seja $a \in K$, existe um elemento em K, indicado por a^{-1}, tal que $a \cdot a^{-1} = 1 = a^{-1} \cdot a$;

i) a multiplicação é distributiva em relação à adição, à direita e à esquerda: se $a, b, c \in K$, então $a \cdot (b + c) = a \cdot b + a \cdot c$ e $(b + c) \cdot a = b \cdot a + c \cdot a$.

Exemplo 2.1.5. *O conjunto $\mathbb{R}$ dos números reais é um corpo sob a adição e a multiplicação usuais.*

Exemplo 2.1.6. *O conjunto $\mathbb{Q}(\sqrt{2})$ é o conjunto formado por todos os elementos da forma $a + b\sqrt{2}$, em que $a, b \in \mathbb{Q}$. $\mathbb{Q}(\sqrt{2})$ é um corpo, com as operações de adição e multiplicação definidas da seguinte forma:*

1. $(a + b\sqrt{2}) + (c + d\sqrt{2}) = (a + c) + (b + d)\sqrt{2}$;
2. $(a + b\sqrt{2}) \cdot (c + d\sqrt{2}) = (a \cdot c + 2b \cdot d) + (a \cdot d + b \cdot c)\sqrt{2}$.

Um corpo K com um número finito de elementos é chamado um corpo finito. Esses corpos também são conhecidos como *Corpos de Galois*, em honra ao matemático francês Évariste Galois, seu descobridor. Os Corpos de Galois são uma estrutura algébrica especial, pois representam a base sobre a qual grande parte da teoria de codificação algébrica, construção de códigos e decodificação é construída.

2.1.4 Corpos de Galois

Definição 2.1.6. *Um Corpo de Galois é um corpo com um número finito de elementos representado por $GF(p^m)$, onde p é um número primo e m um inteiro positivo.*

Exemplo 2.1.7. *Considere o conjunto $\mathbb{Z}_2 = \{0, 1\}$ com a adição e multiplicação definidas nas Tabelas 4 e 5.*

Tabela 4 – Tábua da adição de $\mathbb{Z}_2$

$+$	0	1
0	0	1
1	1	0

Fonte: Da autora.

Tabela 5 – Tábua da multiplicação de $\mathbb{Z}_2$

$\cdot$	0	1
0	0	1
1	1	0

Fonte: Da autora.

Por meio das Tábuas de Cayley, pode-se verificar que o conjunto $\mathbb{Z}_2 = \{0, 1\}$ é um corpo sob a adição e multiplicação módulo 2, constituído de dois elementos, e este corpo é usualmente chamado de corpo binário, denotado por $GF(2)$.

Geralmente, podemos construir códigos com símbolos de qualquer Corpo de Galois $GF(p^m)$, porém os códigos com símbolos binários são os mais utilizados em transmissão de dados digitais e sistemas de armazenamento.

Em seguida, para a construção de embasamentos teóricos importantes, considere cálculos com polinômios cujos coeficientes estão no corpo binário. Polinômios sobre $GF(2)$ podem ser somados, multiplicados ou divididos, de forma usual. Entretanto, na aritmética binária sob os coeficientes é utilizada adição e multiplicação módulo 2, conforme as Tabelas 4 e 5, assim, nesta aritmética consideramos 2 igual a 0 ($1 + 1 = 2 = 0$), ou seja, $1 = -1$. Com isso, temos que na aritmética binária a subtração é igual a adição.

Exemplo 2.1.8. *Sejam $a(x) = 1 + x + x^3 + x^5$ e $b(x) = 1 + x^2 + x^3 + x^4 + x^7$, em que os coeficientes dos polinômios $a(x)$ e $b(x)$ são elementos de $GF(2)$. Então,*

$$\begin{aligned} a(x) + b(x) &= (1 + x + x^3 + x^5) + (1 + x^2 + x^3 + x^4 + x^7) \\ &= (1 + 1) + x + x^2 + (1 + 1)x^3 + x^4 + x^5 + x^7 \\ &= 2 + x + x^2 + 2x^3 + x^4 + x^5 + x^7 \\ &= 0 + x + x^2 + 0x^3 + x^4 + x^5 + x^7 \\ &= x + x^2 + x^4 + x^5 + x^7. \end{aligned}$$

Se na divisão de um polinômio $d(x)$ sobre $GF(2)$ de grau n por um polinômio $e(x)$ sobre $GF(2)$, de grau menor ou igual n, o resto for igual a zero dizemos que $d(x)$ é divisível por $e(x)$. E o polinômio $d(x)$ pode ser escrito como produto de $e(x)$ com o quociente. Contudo, se um polinômio sobre $GF(2)$ de grau m não é divisível por qualquer polinômio sobre $GF(2)$ de grau menor que m, este polinômio é chamado de irredutível. No que sucede, é dado um importante teorema a respeito dos polinômios irredutíveis.

Definição 2.1.7. *Um polinômio é denominado mônico se o termo de maior potência tem coeficiente 1.*

Teorema 2.1.4. *(COSTELO; LIN, 1983) Qualquer polinômio irredutível sobre $GF(2)$ de grau m divide $x^{2^m-1}+1$.*

Um outro conceito importante utilizado nos corpos de Galois é o de polinômio primitivo, definido da seguinte forma:

Definição 2.1.8. *Um polinômio $p(x)$ irredutível de grau m é primitivo se o menor inteiro positivo n para o qual $p(x)$ divide x^n+1 for $n=2^m-1$.*

Pela Definição 2.1.8 segue que todos os polinômios primitivos são irredutíveis. Pode-se verificar, por exemplo, que $p(x)=x^4+x+1$ divide $x^{15}+1$ e, pelo Teorema 2.1.4, temos que $p(x)$ é um polinômio irredutível. Além disso, $p(x)=x^4+x+1$ é um polinômio primitivo, pois não divide nenhum outro polinômio x^n+1 com $1\leq n\leq 15$.

Não é uma tarefa muito simples identificar polinômios primitivos, até porque para um dado m pode existir mais do que um polinômio irredutível de ordem m. No entanto, existem tabelas que indicam quais são os polinômios irredutíveis, associados a uma ordem específica, e destes quais são primitivos.

2.1.5 Construção de Corpos de Galois $GF(2^m)$

Será apresentado a seguir um método para construir corpos de Galois de 2^m elementos, com $m>1$, sobre o corpo binário $GF(2)$. Para isso, considere os dois elementos de $GF(2)$, 0 e 1, e um novo símbolo α. A multiplicação "$\cdot$" é definida da seguinte forma:

$$\begin{aligned} 0.\alpha^j &= \alpha^j.0=0 \\ 1.\alpha^j &= \alpha^j.1=\alpha^j \\ \vdots &= \vdots \\ \alpha^i.\alpha^j &= \alpha^j.\alpha^i=\alpha^{i+j}. \end{aligned} \tag{2.3}$$

Forma-se assim um conjunto F de elementos no qual a operação de multiplicação "$\cdot$" está definida:

$$F = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^j, \ldots\}.$$

Agora, estabelece-se uma condição sobre o elemento α, de modo que o conjunto F contenha apenas 2^m elementos e seja fechado sobre a multiplicação, definida em (2.4). Seja $p(x)$ um polinômio irredutível de grau m sobre $GF(2)$, e suponha que $p(\alpha) = 0$, ou seja, α é raiz de $p(x)$. Como qualquer polinômio irredutível de grau m divide $x^{2^m-1} + 1$, segue que:

$$x^{2^m-1} + 1 = q(x) \cdot p(x),$$

em que $q(x)$ é um polinômio qualquer sobre $GF(2)$. Substituindo x por α, obtemos

$$\alpha^{2^m-1} + 1 = q(\alpha) \cdot p(\alpha).$$

Porém, como α é raíz de $p(x)$, temos:

$$\alpha^{2^m-1} + 1 = q(\alpha) \cdot 0 \Rightarrow \alpha^{2^m-1} + 1 = 0.$$

Adicionando 1 a ambos os lados da igualdade, obtemos:

$$\alpha^{2^m-1} = 1.$$

Portanto, sob a condição de que $p(\alpha) = 0$, o conjunto F se torna finito e contém os seguintes elementos $F^* = \left\{0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^m-2}\right\}$.

Pode-se mostrar que os elementos não nulos de F^* forma um grupo de ordem 2^{m-1} em relação a operação de multiplicação definida em (2.4), onde 1 é o elemento identidade.

O inverso multiplicativo de α^i, onde $0 < i < 2^m - 1$, é dado por α^{2^m-1-i}, pois:

$$\alpha^i \cdot \alpha^{2^m-1-i} = \alpha^{2^m-1-i+i} = \alpha^{2^m-1} = 1.$$

Logo, os elementos não nulos de F^* formam um grupo comutativo de ordem $2^m - 1$ sobre a multiplicação, definida em (2.4).

Define-se a adição "+" sobre F^* do seguinte modo:

$$0 + 0 = 0, \tag{2.4}$$

e para $0 \leq i,\ j \leq 2^m - 1$,

$$0 + \alpha^i = \alpha^i + 0 = \alpha^i, \tag{2.5}$$

e

$$\begin{aligned} \alpha^i + \alpha^j &= a_{i0} + a_{i1}\alpha + a_{i2}\alpha^2 + \ldots + a_{im-1}\alpha^{m-1} + a_{j0} + a_{j1}\alpha + a_{j2}\alpha^2 + \ldots + a_{jm-1}\alpha^{m-1} \\ &= (a_{i0} + a_{j0}) + (a_{i1} + a_{j1})\alpha + \ldots + (a_{im-1} + a_{jm-1})\alpha^{m-1}. \end{aligned} \tag{2.6}$$

Assim, (2.6) deve ser a expressão polinomial para algum α^k em F^*. Com isso, pode-se verificar que F^* é um grupo comutativo sob a adição e sob a multiplicação.

Usando a representação polinomial para os elementos em F^*, averigua-se facilmente que a multiplicação sobre F^* é distributiva em relação à adição sobre F^*. Portanto, $F^* = \left\{0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^m-2}\right\}$ é um corpo de Galois de 2^m elementos.

No processo de construção de $GF(2^m)$ a partir de $GF(2)$, os elementos não nulos de $GF(2^m)$ podem ser representados por potências de uma raiz de um polinômio irredutível em $GF(2)$ de grau m, polinômios ou vetores.

Exemplo 2.1.9. *Seja $m = 3$. O polinômio $p(x) = 1 + x + x^3$ é primitivo sobre $GF(2)$. Estabelecendo $p(\alpha) = 0$, temos $\alpha^3 = 1 + \alpha$. Usando essa relação, podemos construir $GF(2^3)$. Os elementos de $GF(2^3)$ são apresentados na Tabela 6.*

Tabela 6 – Representações do corpo $GF(2^3)$

POTÊNCIA	POLINOMIAL	VETORIAL
0	0	(0,0,0)
α^0	1	(1,0,0)
α^1	α^1	(0,1,0)
α^2	α^2	(0,0,1)
α^3	$1 + \alpha$	(1,1,0)
α^4	$\alpha + \alpha^2$	(0,1,1)
α^5	$1 + \alpha + \alpha^2$	(1,1,1)
α^6	$1 + \alpha^2$	(1,0,1)

Fonte: Da autora.

2.1.6 Propriedades básicas dos Corpos de Galois $GF(2)$

Na álgebra usual, alguns polinômios com coeficientes reais possuem raízes que pertencem ao conjunto dos números complexos. Essa mesma condição também pode ocorrer para polinômios com coeficientes em $GF(2)$. Nesse caso, um polinômio com coeficientes em $GF(2)$ poderá ter raízes em uma extensão de $GF(2)$.

Os próximos teoremas indicam como encontrar todas as raízes de um polinômio com coeficientes em $GF(2)$ a partir de uma raiz conhecida.

Teorema 2.1.5. *Seja $p(x)$ um polinômio com coeficientes em $GF(2)$ e β um elemento em uma extensão do corpo $GF(2)$ que seja raiz de $p(x)$. Então, para todo $i \geq 0$, β^{2^i} também é raiz de $p(x)$.*

Demonstração. *Considere $f(x)$ um polinômio de grau n com coeficientes em $GF(2)$.*

Vamos provar, primeiramente, que $[p(x)]^{2t} = p(x^{2t})$ *é válido para* $t \geq 0$. *De fato, pois:*

$$\begin{aligned} p^2(x) &= (p_0 + p_1x + p_2x^2 + \cdots + p_nx^n)^2 \\ &= [p_0 + (p_1x + p_2x^2 + \cdots + p_nx^n)]^2 \\ &= p_0^2 + p_0(p_1x + p_2x^2 + \cdots + p_nx^n) + p_0(p_1x + p_2x^2 + \cdots + p_nx^n) + \\ &\quad (p_1x + p_2x^2 + \cdots + p_nx^n)^2 \\ &= p_0^2 + (p_1x + p_2x^2 + \cdots + p_nx^n)^2. \end{aligned}$$

Expandindo a equação anterior repetidamente, obtemos:

$$p^2(x) = p_0^2 + (p_1x)^2 + (p_2x^2)^2 + \cdots + (p_nx^n)^2.$$

Como os coeficientes p_i, $i = 0, \ldots, n$, *são 0 ou 1, implica em* $p_i^2 = p_i$ *e temos*

$$\begin{aligned} p^2(x) &= p_0 + p_1x^2 + p_2(x^2)^2 + \cdots + p_n(x^2)^n \\ &= p(x^2). \end{aligned}$$

Logo, para algum $t > 0$,

$$[p(x)]^{2t} = p(x^{2t}). \tag{2.7}$$

Agora, substituindo β *na equação (2.7), obtemos*

$$[p(\beta)]^{2t} = p(\beta^{2t}),$$

desde que $p(\beta) = 0$, $p(\beta^{2t}) = 0$. *Portanto,* β^{2t} *também é raíz de* $p(x)$. ■

O elemento β^{2^i} é denominado conjugado de β. Considere o polinômio $x^4 + x^3 + 1$. É fácil verificar que se o elemento α^7 é uma raíz desse polinômio. Então,

$$(\alpha^7)^2 = \alpha^{14}; \quad (\alpha^7)^{2^2} = \alpha^{13}; \quad (\alpha^7)^{2^3} = \alpha^{11}$$

são também raízes.

Outra propriedade dos Corpos de Galois que é importante destacar diz respeito aos *polinômios minimais de* β. Define-se polinômio minimal de β como o polinômio de menor grau para o qual β é raíz. Dessa propriedade decorrem os seguintes teoremas:

Teorema 2.1.6. *Se* $p(x)$ *for um polinômio em* $GF(2)$ *e* $\phi(x)$ *for o polinômio minimal de* β, *se* β *também for raiz de* $p(x)$, *então,* $p(x)$ *também é divisível por* $\phi(x)$.

Demonstração. *Dividindo o polinômio* $p(x)$ *pelo polinômio minimal* $\phi(x)$, *obtemos:*

$$p(x) = a(x)\phi(x) + r(x),$$

em que o grau do polinômio $r(x)$ *é menor do que o grau de* $\phi(x)$.

Por hipótese, temos que ϕ é polinômio minimal de β e β é raíz de $p(x)$, utilizando esse fato, segue que:

$$p(\beta) = \phi(\beta) = 0.$$

Consequentemente, $r(x) = 0$, pois se $r(x) \neq 0$, $r(x)$ seria um polinômio com grau menor que o grau de $\phi(x)$, que tem β como raiz. No entanto, isso não pode acontecer pois $\phi(x)$ é o polinômio minimal de β. Portanto, $r(x)$ tem que ser 0 e $\phi(x)$ divide $p(x)$.

■

Teorema 2.1.7. *Sejam $p(x)$ um polinômio irredutível sobre $GF(2)$ e β um elemento em $GF(2^m)$. Considere $\phi(x)$ polinômio minimal de β. Se $p(\beta) = 0$, então $\phi(x) = p(x)$.*

Demonstração. *Pelo Teorema 2.1.6 temos que $\phi(x)$ divide $p(x)$. Sendo $\phi(x) \neq 1$, $p(x)$ um polinômio irredutível e $p(\beta) = 0$, segue que:*

$$\phi(x) = p(x).$$

■

O Teorema 2.1.7 diz que se um polinômio irredutível tem β como raíz, então ele é o polinômio minimal de β.

Teorema 2.1.8. *Seja $\phi(x)$ o polinômio minimal de um β em $GF(2^m)$. Seja e o menor inteiro para o qual $\beta^{2^e} = \beta$, então:*

$$\phi(x) = \prod_{i=0}^{s-1}(x + \beta^{2^i}).$$

Demonstração. *Considere:*

$$[\phi(x)]^2 = \left[\prod_{i=0}^{e-1}(x + \beta^{2^i})\right]^2 = \prod_{i=0}^{e-1}(x + \beta^{2^i})^2.$$

Desde que, $(x+\beta^{2^i})^2 = x^2 + 2\beta^{2i}x + \beta^{2i+1} = x^2 + \beta^{2i}x + \beta^{2i}x + \beta^{2i+1} = x^2 + x(\beta^{2i} + \beta^{2i}) + \beta^{2i+1} = x^2 + \beta^{2i+1}$, assim,

$$[\phi(x)]^2 = \prod_{i=0}^{e-1}(x^2 + \beta^{2^{i+1}}) = \prod_{i=1}^{e}(x^2 + \beta^{2^i}) = \left[\prod_{i=1}^{e-1}(x^2 + \beta^{2^i})\right](x^2 + \beta^{2^e}).$$

Sendo β um elemento de $GF(2^m)$, tal que $\beta^{2^e} = \beta$, então

$$[\phi(x)]^2 = [\phi(x)^2]. \tag{2.8}$$

Seja $\phi(x) = \phi_0 + \phi_1 x + \ldots + \phi_e x^e$, *onde* $\phi_e = 1$. *Expandindo*

$$[\phi(x)]^2 = (\phi_0 + \phi_1 x + \ldots + \phi_e x^e)^2 = \sum_{i=0}^{e} \phi_i^2 x^{2i}. \tag{2.9}$$

De (2.8) e (2.9), obtemos:

$$\sum_{i=0}^{e} \phi_i x^{2i} = \sum_{i=0}^{e} \phi_i^2 x^{2i}.$$

Então, para $0 \leq i \leq e$, *temos:*

$$\phi_i = \phi_i^2.$$

Sendo isso verdade apenas quando $\phi_i = 0$ *ou 1. Portanto, os coeficientes de* $\phi_i \in GF(2)$.

Agora suponha que $\phi(x)$ *não seja irredutível sobre* $GF(2)$ *e* $\phi(x) = a(x)b(x)$. *Seja* $\phi(\beta) = 0$, *então* $a(\beta) = 0$ *ou* $b(\beta) = 0$. *Se* $a(\beta) = 0$, $a(x)$ *tem como raízes* $\beta, \beta^e, \cdots, \beta^{e-1}$, *então* $a(x)$ *tem grau* e *e* $a(x) = \phi(x)$. *Da mesma forma podemos ter* $b(x) = \phi(x)$. *Logo,* $\phi(x)$ *é irredutível.*

■

Exemplo 2.1.10. *Considere* $p(x) = x^4 + x + 1$, *pode-se notar que* α^3 *é uma raíz de* $p(x)$. *Se* $\beta = \alpha^3$, *os conjugados são:*

$$\beta^2 = \alpha^6; \quad \beta^{2^2} = \alpha^{12}; \quad \beta^{2^3} = \alpha^9.$$

O polinômio minimal de $\beta = \alpha^3$ *então é dado por:*

$$\begin{aligned} \phi(x) &= (x + \alpha^3)(x + \alpha^6)(x + \alpha^{12})(x + \alpha^9) \\ &= [x^2 + (\alpha^3 + \alpha^6)x + x^9][x^2 + (\alpha^{12} + \alpha^9)x + \alpha^{21}] \\ &= x^4 + x^3 + x^2 + x + 1. \end{aligned}$$

2.2 CÓDIGOS CORRETORES DE ERROS

Um canal de comunicação pode manifestar uma série de perturbações que dificultam o entendimento dos dados enviados, essas perturbações podem ser ruídos, interferências, etc (OLIVEIRA, 2012). Podemos considerar que um sistema de comunicação é um conjunto de mecanismos que possibilita a transmissão da informação de um transmissor para um determinado receptor, através de um canal de comunicação. É desejável que essa transmissão seja efetuada com alta confiabilidade, permitindo que a mensagem recebida seja igual a original, portanto, existe uma grande preocupação em relação ao controle de perturbações.

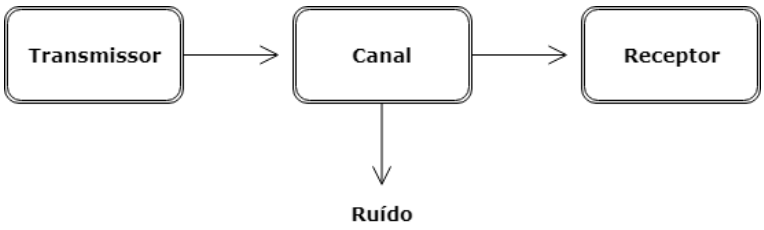
Um sistema de comunicação possui os seguintes elementos, sendo representado através de um diagrama de blocos, apresentado na Figura 1. Cada um dos blocos constituintes desse sistema são definidos da seguinte forma:

a) **transmissor:** responsável por gerar a informação a ser transmitida através de mensagem.

b) **canal:** é o meio físico que conecta o transmissor ao receptor, através do qual a mensagem será transmitida. No entanto, o sinal enviado sofre ação de ruídos, contribuindo para que a mensagem chegue ao seu destino com erros.

c) **receptor:** é o usuário que recebe a mensagem transmitida pelo canal.

Figura 1 – Diagrama de blocos do sistema de comunicação



Fonte: Da autora.

Quando se deseja transmitir ou armazenar dados, a teoria dos códigos é utilizada. O estudo dessa teoria foi iniciado no final da década de 40 com os trabalhos de Golay, Hamming e Shannon (TRANCOSO, 1995). No entanto, segundo Rocha (2010), a grande descoberta da época surgiu principalmente de Shannon, que demonstrou que se a taxa da mensagem enviada for menor que a capacidade do canal utilizado, é possível garantir através de técnicas adequadas de codificação e decodificação que a mensagem destinada chegará com uma quantidade de erros pequena ao receptor. Desde então, há um grande esforço para desenvolver códigos eficientes com controle de erros, em sistemas de comunicação passíveis de ruídos.

Atualmente, os códigos corretores de erros podem ser divididos em dois grupos: códigos de blocos e códigos convolucionais, sendo que cada um dos grupos possui diversos códigos, para as mais diferentes aplicações.

Os códigos convolucionais possuem caráter probabilístico, pois a princípio, alguns matemáticos procuraram estimar a probabilidade de erros das "melhores" famílias de códigos de blocos, além do mais, tinham o objetivo de compreender a codificação e a decodificação de um ponto de vista probabilístico, levando-se a noção de decodificação sequencial. A decodificação sequencial exigiu a criação de uma nova classe de códigos sem blocos, com comprimento indefinido, representáveis por uma árvore em que a decodificação é feita percorrendo toda a extensão dessa árvore. Os códigos convolucionais costumam ser mais comuns do que os códigos de blocos por serem melhores estruturados e mais

fáceis de serem implementados, porém são códigos relativamente mais difíceis de serem decodificados.

Já os códigos de bloco são denominados dessa forma, pois o processo de codificação é realizado sobre blocos de bits. São códigos que operam sobre fortes estruturas algébricas, conhecidas como corpos algébricos e isso permite encontrar várias técnicas de decodificação (ROCHA, 2010).

Nesta seção serão introduzidos os conceitos básicos de códigos de bloco, códigos de bloco lineares, códigos cíclicos e também de uma importante família de códigos, os códigos BCH. Esses conceitos podem ser encontrados em Barbosa (2000), Costelo e Lin (1983), Faria (2011), Gonzalez (2017), Interlando (1994), Rocha (2010) e Trancoso (1995).

2.2.1 Códigos de bloco

Definição 2.2.1. *Um código de bloco $\mathcal{C}$ de comprimento n sobre um alfabeto $\mathcal{A}$ é qualquer subconjunto do conjunto A^n das sequências $c = \{c_i | 1 \leq i \leq n\}$.*

Um código de bloco é caracterizado por três parâmetros: comprimento, dimensão e distância mínima. Estes dois últimos são definidos a seguir.

Definição 2.2.2. *A dimensão de um código $\mathcal{C}$ é dada por:*

$$k = log_{|\mathcal{A}|}|\mathcal{C}|,$$

em que $|.|$ é a cardinalidade do conjunto.

Definição 2.2.3. *A distância de Hamming, denotada por $d(u, v)$, é definida como o número de elementos em que dois vetores u e v se diferem.*

Definição 2.2.4. *A distância mínima d_{min} de um código de bloco $\mathcal{C}$ é a menor distância de Hamming entre dois vetores distintos quaisquer desse código, ou seja,*

$$d_{min} = \{d(u, v) : u, v \in \mathcal{C}, u \neq v\}.$$

Assim, representamos um código de bloco pela terna ordenada (n, k, d_{min}). É importante destacar outro importante parâmetro dos códigos de blocos que é a capacidade de correção de erro, sendo definida como o número máximo de erros que podem ser corrigidos por palavra-código, e é dada por:

$$t = \lfloor (d_{min} - 1)/2 \rfloor,$$

isto é, t é o maior inteiro não superior a $(d_{min} - 1)/2$. A capacidade de correção de erro, t, está relacionada a distância mínima do código da seguinte forma:

$$d_{min} \leq 2t + 1,$$

consequentemente, quanto maior a capacidade de correção de erros de um código, maior a sua distância mínima.

2.2.2 Códigos de blocos lineares

Nessa subseção, serão abordados os conceitos básicos de uma subclasse de códigos de bloco, conhecida como códigos de blocos lineares. No entanto, como a maioria dos computadores e sistemas de comunicação utilizam a codificação binária, iremos tratar apenas de códigos de bloco com símbolos no corpo binário $GF(2)$.

Definição 2.2.5. *Um código de bloco de comprimento n e 2^k palavras-código é um código linear se e só se as suas 2^k palavras-código formam um subespaço vetorial de dimensão k em relação ao espaço formado pelas 2^n n-uplas possíveis em $GF(2)$.*

Pela Definição 2.2.5 um código de bloco é linear se a soma módulo 2 de quaisquer duas palavras do código resultar em uma palavra-código também. Uma vez que a soma de duas palavras-código é uma palavra-código, temos que uma palavra-código v pode ser escrita como combinação linear das outras palavras-código.

Assim, se o código em questão tiver k palavras-código independentes, $g_0, g_1, \ldots, g_{k-1}$, v pode ser escrito da seguinte maneira:

$$v = u_0 g_0 + u_1 g_1 + \cdots + u_{k-1} g_{k-1},$$

com u_i variando entre 0 ou 1, para $0 \leq i \leq k-1$.

Se definirmos com essas k palavras-código independentes uma matriz de dimensão $k \times n$, obtemos:

$$G = \begin{bmatrix} g_{00} & g_{01} & g_{02} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & g_{12} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdots & g_{k-1,n-1} \end{bmatrix}.$$

As linhas da matriz G geram o código linear (n, k). Por isso, a matriz G é denominada de matriz geradora.

Exemplo 2.2.1. *O código linear (7, 4), listado na Tabela 7, tem a seguinte matriz geradora:*

$$G = \begin{bmatrix} 1101000 \\ 0110100 \\ 1110010 \\ 1010001 \end{bmatrix}.$$

Tabela 7 – Código de bloco linear com $k = 4$ e $n = 7$.

MENSAGEM	PALAVRA CÓDIGO
(0000)	(0000000)
(1000)	(1101000)
(0100)	(0110100)
(1100)	(1011100)
(0010)	(1110010)
(1010)	(0011010)
(0110)	(1000100)
(1110)	(0101110)
(0001)	(1010001)
(1001)	(0111001)
(0101)	(1100101)
(1101)	(0001101)
(0011)	(0100011)
(1011)	(1001011)
(0111)	(0010111)
(1111)	(1111111)

Fonte: Da autora.

Se $u = (1101)$ *for a mensagem a ser codificada, a palavra código correspondente será:*

$$\begin{aligned} v &= 1.(1101000) + 1.(0110100) + 0.(1110010) + 1.(1010001) \\ &= (1101000) + (0110100) + (1010001) \\ &= (0001101), \end{aligned}$$

onde as somas são realizadas módulo 2.

Associada à matriz geradora G, existe uma outra matriz de dimensão $(n-k) \times n$, chamada matriz de verificação de paridade, que é denotada por H, com $n-k$ linhas linearmente independentes, de forma que as linhas da matriz G sejam ortogonais às linhas da matriz H. Ou seja, $G \cdot H^T = 0$, em que H^T é a matriz transposta de H. Com isso, podemos descrever um código linear (n, k) gerado por G de uma forma diferente, pois uma palavra-código v do código gerado por G é palavra código se somente se $v \cdot H^T = 0$, sendo 0 o vetor nulo.

Exemplo 2.2.2. *A matriz de verificação de paridade* H*, associada a matriz* G *do Exemplo 2.2.1 é*

$$H = \begin{bmatrix} 1001011 \\ 0101110 \\ 0010111 \end{bmatrix}.$$

E podemos verificar que $v = (0001101)$ *pertence ao código, pois*

$$v \cdot H^T = (0001101) \begin{bmatrix} 100 \\ 010 \\ 001 \\ 110 \\ 011 \\ 111 \\ 101 \end{bmatrix} = (000).$$

Agora, considere um código linear (n, k) com matriz geradora G e matriz de verificação de paridade H. Suponha que uma palavra-código $v = (v_0, v_1, \ldots, v_{n-1})$ seja codificada e transmitida por um canal ruidoso. Esse canal ruidoso atribuirá a essa palavra código um certo erro, denotado por $e = (e_0, e_1, \ldots, e_{n-1})$, e fará com que a mensagem recebida pelo transmissor, r, seja composta pela palavra-código v e pelo erro e, ou seja,

$$r = v + e.$$

Depois disso, é necessário verificar se a palavra r pertence ou não ao código. Para essa verificação, determina-se a síndrome s da palavra recebida r, que é dada por:

$$s = v \cdot H^T = \{s_0, s_1, \ldots, s_{n-k-1}\}.$$

Tem-se que $s = 0$ se e somente se r é uma palavra do código e $s \neq 0$ se e somente se r não é uma palavra do código.

2.2.3 Códigos cíclicos

Os códigos cíclicos formam uma importante subclasse dos códigos de bloco lineares. Estes códigos se tornam interessantes devido ao fato de terem uma forte estrutura algébrica, o que facilita o processo de decodificação.

Definição 2.2.6. *Um código linear* $\mathcal{C}(n, k)$ *é um código cíclico se qualquer deslocamento cíclico de uma palavra-código de* $\mathcal{C}$ *resulta em uma outra palavra-código do código* $\mathcal{C}$*.*

Como citado anteriormente, os códigos cíclicos possuem muitas propriedades algébricas que facilitam seus processos de codificação e decodificação. Para desenvolver tais propriedades dos códigos cíclicos, considere as componentes de uma palavra-código $v = (v_0, v_1, \ldots, v_{n-1})$ como os coeficientes de um polinômio:

$$v(x) = v_0 + v_1 x + v_2 x^2 + \cdots + v_{n-1} x^{n-1}.$$

Com isso, cada vetor ou palavra-código corresponde a um polinômio de grau menor ou igual a $n - 1$.

Os teoremas a seguir caracterizam propriedades importantes dos códigos cíclicos, além de serem relevantes para a compreensão de que forma esses códigos são gerados.

Teorema 2.2.1. *O polinômio não nulo de grau mínimo de um código cíclico $\mathcal{C}$ é único.*

Demonstração. *Seja $g(x) = g_0+g_1x+\cdots+g_{r-1}x^{r-1}+x^r$ um polinômio não nulo de grau mínimo de um código cíclico $\mathcal{C}$. Suponhamos que $g(x)$ não é único. Então existe outro polinômio de grau r do código $\mathcal{C}$, denotado por $g'(x) = g'_0+g'_1x+\cdots+g'_{r-1}x^{r-1}+x^r$. Como $\mathcal{C}$, em particular, é um código linear, temos que a soma de dois polinômios do código $\mathcal{C}$ resulta em outro polinômio do código $\mathcal{C}$, ou seja,*

$$g(x) + g'(x) = (g_0 + g'_0) + (g_1 + g'_1)x + \cdots + (g_{r-1} + g'_{r-1})x^{r-1}$$

é também um polinômio de grau $r-1$.

Se $g(x)+g'(x) \neq 0$, então $g(x)+g'(x)$ é um polinômio não nulo de grau mínimo $r-1$, porém isso não é verdade. Então $g(x) + g'(x) = 0$, isso implica que $g(x) = g'(x)$. Logo, $g(x)$ é único. ■

Teorema 2.2.2. *Seja $g(x) = g_0 + g_1x + \cdots + g_{r-1}x^{r-1} + x^r$ um polinômio diferente de zero de grau mínimo em um código cíclico $\mathcal{C}$. Então o termo constante g_0 deve ser igual a 1.*

Demonstração. *Suponhamos que $g_0 = 0$. Então,*

$$\begin{aligned} g(x) &= g_1x + g_2x^2 + \cdots + g_{r-1}x^{r-1} + x^r \\ &= x(g_1 + g_2x + \cdots + g_{r-1}x^{r-2} + x^{r-1}). \end{aligned}$$

Se deslocarmos ciclicamente o polinômio $g(x)$ $n-1$ vezes a direita, obteremos um polinômio de código diferente de zero, $g_1+g_2x+\cdots+g_{r-1}x^{r-2}+x^{r-1}$, que possui um grau menor que r. Isso é contraditório com a suposição de que $g(x)$ é o polinômio de código diferente de zero com grau mínimo. Então, $g_0 = 1$. ■

Desse modo, em um código cíclico existirá apenas um polinômio não nulo, com grau menor que todos os outros. Temos ainda, que esse polinômio possui necessariamente o coeficiente do termo constante igual a 1. Se $g(x)$ for esse polinômio, então será da forma (considerando r como grau):

$$g(x) = 1 + g_1x + g_2x^2 + \ldots + g_{r-1}x^{r-1} + x^r.$$

Teorema 2.2.3. *Seja $g(x) = 1 + g_1x + g_2x^2 + \ldots + g_{r-1}x^{r-1} + x^r$ o polinômio de menor grau, não nulo, de um código cíclico $\mathcal{C}(n,k)$. Um polinômio binário de grau menor ou igual a $n-1$ é um polinômio do código se e somente se for um múltiplo de $g(x)$.*

Demonstração. *Seja $v(x)$ um polinômio binário de grau $n-1$ ou menor. Suponha que $v(x)$ seja um múltiplo de $g(x)$. Então*

$$\begin{aligned} v(x) &= (a_0 + a_1x + \cdots + a_{n-r-1}x^{n-r-1})g(x) \\ &= a_0g(x) + a_1g(x)x + \cdots + a_{n-r-1}x^{n-r-1}g(x). \end{aligned}$$

Como $v(x)$ é uma combinação linear dos polinômios do código $\mathcal{C}$, $g(x), \ldots, x^{n-r-1}g(x)$ também são polinômios do código $\mathcal{C}$. Com isso, temos que se um polinômio de grau $n-1$ ou menor for um múltiplo de $g(x)$, ele também é um polinômio do código.

Agora, seja $v(x)$ um polinômio do código C. Dividindo $v(x)$ por $g(x)$, obtemos:

$$v(x) = a(x)g(x) + b(x),$$

em que $b(x)$ é igual a 0 ou possui grau menor que o grau do polinômio $g(x)$. Rearranjando a equação anterior, temos:

$$b(x) = v(x) + a(x)g(x).$$

Logo, segue que $a(x)g(x)$ também é um polinômio do código. Desde que $v(x)$ e $a(x)g(x)$ sejam polinômios do código, então $b(x)$ também é. Se $b(x) \neq 0$, então $b(x)$ é um polinômio do código diferente de 0, cujo grau é menor do que o grau de $g(x)$. Isso contradiz a hipótese de que $g(x)$ é o polinômio diferente de 0 de grau mínimo. Assim, $b(x) = 0$. Portanto, provamos que um polinômio do código é múltiplo de $g(x)$. ■

Isto é, se $v(x)$ é um polinômio do código, ele pode ser escrito na forma:

$$v(x) = a(x)g(x).$$

Logo, um código cíclico (n, k) pode ser identificado pelo seu polinômio $g(x)$. Este polinômio $g(x)$ é denominado de *polinômio gerador do código.*

Teorema 2.2.4. *(INTERLANDO, 1994) Em um código cíclico (n, k), existe um e apenas um polinômio de código de grau $n-k$,*

$$g(x) = 1 + g_1x + g_2x^2 + \ldots + g_{n-k-1}x^{n-k-1} + x^{n-k}.$$

Todo polinômio do código cíclico (n, k) é um múltiplo de $g(x)$ e todo polinômio binário de grau $n-1$ ou menos que é múltiplo de $g(x)$ é um polinômio de código cíclico (n, k).

Teorema 2.2.5. *O polinômio gerador g(x) de um código cíclico (n, k) é um divisor de $x^n + 1$.*

Demonstração. *Considere o polinômio gerador $g(x)$ de um código cíclico (n, k). Agora, multiplique $g(x)$ por x^k. Isso resulta no polinômio $x^k g(x)$ de grau n. Dividindo $x^n + 1$ por x^k, obtemos*

$$x^k g(x) = (x^n + 1) + g^k(x), \tag{2.10}$$

em que $g^k(x)$ é o resto da divisão. Segue que $g^k(x)$ é o polinômio do código cíclico (n, k) obtido deslocando $g(x)$ para a direita k vezes. Consequentemente, $g^k(x)$ é um múltiplo de $g(x)$, ou seja,

$$g^k(x) = a(x)g(x). \tag{2.11}$$

Usando (2.10) para reescrever (2.11), temos

$$\begin{aligned} x^k g(x) &= (x^n + 1) + a(x)g(x) \\ x^n + 1 &= x^k g(x) + a(x)g(x) \\ x^n + 1 &= (x^k + a(x))g(x). \end{aligned}$$

Portanto, $g(x)$ é um divisor de $x^n + 1$. ■

Teorema 2.2.6. *Se um polinômio $g(x)$ de grau $n-k$ for fator de x^n+1, então $g(x)$ gera um código cíclico (n, k).*

Demonstração. *Considere os k polinômios $g(x), xg(x), \ldots, x^{k-1}g(x)$, os quais possuem grau $n-1$ ou menor. Uma combinação linear dos k polinômios*

$$\begin{aligned} v(x) &= a_0 g(x) + a_1 x g(x) + \cdots + a_{k-1} x^{k-1} g(x) \\ &= (a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}) g(x) \end{aligned}$$

também é um polinômio de grau $n-1$ ou menor e é múltiplo de $g(x)$. Assim, temos um total de 2^k e eles formam um código linear (n, k).

Seja $v(x) = v_0 + v_1 x + \cdots + v_{n-1} x^{n-1}$ um polinômio pertencente a este código. Multiplicando $v(x)$ por x, obtemos:

$$\begin{aligned} xv(x) &= v_0 x + v_1 x^2 + \cdots + v_{n-2} x^{n-1} + v_{n-1} x^n \\ &= v_{n-1}(x^n + 1) + (v_{n-1} + v_0 x + \cdots + v_{n-2} x^{n-1}) \\ &= v_{n-1}(x^n + 1) + v^1(x), \end{aligned}$$

em que $v^1(x)$ é um deslocamento cíclico de $v(x)$. Se $xv(x)$ e $x^n + 1$ são divisíveis por $g(x)$, então $v^1(x)$ também é divisível por $g(x)$. Como $v^1(x)$ é um múltiplo de $g(x)$, logo ele é uma combinação linear de $g(x), xg(x), \ldots, x^{k-1}g(x)$. Então, $v^1(x)$ é um polinômio do código. Pela definição de código cíclico segue que o código linear gerado por $g(x), xg(x), \ldots, x^{k-1}g(x)$ é um código cíclico (n, k). ■

Ou seja, um polinômio $g(x)$ é um polinômio gerador do código cíclico (n, k) se ele for um fator $x^n + 1$. A matriz geradora para um código cíclico gerado pelo polinômio gerador $g(x) = 1 + g_1x + g_2x^2 + \ldots + g_{r-1}x^{r-1} + x^r$ pode ser obtida realizando o deslocamento dos termos do polinômio $g(x)$ da esquerda para a direita, obtendo-se a matriz geradora de $\mathcal{C}$ de dimensão $k \times n$:

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & \ldots & g_{n-k-1} & g_{n-k} & 0 & 0 & \ldots & 0 \\ 0 & g_0 & g_1 & g_2 & \ldots & g_{n-k-1} & g_{n-k} & 0 & \ldots & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & \ldots & g_{n-k-1} & g_{n-k} & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & \ldots & g_0 & g_1 & \ldots & g_{n-k-1} & g_{n-k} \end{bmatrix}.$$

Exemplo 2.2.3. *O código cíclico (7,4) definido a partir do polinômio gerador* $g(x) = 1 + x + x^3$*, tem a seguinte matriz geradora:*

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Pelo Teorema 2.2.6 temos que o polinômio gerador $g(x)$ é um fator de $x^n + 1$, desse modo pode-se escrevê-lo da seguinte forma:

$$x^n + 1 = g(x)h(x),$$

onde $h(x)$ é de grau k e tem a seguinte forma:

$$h(x) = h_0 + h_1x + \ldots + h_kx^k,$$

com $h_0 = h_k = 1$. A partir do polinômio $h(x)$ pode-se obter a matriz de verificação de paridade H, com dimensão $(n - k) \times n$, em que qualquer palavra-código do código $\mathcal{C}$ é ortogonal a qualquer linha da matriz H. Essa matriz possui a seguinte forma:

$$H = \begin{bmatrix} h_k & h_{k-1} & h_{k-2} & \ldots & h_0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & h_k & h_{k-1} & h_{k-2} & \ldots & h_0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & h_k & h_{k-1} & h_{k-2} & \ldots & h_0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & \ldots & h_k & h_{k-1} & \ldots & h_1 & h_0 \end{bmatrix}.$$

Vale destacar que a matriz de verificação de paridade H, é determinada a partir do polinômio gerador $g(x)$, pois essa matriz é obtida com base no polinômio de verificação de paridade $h(x) = (x^n - 1)/g(x)$.

Exemplo 2.2.4. *Considere o código cíclico (7,4) com polinômio gerador* $g(x) = 1+x+x^3$. *O polinômio de verificação de paridade é:*

$$\frac{x^7+1}{1+x+x^3} = 1 + x + x^2 + x^4.$$

2.2.4 Códigos cíclicos sobre anéis

Nesta subseção, apresentamos definições e teoremas relacionados a códigos cíclicos sobre $\mathbb{Z}_q$ ($q \geq 4 \in \mathbb{Z}$), onde q é uma potência de um primo, $q = p^k$, $k \geq 2$. Em particular, os resultados aqui enunciados também são válidos para $p = 2$. Os conceitos apresentados servirão de base para a construção de códigos BCH.

Definição 2.2.7. *Seja R um anel. Um R-módulo consiste de um grupo abeliano G e uma operação de multiplicação de cada elemento de G por cada elemento de R pela esquerda, tal que para todo $\alpha, \beta \in G$ e $r, s \in R$, as seguintes condições são satisfeitas:*

a) $(r\alpha) \in G$;
b) $r(\alpha + \beta) = r\alpha + r\beta$;
c) $(r + s)\alpha = r\alpha + s\alpha$;
d) $(rs)\alpha = r(s\alpha)$.

Um R-módulo assemelha-se com um espaço vetorial exceto que os escalares precisam somente formar um anel. Temos ainda que, se R é um anel, nem todos os R - módulos possuem uma base. Mais especificadamente, podem ter vetores gerando um R - módulo e que, entretanto, não são independentes. Aqueles R - módulos que são gerados por um conjunto de vetores linearmente independentes são chamados de módulos livres.

Definição 2.2.8. *Um código linear (n, k) sobre $\mathbb{Z}_q$ é definido como um módulo livre de dimensão k no espaço de todas as $n-uplas$ de $\mathbb{Z}_q$.*

Definição 2.2.9. *Um código linear $\mathcal{C}$ com parâmetros (n, k) sobre $\mathbb{Z}_q$ é cíclico se, para $v = (v_0, v_1, v_2, \ldots, v_{n-1}) \in \mathcal{C}$ todo deslocamento cíclico $v^{(1)} = (v_{n-1}, v_0, v_1, \ldots, v_{n-2}) \in \mathcal{C}$, com $v^i \in \mathbb{Z}_q$, $0 \leq i \leq n-1$.*

Os códigos cíclicos são geralmente representados na forma polinomial. Dessa forma, considere a palavra código $v = (v_0, v_1, v_2, \ldots, v_{n-1}) \in \mathcal{C}$, representada pelo polinômio

$$v(x) = v_0 + v_1x + v_2x^2 + \cdots + v_{n-1}x^{n-1}.$$

Se fizermos o produto $x.v(x) (\text{mod } x^n - 1)$, o resultado será:

$$v^{(1)}(x) = v_{n-1} + v_0x + v_1x^2 + \cdots + v_{n-2}x^{n-1},$$

que corresponde à palavra código:

$$v^{(1)} = (v_{n-1}, v_0, v_1, \ldots, v_{n-2}) .$$

Portanto, $v^{(1)}(x)$ é obtido por meio do produto $x.\, v(x)$ no anel $R_n = \mathbb{Z}_q[x]/ < x^n - 1 >$ de comprimento n, onde $< x^n - 1 >$ representa o ideal gerado por $x^n - 1$.

Teorema 2.2.7. *Um conjunto S de elementos em R_n é um código cíclico se, e somente se, S é um ideal em R_n.*

Demonstração. *Seja S um código cíclico. Se $v_1(x)$ e $v_2(x)$ são palavras código do código S, a soma entre eles também é. Pela definição de código cíclico, se $v(x)$ pertence a S, então $xv(x)$ também pertence a este código. Logo se $w(x) = w_0 + w_1 x + \cdots + w_{n-1}x^{n-1} \in R_n$ e $v(x) \in S$ então $w(x)v(x) = w_0 v(x) + w_1 v(x)x + \cdots + w_{n-1}v(x)x^{n-1} \in S$, pois cada um dos termos isoladamente pertencem ao código S, caracterizando S como sendo um ideal de R_n. Assim provamos a primeira parte do teorema.*

Agora, se S é um ideal em R_n, então a soma de dois elementos em S é um elemento em S e se $v(x) \in S$ então $xv(x) \in S$. Assim, conclui-se que S é um código cíclico. ■

Teorema 2.2.8. *Seja C um ideal em R_n, isto é, um código cíclico de comprimento n. Se o coeficiente dominante do polinômio gerador, $g(x)$, é um elemento inversível, então $g(x)$ divide $x^n - 1$.*

Demonstração. *Suponha que $g(x)$ seja o polinômio de menor grau de C. Então existem $a(x)$ e $r(x)$, tais que:*

$$x^n - 1 = g(x)a(x) + r(x),$$

em que o grau do polinômio $r(x)$ é menor do que o grau do polinômio $g(x)$. Disto segue que:

$$-g(x)a(x) = r(x) - (x^n - 1).$$

Portanto, $r(x)$ está em $< g(x) >$ e tem grau menor que $g(x)$. Isto é uma contradição a menos que $r(x) = 0$. Isso implica que $g(x)$ divide $x^n - 1$. ■

O Teorema 2.2.8 fornece um método de construção de códigos cíclicos sobre anéis de inteiros residuais, através da fatoração do polinômio $x^n - 1$ sobre o anel de interesse.

2.2.5 Códigos BCH sobre anéis

Os códigos BCH (Bose, Chaudhuri e Hocquenghen) são uma extensa família de códigos cíclicos com uma alta capacidade de correção de erros. Esses códigos são uma extensão dos códigos de Hamming para correção de erros múltiplos. A construção destes códigos utiliza a estrutura algébrica dos corpos finitos, e por ser uma classe dos códigos cíclicos

permite uma representação em termos de polinômios, o que simplifica os processos de codificação e decodificação dos códigos BCH.

Define-se um código BCH de comprimento n e distância de projeto δ, como sendo um código cíclico cujo polinômio gerador tem como raízes:

$$\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+\delta-2},$$

onde $\alpha \in GF(2^m)$ é uma raiz primitiva de $x^n - 1$, b é um inteiro não negativo e m é tal que $n/2^m - 1$. Quando $n = 2^m - 1$, temos um código BCH primitivo.

Nesta Subseção iremos descrever a construção de códigos BCH sobre anéis da forma $\mathbb{Z}_q$ ($q \geq 4 \in \mathbb{Z}$), $q = 2^k$, $k \geq 2$, a qual se assemelha a construção de códigos BCH sobre corpos. A principal diferença na construção desses dois códigos está no fato de que as raízes do polinômio gerador dos códigos BCH sobre anéis $\mathbb{Z}_q$, encontram-se na extensão do anel $\mathbb{Z}_q$.

Vamos assumir que a ordem do anel e o comprimento do código sejam relativamente primos, assim garantimos que $x^n - 1$ não apresentará fatores quadráticos. Seja $\mathbb{Z}_q[x]$ o anel de polinômios na variável x sobre $\mathbb{Z}_q$ e $p(x)$ um polinômio primitivo de grau r. Representamos por $GR(p^k, r)$ o quociente $\mathbb{Z}_q[x]$ pelo ideal gerado por $p(x)$, ou seja,

$$GR(p^k, r) \cong \frac{\mathbb{Z}_q[x]}{< p(x) >}.$$

Assim, o anel R consiste de todos os polinômios de grau $r - 1$ cujas operações de adição e multiplicação são realizadas módulo $p(x)$.

Considere, agora, o grupo dos elementos inversíveis de R, R^*. Como R^* é um grupo abeliano multiplicativo, ele pode ser expresso como produto de grupos cíclicos. Uma vez identificado este grupo, o problema da construção desses códigos se reduz a escolher certos elementos do mesmo para serem raízes do polinômio gerador, $g(x)$, o qual será um divisor de $x^n - 1$.

Os teoremas enunciados a seguir auxiliarão na construção de G_n, o subgrupo cíclico de R^*. Os teoremas que seguem são expressos para um p primo, em particular valem para $p = 2$.

Teorema 2.2.9. *(INTERLANDO, 1994) R^* tem um e somente um subgrupo cíclico de ordem relativamente prima a p. Este subgrupo cíclico tem ordem $p^r - 1$.*

Teorema 2.2.10. *(INTERLANDO, 1994) Suponha que f gere um subgrupo de ordem n em R^*, onde $(n, p) = 1$. Então, o polinômio $x^n - 1$ pode ser fatorado como $x^n - 1 = (x - f)(x - f^2) \ldots (x - f^n)$ se, e somente se, a redução de f módulo p denotado por $R_p(f)$ tem ordem n em um grupo multiplicativo de $GF(p^r)$.*

Teorema 2.2.11. *(INTERLANDO, 1994) Suponha que $\bar{f} = R_p(f)$, em que $\bar{f}$ representa a redução de f módulo 2, gere um subgrupo cíclico de ordem n em um grupo multiplicativo*

de $GF(p^r)$. Então, f gera um subgrupo cíclico de ordem n.d em R^*, onde d é um inteiro maior ou igual a 1 e f^d gera o subgrupo cíclico G_n em R^*.

O Teorema 2.2.11 é útil na determinação do polinômio gerador sobre G_n. O polinômio minimal de β^i sobre R^*, onde β é um primitivo em G_n terá como raízes todos os elementos distintos na sequência $\beta^i, (\beta^i)^p, (\beta^i)^{p2}, \ldots, (\beta^i)^{pr-1}$. Assim, um código BCH de comprimento n sobre $\mathbb{Z}_q[x]$ pode ser especificado em termos das raízes do polinômio gerador sobre G_n. Seja β um elemento primitivo de G_n. Se $\beta^{e^1}, \beta^{e^2}, \ldots, \beta^{e^j}$ são raízes de $g(x)$, então podemos gerar um código BCH sobre $\mathbb{Z}_q[x]$ se $g(x)$ for:

$$g(x) = mmc(M_{e_1}(x), M_{e_2}(x), \ldots, M_{e_j}(x)),$$

onde $M_{e_i}(x)$ é o polinômio minimal de β^{e^i}.

Exemplo 2.2.5. *Considere um código BCH sobre o anel $\mathbb{Z}_8$. Assim, temos $p = 2$ e $k = 3$. Seja $\phi(x) = x^4 + x + 1$ um polinômio irredutível sobre $\mathbb{Z}_8$. O corpo $k = GF(2^4)$ é formado por 16 elementos.*

O anel R (extensão de Galois de $\mathbb{Z}_8$) é formado por $R = GR(8, 4) \cong \mathbb{Z}_8[x]/ < x^4 + x + 1 >$. Seja agora $f = (0100) \in R^*$; $\bar{f} = R_2(f) = (0100) = \alpha$ gera um subgrupo de ordem $n^4 - 1 = 15$ no corpo inversível K^*. Assim, f deverá gerar um grupo de ordem $15d$ em R^*. Como neste caso as operações são realizadas módulo $x^4 + x + 1$, segue que $x^4 = x + 1 = 7x + 7$. Então, sendo $f = (0100) = x$ encontramos que $f^{60} = (1000)$.

Portanto, $d = 4$ e f gera um grupo de ordem 60 em R^* e disto segue que $f^4 = x^4 = 7x + 7$ gera um subgrupo de ordem 15. Com isso, temos que $\beta = 7x + 7$ é um elemento primitivo de G_{15}.

Agora, iremos determinar o polinômio $g(x)$ para especificar o código BCH sobre o anel $\mathbb{Z}_8$ de comprimento $n = 15$. O polinômio gerador $g(x)$ será o mínimo múltiplo comum dos polinômios minimais dos elementos $\beta, \beta^2, \beta^3, \beta^4$.

Logo, o polinômio minimal de β é dado por:

$$M_1(x) = x^4 + 4x^3 + 6x^2 + 3x + 1.$$

Esse também é o polinômio minimal de β^2 e β^4. E o polinômio minimal de β^3 é dado por

$$M_3(x) = x^4 + x^3 + x^2 + x + 1.$$

E assim, $g(x) = mmc(M_1(x), M_3(x)) = x^8 + 5x^7 + 3x^6 + 6x^5 + 7x^4 + 6x^3 + 2x^2 + 4x + 1$ gera um código BCH de comprimento 15.

2.3 BIOLOGIA

Nesta seção serão apresentados conceitos biológicos necessários para a compreensão deste trabalho. Na Subseção 2.3.1 são mostradas as principais características das células.

Na Subseção 2.3.2 o estudo se refere aos ácidos nucléicos. Nas Subseções 2.3.3, 2.3.4 e 2.3.5 os processos de duplicação, transição e síntese, respectivamente, são explicados. Na Subseção 2.3.6 são apresentadas as proteínas e os principais aminoácidos. O código genético é apresentado na Subseção 2.3.7. E por fim, na Subseção 2.3.8 é apresentado um estudo sobre as mutações e suas consequências no organismo.

Os conceitos apresentados a seguir podem ser encontrados de forma detalhada em Alberts *et al.* (2010), Hib e Robertis (2006) e Griffits *et al.* (2006).

2.3.1 Célula

As células podem ser definidas como a unidade fundamental da vida. Representam, também, o primeiro nível de organização dos seres vivos e a menor porção do organismo, sendo capaz, sozinha, de obter energia, crescer e se reproduzir. Além de tudo, essas estruturas são responsáveis por carregar informações genéticas que poderão ser propagadas durante a divisão celular.

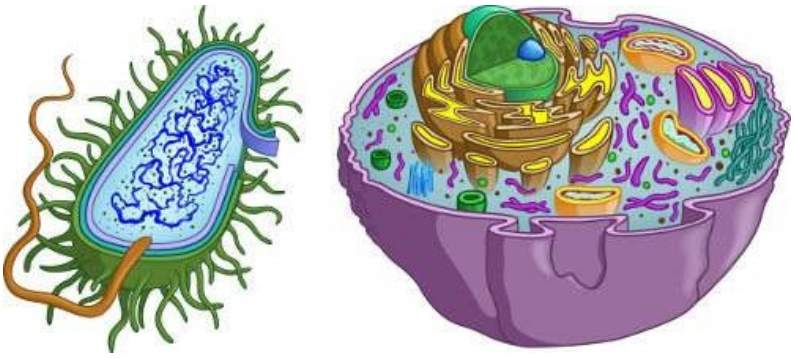
De acordo com a Teoria Celular, todos os seres vivos são formados por células, porém, elas se diferenciam em relação ao formato e a função que exercem em cada organismo. Além disso, as principais atividades que definem a vida ocorrem no interior das células. Logo, para compreender a "máquina da vida" é necessário conhecer as células.

As células desempenham funções diferentes e possuem formatos diversos, assim, morfologicamente, elas apresentam características bem distintas. No entanto, em relação a sua organização interna manifestam semelhanças, sendo possível dividi-las em dois tipos básicos: procariontes e eucariontes. As células procariontes possuem como característica principal a ausência de um núcleo definido, isto é, o material genético não está coberto por uma membrana nuclear e, portanto, fica disperso em um conteúdo que delimita a membrana celular, conhecido como citoplasma. No entanto, as células procariontes não se diferenciam apenas pela inexistência de um núcleo, essas células também são desprovidas de estruturas como mitocôndrias, retículos endoplasmáticos, complexo golgiense e vacúolos. Como representantes de organismos procariontes, é possível citar as bactérias e algas azuis.

Já as células eucariontes formam os organismos unicelulares ou pluricelulares mais comuns do planeta, como por exemplo: os fungos, as plantas e os animais. As células eucariontes, se comparadas com as procariontes, são mais complexas. A principal característica que as difere é a presença de um núcleo definido nas células eucariontes. Além do mais, essas células possuem retículos endoplasmáticos, complexo golgiense, mitocôndrias e cloroplastos. A Figura 2 ilustra um exemplo de célula procarionte e eucarionte.

É importante destacar, que o núcleo comanda as atividades celulares e no seu interior estão presentes os cromossomos, que são sequências do DNA que possuem genes e nucleotídeos. Diante de tal informação, o nosso estudo baseia-se nas células eucariontes por possuírem núcleo.

Figura 2 – Célula procarionte, à esquerda, e eucarionte, à direita



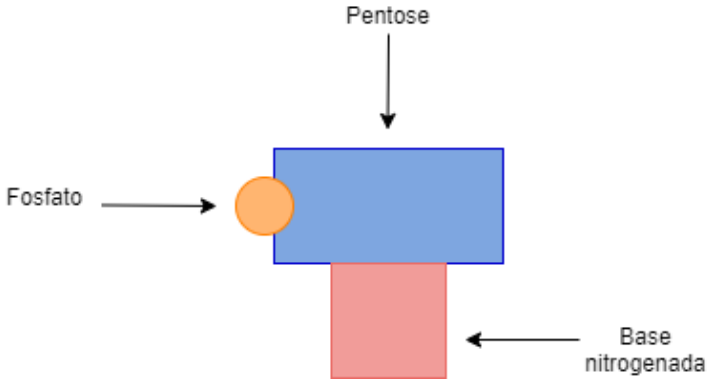
Fonte: ALBERTS *et al.*, 2010, p. 86.

2.3.2 Ácidos Nucleicos

Os ácidos nucleicos são denominados dessa forma por possuírem característica ácida, além de serem descobertos no interior das células. São macromoléculas de suma importância, pois estão relacionados ao funcionamento das células e constituem os genes que são responsáveis pela herança genética.

Existem dois tipos de ácidos nucleicos: o ácido desoxirribonucleico e o ácido ribonucleico, conhecidos pelas siglas DNA e RNA, respectivamente. Tanto o DNA quanto o RNA são compostos essencialmente de três componentes: glicerídeos, do grupo das pentoses, sendo desoxirribose no DNA e ribose no RNA; ácido fosfórico e bases nitrogenadas. Dentre os cinco tipos de bases nitrogenadas que podem ser encontradas nos ácidos nucleicos, três acometem o DNA bem como o RNA, são elas: adenina (A), citosina (C) e guanina (G). A base nitrogenada timina (T) ocorre somente no DNA e a uracila (U) apenas no RNA. Os três componentes dos ácidos nucleicos organizados formam o que denominamos de nucleotídeo, como pode ser visto na Figura 3. Os nucleotídeos são moléculas presentes nas células que participam de importantes reações do metabolismo celular, dentre as quais podemos destacar, a título de exemplo, a transferência de energia na forma de ATP e o armazenamento e transmissão da informação genética.

Figura 3 – Componentes químicos que formam um nucleotídeo

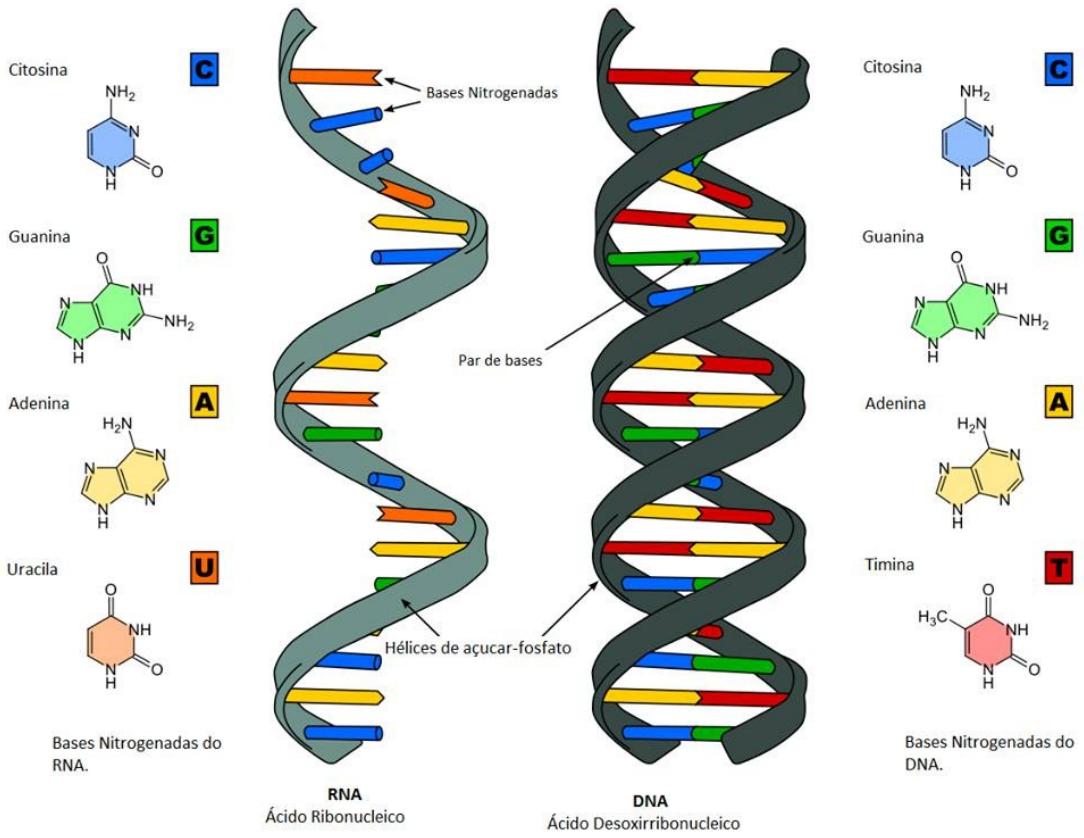


Fonte: Da Autora.

As moléculas de DNA são formadas por uma fita dupla de polinucleotídeos, vários nucleotídeos ligados, que possuem formato similar a de uma escada helicoidal. Essas duas fitas são unidas por meio de pontes de hidrogênio, que estão localizadas entre pares de bases específicas: a timina emparelha-se com a adenina e a guanina une-se com a citosina, respeitando a complementariedade da Regra de Chargaff, que pode ser vista na Figura 4.

Já as moléculas de RNA são, geralmente, constituídas de uma única fita que se enrola entre si devido ao emparelhamento entre as bases complementares: a uracila emparelha-se com a adenina e a guanina une-se com a citosina, seguindo a Regra de Chargaff. Existem três tipos de RNA: RNA mensageiro (RNAm), RNA transportador (RNAt) e RNA ribossômico (RNAr). O RNA mensageiro é uma cópia das fitas de DNA, ficando responsável em levar as informações obtidas do DNA até o citoplasma, onde as proteínas serão produzidas. O RNA transportador é encarregado de transportar os aminoácidos que serão usados na formação das proteínas até o ribossomo, enquanto, que o RNA ribossômico faz parte da constituição dos ribossomos. A estrutura do RNA pode ser observada na Figura 4.

Figura 4 – Diferença estrutural entre o RNA e o DNA, em ordem



Fonte: GRIFFITHS *et al.*, 2006, p. 174.

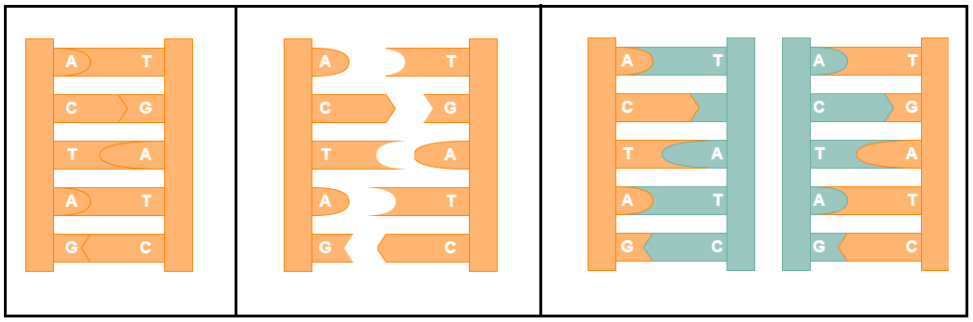
2.3.3 Duplicação do DNA

A duplicação ou replicação do DNA é um processo que assegura a autoduplicação de informações genéticas, sendo importante para a propagação de características hereditárias, ou até mesmo para a regeneração de algum tecido lesionado. Watson e Crick, em 1953, apresentaram um mecanismo de duplicação da molécula de DNA e isso auxiliou os cientistas a compreenderem, hoje, de forma detalhada como a duplicação ocorre nas células.

De acordo com o modelo proposto por Watson e Crick, para que ocorra esse processo, é necessário que ocorra alguns eventos na molécula de DNA. Primeiramente, ocorre o rompimento das pontes de hidrogênio, que ligam as bases nitrogenadas, assim as fitas se afastam, gerando duas cadeias de nucleotídeos moldes. Sobre cada uma dessas cadeias de nucleotídeos molde vão se emparelhando novos nucleotídeos que estavam dispersos no núcleo, este emparelhamento ocorre de maneira que as bases nitrogenadas sejam complementares: nucleotídeos com timina (T) ligam-se a adenina (A), nucleotídeos com citosina (C) emparelham-se com guanina (G), nucleotídeos com guanina (G) emparelham-se com citosina (C) e nucleotídeos com adenina (A) emparelham-se com timina (T).

Ao final do processo, quando as duas fitas originais tiverem sido completadas por novos nucleotídeos, existirão duas moléculas de DNA idênticas. Este processo é esquematizado pela Figura 5.

Figura 5 – Esquema do processo de duplicação de uma molécula de DNA



Fonte: Da Autora.

Vale ressaltar, que o processo de duplicação do DNA só ocorre mediante a presença de uma enzima, chamada de DNA polimerase, sendo ela responsável pela quebra das pontes de hidrogênio e por orientar o emparelhamento dos novos nucleotídeos com as fitas rompidas.

2.3.4 Transcrição gênica

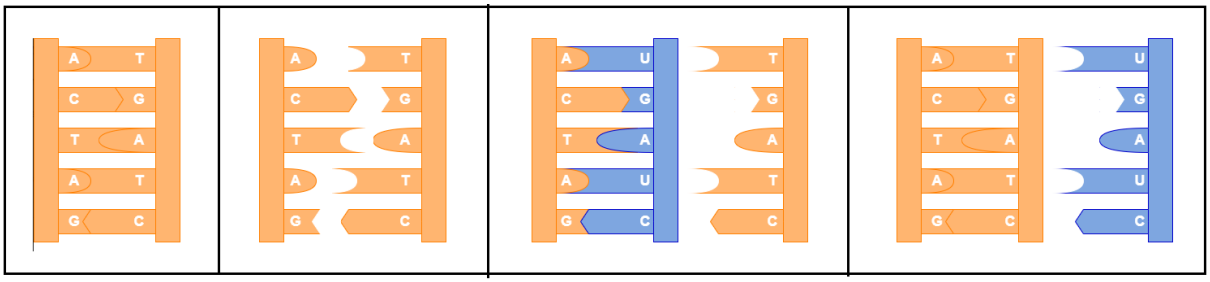
A transcrição gênica é o processo pelo qual é sintetizada uma molécula de RNA, a partir de uma molécula de DNA. Por meio desse processo, são obtidos os três tipos de

RNA (RNAm, RNAt e RNAr). É importante destacar, que o processo de transcrição só ocorre mediante a presença da enzima RNA polimerase.

A síntese de RNA se inicia com a separação das duas fitas de DNA, sendo que uma dessas fitas será utilizada como molde para a formação do RNA, enquanto que a outra fita permanecerá inativa durante todo esse processo. Após a separação das fitas, a enzima RNA polimerase orienta o emparelhamento dos ribonucleotídeos livres na fita de DNA molde. Esse emparelhamento segue a seguinte regra: uracila (U) se emparelha à adenina (A) da fita de DNA molde, adenina (A) emparelha-se com uracila (U), citosina (C) emparelha-se com guanina (G) e guanina (G) emparelha-se com citosina (C).

À medida que ocorrem esses emparelhamentos, os ribonucleotídeos se unem, formando a molécula de RNA. Ao final desse processo, a molécula de RNA separa-se do DNA molde e esta volta a se unir com a outra fita inativa. Este processo é ilustrado na Figura 6.

Figura 6 – Esquema do processo de transcrição



Fonte: Da Autora.

2.3.5 Síntese proteica

O processo de síntese proteica, também conhecido como tradução, baseia-se na leitura de um RNA específico, o RNAm, e na união de aminoácidos equivalentes, a três sequências de bases nitrogenadas presentes no RNAm. Essas trincas de bases nitrogenadas são denominadas códons.

Para que ocorra o processo de tradução, é necessário a participação de um RNAr, um RNAm, vários RNAt, aminoácidos e enzimas, que possuem papéis bem específicos nesse processo. O RNAr é encarregado de formar o ribossomo, que são estruturas nas quais são produzidas as proteínas, o RNAm possui a sequência de bases nitrogenadas que orientará a síntese e o RNAt é responsável por carregar os aminoácidos para a formação da molécula de proteína.

A síntese proteica é realizada da seguinte forma: um ribossomo encaixa-se em uma das extremidades do RNAm e percorre toda a sua extensão. Conforme o ribossomo vai se deslocando, os RNAt vão encaixando os aminoácidos em uma ordem definida por trincas de bases nitrogenadas do RNAm. Desse modo, as informações inscritas no RNAm vão sendo traduzidas como uma sequência de aminoácidos na proteína.

2.3.6 Proteínas e aminoácidos

As proteínas são as macromoléculas de maior abundância nos seres vivos, estando presentes em todas as células, além disso, desempenham funções vitais dos seres vivos. Essas macromoléculas assumem funções biológicas distintas, tendo suas propriedades presentes nos músculos, cabelos, unhas, anticorpos, entre outros. É conhecido também, que a falta ou modificação de alguma proteína pode gerar algumas doenças e erros no metabolismo, como por exemplo, os portadores da doença fenilcetonúria, que é causada pelo acúmulo da proteína fenilalanina.

As proteínas são polímeros constituídos basicamente de aminoácidos. Estes, por sua vez, são moléculas orgânicas, de modo geral são denominadas dessa forma as moléculas que possuem carbono e hidrogênio em sua estrutura. Assim, os aminoácidos são quimicamente estruturados por um átomo de carbono, um átomo de hidrogênio, um grupo amina, um grupo carboxílico e uma cadeia lateral (grupo-R), que varia de acordo com cada aminoácido. Essa cadeia lateral é responsável por diferenciar os aminoácidos em tamanhos, funções e solubilidade em água.

A união de dois aminoácidos gera um dipeptídeo, a junção de três aminoácidos forma um tripeptídeo, e assim segue sucessivamente, sendo que quando ocorre a união de vários aminoácidos temos uma cadeia polipeptídica. Essas estruturas são formadas por no máximo 20 tipos diferentes de aminoácidos, sendo possível agrupá-los em quatro classes diferentes, de acordo com os seus grupamentos: hidrofóbicos, hidrofílicos, básicos e ácidos.

a) aminoácidos hidrofóbicos: são os menos solúveis, em decorrência da ausência de grupamento hidrofóbico. São eles: alanina, leucina, isoleucina, valina, prolina, fenilalanina, triptofano, metionina e glicina;

b) aminoácidos hidrofílicos: possuem grupamentos hidrofílicos que não se ionizam. São eles: serina, treonina, tirosina, asparagina, glutamina e cisteína.

c) aminoácidos básicos: possuem grupamentos com 6 carbonos e a carga positiva é localizada em um átomo de hidrogênio presente no grupo lateral. São eles: lisina, arginina e histidina.

d) aminoácidos ácidos: se ionizam em pH fisiológico adquirindo carga negativa no grupamento carboxila. São eles: ácido aspártico e ácido glutâmico.

2.3.7 Código genético

O código genético pode ser compreendido como a relação das trincas de bases nitrogenadas encontradas no RNAm (A, U, G, e C) e os aminoácidos presentes nas proteínas. Esse código é universal, sendo o mesmo para todos os seres vivos da Terra e degenerado, porque mais de um códon pode codificar um mesmo aminoácido.

O código genético possui 64 códons, onde somente 61 codificarão os 20 tipos diferentes de aminoácidos. Os 64 códons são apresentados no Quadro 1.

Quadro 1 – 64 códons pertencentes ao código genético

primeira posição extremidade 5'↓	segunda posição				terceira posição extremidade 3'↓
	U	C	A	G	
U	UUU	UCU	UAU	UGU	U
	UUC	UCC	UAC	UGC	C
	UUA	UCA	UAA	UGA	A
	UUG	UCG	UAG	UGG	G
C	CUU	CCU	CAU	CGU	U
	CUC	CCC	CAC	CGC	C
	CUA	CCA	CAA	CGA	A
	CUG	CCG	CAG	CGG	G
A	AUU	ACU	AAU	AGU	U
	AUC	ACC	AAC	AGC	C
	AUA	ACA	AAA	AGA	A
	AUG	ACG	AAG	AGG	G
G	GUU	GCU	GAU	GGU	U
	GUC	GCC	GAC	GGC	C
	GUA	GCA	GAA	GGA	A
	GUG	GCG	GAG	GGG	G

Fonte: OLIVEIRA, 2012, p. 35.

Os códons UAA, UAG e UGA não especificam nenhum aminoácido e são usados para indicar a interrupção da síntese de uma proteína, são conhecidos como códons de parada. O códon AUG, possui dupla função, além de codificar a proteína metionina também marca o início da síntese proteica.

2.3.8 Mutações

Biologicamente, mutações podem ser compreendidas como mudanças na sequência dos nucleotídeos do material genético. Essas mutações podem ser provocadas por erros durante a cópia do material na divisão celular, por exposição a fatores químicos e físicos, ou vírus.

As mutações podem ser benéficas, maléficas ou neutras e ocorrem de forma aleatória. Isto quer dizer que uma mudança no DNA pode proporcionar melhorias no organismo, como por exemplo, uma borboleta pode produzir uma prole com mutações que mudem a cor dos descendentes desse indivíduo, tornando-os mais difíceis de serem vistos pelos predadores, com isso as chances dessa prole sobreviver e se reproduzir será maior; podem não causar nenhuma vantagem, isto é, as mutações não influenciam na aptidão dos indivíduos; ou podem ainda desencadear prejuízos e problemas, como é o caso das doenças genéticas causadas pela alteração de alguma proteína, como no caso do albinismo, onde ocorre a ausência parcial ou total da melanina.

As mutações podem ocorrer em dois níveis distintos: gênicas ou cromossômicas. As mutações cromossômicas traduzem-se em alterações da estrutura ou do número de cro-

mossomos. Já as alterações gênicas alteram a sequência de nucleotídeos do DNA. Estruturalmente, as mutações podem ser classificadas em:

1. Mutações de pequena escala, como aquelas que afetam um gene em um ou poucos nucleotídeos; como mutação de ponto, inserção e deleção.

    a) **mutação de ponto:** essas mutações ocorrem quando apenas uma base nitrogenada é alterada, o que leva o códon a ficar diferente da sequência normal. Mutações pontuais podem ser classificadas em três tipos: mutação silenciosa, "missense" e mutação sem sentido. As mutações silenciosas são aquelas em que uma base é modificada, no entanto devido à característica de redundância do código genético, acaba por codificar o mesmo aminoácido. Já nas mutações missense ocorre alteração de uma das bases do DNA, tendo como consequência a substituição de um aminoácido por outro na proteína codificada. Por sua vez, a mutação sem sentido produz um códon de STOP que impede que a proteína seja produzida integralmente.

    b) **inserção:** este tipo de mutação ocorre devido a adição de um ou mais nucleotídeo no DNA. Inserções na região codificadora de um gene pode causar mudança na leitura de um códon ou podem alterar o corte do RNAm.

    c) **deleção:** nas mutações por deleção, retira-se um ou mais nucleotídeos, alterando a composição da proteína. Vale destacar ainda, que mutações por deleção não é o oposto de inserção, pois enquanto que a deleção é aleatória, inserção consiste em uma sequência específica sendo inserida em locais que não são completamente aleatórios.

2. Mutações de grande escala da estrutura do cromossomo, como duplicação, deleção de regiões cromossômicas, translocação e inversão.

    a) **duplicação:** este tipo de mutação é dado pela criação de várias cópias de uma região cromossômica.

    b) **deleção de regiões cromossômicas:** já esse tipo de mutação ocorre quando o cromossomo é desprovido de uma parte, levando à perda dos genes presentes nessas regiões.

    c) **translocação:** é representada pela transferência do segmento de um cromossomo preso a outro cromossomo que não é homólogo ao seu.

    d) **inversão:** ocorre a inversão da orientação de um segmento do cromossomo.

Por garantir variabilidade genética, isto é, que indivíduos tenham DNA diferentes, as mutações são consideradas o mecanismo que permite a seleção natural, tornando-se

um ponto importante da evolução dos seres vivos. É por meio delas, que características vantajosas serão multiplicadas nas gerações subsequentes ou características deletérias irão desaparecer.

2.4 ANALOGIA ENTRE O SISTEMA DE COMUNICAÇÃO DIGITAL E O SISTEMA DE COMUNICAÇÃO DE INFORMAÇÃO GENÉTICA

Os processos de replicação, transcrição e replicação, mostrados nas Subseções 2.3.3, 2.3.4, 2.3.5, respectivamente, também são conhecidos como dogma central da biologia e definem os passos em que a informação passa de código genético para a proteína. Já o diagrama de blocos comentado na Seção 2.2 e ilustrado pela Figura 1 é definido como dogma central da teoria das comunicações. O objetivo dessa seção é relacionar as semelhanças existentes entre o dogma central da biologia e o dogma central da teoria das comunicações. Por meio das informações obtidas nas subseções citadas anteriormente é possível realizar as seguintes associações:

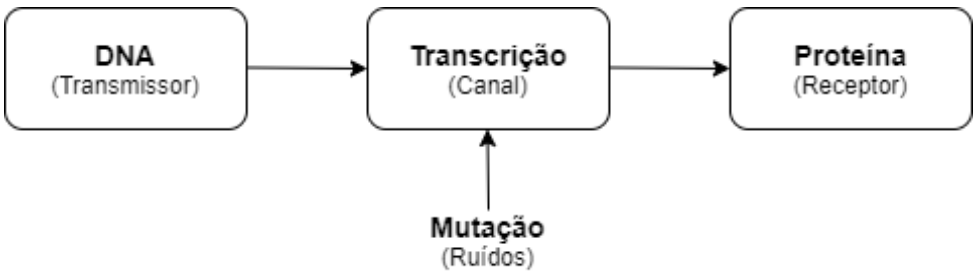
a) em um sistema de comunicação o responsável pela geração das informações a serem transmitidas é o transmissor. Biologicamente quem exerce esta função é o DNA.

b) o processo de transcrição tem como objetivo a transmissão da informação. Durante este processo podem ocorrer alguns erros que irão interferir na informação, como por exemplo, a não leitura de um códon. Do ponto de vista da comunicação, o processo de transcrição é tido como sendo o canal de um sistema de comunicação, e os eventuais erros cometidos durante este processo como sendo o ruído introduzido no canal.

c) o receptor pode ser modelado como o local onde a informação está sendo enviada. Neste caso específico, a nossa informação é a proteína.

Assim, pode-se relacionar cada bloco do sistema de comunicação digital com cada bloco do sistema de comunicação de informação genética, como mostra a Figura 7.

Figura 7 – Analogia entre o sistema de comunicação digital e o sistema de comunicação de informação genética

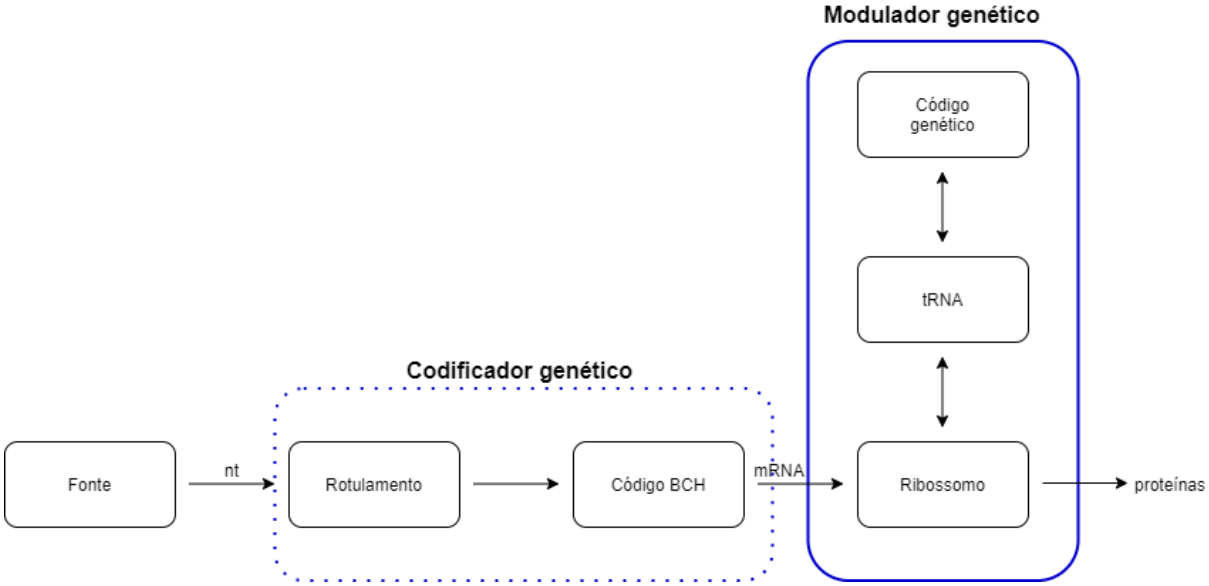


Fonte: ROCHA, 2010, p. 13.

Rocha (2010) apresentaram uma proposta de analogia de um modelo de sistema de comunicação para a importação de proteínas organelares que se baseia em um sistema de

comunicação digital, conforme Figura 8, com o objetivo de identificar nas sequências de DNA estruturas matemáticas associadas.

Figura 8 – Modelo de um sistema de comunicação de informação genética



Fonte: ROCHA, 2010, p. 16.

Este modelo é constituído por um codificador genético e um modulador genético. O codificador é o responsável pelas mudanças das bases nitrogenadas adenina, citosina, guanina e timina em um alfabeto matemático, estabelecendo o mapeamento utilizado pelo código. O código BCH refere-se à palavra-código na saída do codificador que está relacionada à sequência de direcionamento em termos de nucleotídeos, enquanto que o modulador consiste do código genético, do RNAt e do ribossomo e a palavra código na saída se refere à sequência de direcionamento em termos de aminoácidos.

No entanto, para verificar a validade desse modelo, isto é, mostrar que é possível identificar sequências de DNA por meio dos códigos BCH foi necessário o desenvolvimento de um algoritmo para a geração de sequências de DNA que, por meio da codificação, identifica e reproduz tais sequências.

3 ALGORITMO DE GERAÇÃO DE PROTEÍNAS

Neste capítulo, será apresentado o algoritmo de geração de proteínas, proposto por Rocha (2010) e Faria (2011) que identifica e reproduz diferentes sequências de DNA por meio dos códigos BCH. Esse algoritmo possibilita o reconhecimento de uma estrutura de códigos corretores de erros em sequências de DNA, além de permitir uma nova classificação dessas sequências sob um ponto de vista matemático.

Na Seção 3.1 será apresentada a descrição do algoritmo e, na Seção 3.2, será apresentada uma aplicação da execução do programa, por meio de um exemplo, que reproduz uma sequência de DNA relacionada a uma proteína mitocondrial. Na Seção 3.3 serão apresentadas algumas considerações acerca da existência de códigos corretores de erros relacionados à sequências de DNA.

3.1 DESCRIÇÃO DO ALGORITMO DE GERAÇÃO DE SEQUÊNCIAS DE DNA

Um dos maiores desafios da codificação genética é identificar uma estrutura de códigos corretores de erros na estrutura de DNA (ROCHA, 2010). No entanto, Rocha (2010) e Faria (2011) mostraram a existência dos códigos corretores de erros associados às sequências de DNA e que elas são identificadas como palavras código de um código BCH sobre a extensão de um anel de Galois ou sobre a extensão de um corpo de Galois, sendo que a principal diferença entre a geração dos códigos BCH sobre essas duas extensões reside no fato de que as raízes do polinômio gerador dos códigos BCH sobre anel encontram-se na extensão do anel $\mathbb{Z}_q$. Em particular, neste capítulo será considerada a construção dos códigos BCH sobre as extensões do anel de Galois da classe de restos $\mathbb{Z}_4$.

O algoritmo de geração de proteínas, proposto por Rocha (2010) e Faria (2011) realiza, exaustivamente, a construção de códigos BCH sobre estrutura de anel com parâmetros (n, k, d_H), capaz de reproduzir sequências de DNA de comprimento $n = 2^r - 1$, com até dois nucleotídeos de diferença da sequência de DNA disponível no NCBI. Além disso, possui a capacidade de correção de erros dada por meio da relação $d_H = 2t + 1$, em que t indica a quantidade de erros. Os parâmetros do código BCH desse algoritmo são caracterizados da seguinte forma: $n =$ comprimento das sequências de DNA; $k =$ comprimento da sequência de informação responsável pela geração da sequência de DNA e $d_H =$ o menor número de posições em que quaisquer duas palavras código diferem.

No algoritmo, os códigos BCH sobre anéis serão construídos em todas as distâncias mínimas e em todos os polinômios primitivos e geradores de cada extensão de Galois de grau r, com o objetivo de encontrar um código capaz de gerar as sequências de DNA. Como para cada um dos casos teremos um novo código BCH, devemos considerar cada um destes códigos como um novo código a ser analisado.

Esse algoritmo pode ser descrito, passo a passo, da seguinte forma:

Passo 1 - Especificar a estrutura matemática e o alfabeto do código.

Passo 2 - Determinar a extensão de Galois.

Passo 3 - Determinar todos os polinômios primitivos $p(x)$ relacionados à extensão de Galois.

Passo 4 - Determinar a extensão do corpo $GF(2)$.

Passo 5 - Determinar a extensão do anel $\mathbb{Z}_4$.

Passo 6 - Determinar o grupo das unidades.

Passo 7 - Determinar o polinômio gerador da matriz G, $g(x)$.

Passo 8 - Determinar o polinômio gerador da matriz H, $h(x)$.

Passo 9 - Determinar a matriz G.

Passo 10 - Determinar a matriz H e a sua transposta H^T.

Passo 11 - Rotular a sequência de DNA utilizando o Passo 1.

Passo 12 -Verificar se a sequência de DNA é palavra-código de acordo com os padrões de erros estabelecidos: D(a, b) = 0, D(a, b) = 1 e D(a, b) = 2.

Passo 13 - Comparar todas as palavras código armazenadas no Passo 12 com a sequência de DNA original e mostrar onde os erros ocorreram.

Passo 14 - Voltar para o Passo 7 e determinar outro $g(x)$.

Passo 15 - Repetir os Passos 8 ao 12 para o $g(x)$ obtido no Passo 14, até que se esgote todas as possibilidades de $g(x)$.

Passo 16 - Voltar para o Passo 3 e escolher outro $p(x)$, e então, repetir os Passos 4 ao 14 até esgotar todos os $p(x)$ do Passo 3.

Passo 17 - Fim.

A Figura 9 a seguir apresenta um fluxograma com o detalhamento do algoritmo descrito anteriormente.

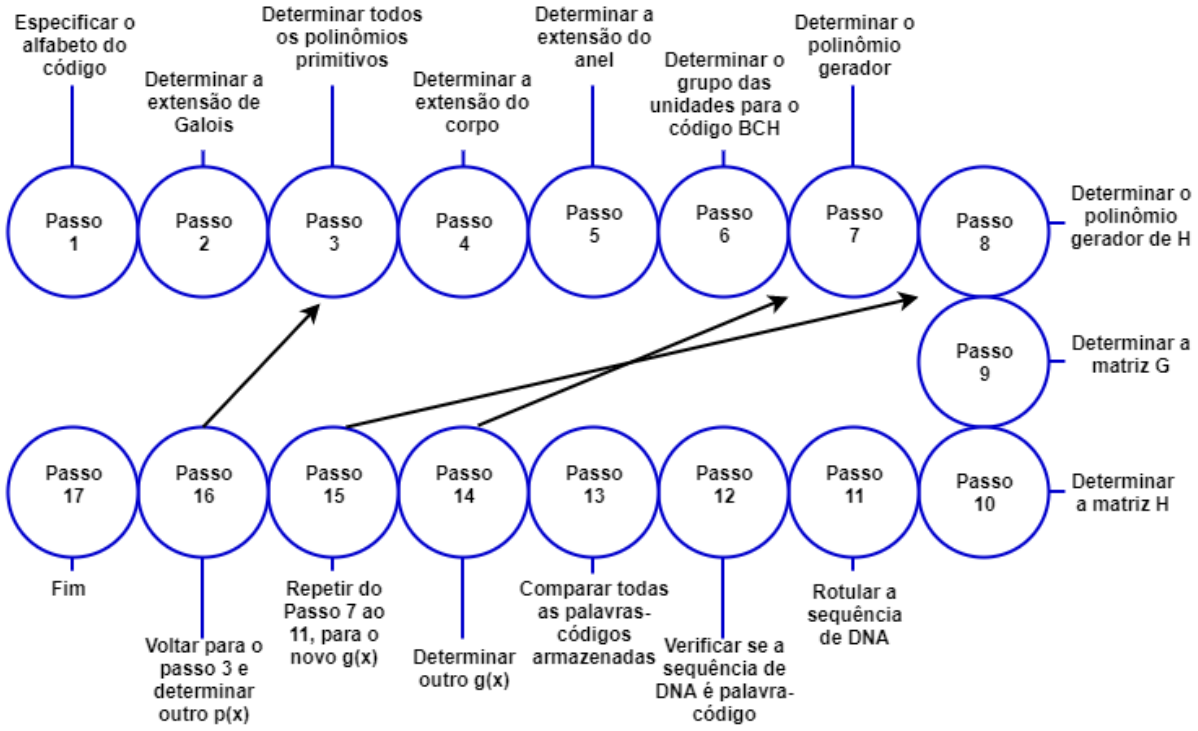
3.2 EXEMPLO: CONSTRUÇÃO DO CÓDIGO BCH (n, k, d_H) SOBRE $GR(4, 6)$

A fim de ilustrar como é feita a construção do código BCH sobre o anel $\mathbb{Z}_4$ pelo algoritmo de geração de proteínas, será apresentado um exemplo da reprodução de uma sequência de direcionamento de uma proteína mitocondrial - GI número 832917 com comprimento $n = 63$ nucleotídeos, por meio dos 17 passos listados na Seção 3.1. Esse exemplo pode ser visto em Rocha (2010).

Passo 1 - Especificar a estrutura matemática e o alfabeto do código

O código genético é composto pela combinação das 4 bases nitrogenadas: adenina (A), citosina (C), guanina (G) e timina (T) ou uracila (U), assim, o alfabeto 4-ário do codigo genético pode ser denotado pelo conjunto $N = \{A, C, G, T/U\}$. Como a estrutura algébrica do alfabeto do código genético é desconhecida, o conjunto N pode ser relacionado ao alfabeto 4-ário dos códigos corretores de erros sobre uma estrutura de anel, indicado por

Figura 9 – Fluxograma do algoritmo de geração de proteínas



Fonte: Da autora.

$\mathbb{Z}_4 = \{0, 1, 2, 3\}$, que obedece as operações de adição e multiplicação módulo 4, conforme as Tabelas 8 e 9. A escolha dos elementos que constituem o conjunto $\mathbb{Z}_4$ é justificada por ser a mais simples estrutura de anel, a qual confere ao algoritmo uma menor complexidade computacional.

Tabela 8 – Adição módulo 4

$+$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Fonte: Da autora.

Tabela 9 – Multiplicação módulo 4

$\cdot$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Fonte: Da autora.

Passo 2 - Determinar a extensão de Galois

Uma das condições para a execução do algoritmo é que o comprimento da sequência de DNA obedeça a restrição $n = 2^r - 1$, onde o r denota o grau da extensão do corpo de Galois, pois assim garantimos que a fatoração de $x^n - 1$ na extensão $GR(4, r)$ seja única. Aqui, será analisada a sequência de DNA cujo comprimento é $n = 63$ nucleotídeos. Assim, o grau da extensão do corpo de Galois é $r = 6$, pois $n = 2^r - 1 = 2^6 - 1 = 63$.

Passo 3 - Determinar todos os polinômios primitivos $p(x)$ relacionados à extensão de Galois

Neste passo, são informados todos os polinômios primitivos $p(x)$ relacionados ao grau de extensão do corpo de Galois $r = 6$, como mostra o Quadro 2. Estes polinômios primitivos podem ser encontrados em Costelo e Lin 1983.

Quadro 2 – Polinômios primitivos da extensão de Galois de grau $r = 6$

Polinômios primitivos $p(x)$	
$p_1(x) = x^6 + x + 1$	$p_4(x) = x^6 + x^5 + x^2 + x + 1$
$p_2(x) = x^6 + x^4 + x^3 + x + 1$	$p_5(x) = x^6 + x^5 + x^3 + x^2 + 1$
$p_3(x) = x^6 + x^5 + 1$	$p_6(x) = x^6 + x^5 + x^4 + x + 1$

Fonte: Da autora.

É importante destacar que, quanto maior o grau da extensão de Galois, maior será a quantidade de polinômios primitivos associados, aumentando assim, a complexidade computacional para a realização dos cálculos.

Passo 4 - Determinar a extensão do corpo $GF(2)$

Neste passo, é obtida a extensão do corpo $GF(2)$ por meio de um polinômio primitivo de grau $r = 6$. Considere o corpo de Galois $GF(2^r) = GF(2^6) = GF(64) = \mathbb{F}_{64}$ e assuma que o polinômio primitivo $p(x)$ utilizado é o $p_1(x) = x^6 + x + 1$.

Seja α um elemento primitivo em $\mathbb{F}_{64}$, assim α é uma raíz do polinômio primitivo $x^6 + x + 1$, ou seja, $\alpha^6 + \alpha + 1 = 0$, implicando em $\alpha^6 = -\alpha - 1$. No entanto, como os coeficientes dos elementos que formam o conjunto $\mathbb{F}_{64}$ pertecem a $\mathbb{F}_2$, realizamos a redução módulo 2 desses coeficientes obtendo $\alpha^6 = \alpha + 1$. Os elementos de $\mathbb{F}_{64}$ são mostrados na Tabela 10.

Tabela 10 – Elementos de $\mathbb{F}_{64}$

Potência	**Vetorial**	**Potência**	**Vetorial**	**Potência**	**Vetorial**
0	(000000)	α^{21}	(110111)	α^{43}	(111011)
α^0	(100000)	α^{22}	(101011)	α^{44}	(101101)
α^1	(010000)	α^{23}	(100101)	α^{45}	(100101)
α^2	(001000)	α^{24}	(100010)	α^{46}	(010011)
α^3	(000100)	α^{25}	(010001)	α^{47}	(111001)
α^4	(000010)	α^{26}	(111000)	α^{48}	(101100)
α^5	(000001)	α^{27}	(011100)	α^{49}	(010110)
α^6	(110000)	α^{28}	(001110)	α^{50}	(001011)
α^7	(011000)	α^{29}	(000111)	α^{51}	(110101)
α^8	(001100)	α^{30}	(110011)	α^{52}	(101010)
α^9	(000110)	α^{31}	(101001)	α^{53}	(010101)
α^{10}	(000011)	α^{32}	(100100)	α^{54}	(111010)

Continua

Tabela 10 – Elementos de $\mathbb{F}_{64}$

Potência	Vetorial	Potência	Vetorial	Potência	Vetorial
α^{11}	(110001)	α^{33}	(010010)	α^{55}	(011101)
α^{12}	(101000)	α^{34}	(001001)	α^{56}	(111110)
α^{13}	(010100)	α^{35}	(110100)	α^{57}	(011111)
α^{14}	(001010)	α^{36}	(011010)	α^{58}	(111111)
α^{15}	(000110)	α^{37}	(001101)	α^{59}	(101111)
α^{16}	(110010)	α^{38}	(110110)	α^{60}	(100111)
α^{17}	(011001)	α^{39}	(011011)	α^{61}	(100011)
α^{18}	(111100)	α^{40}	(111101)	α^{62}	(100001)
α^{19}	(011110)	α^{41}	(101110)		
α^{20}	(001111)	α^{42}	(010111)		

Fonte: Da autora.

Passo 5 - Determinar a extensão do anel $\mathbb{Z}_4$

A extensão do anel $GF(4,6)$ é dada pelo quociente do conjunto de todos os polinômios com coeficientes em $\mathbb{Z}_4$ pelo ideal gerado pelo mesmo polinômio primitivo $p(x)$, usado na extensão do corpo, realizada no Passo 4, ou seja,

$$\frac{\mathbb{Z}_4[x]}{\langle x^6+x+1\rangle} = \left\{b_0 + b_1x + \ldots + b_5x^5 : b_i's \in \mathbb{Z}_4\right\}.$$

Como no Passo 4, consideramos α como sendo um elemento primitivo. Temos que α é uma raíz do polinômio x^6+x+1, isto é, $\alpha^6+\alpha+1=0$, implicando em $\alpha^6=-\alpha-1$. Como os coeficientes dos polinômios em $GF(4,6)$ estão em $\mathbb{Z}_4$, ao realizar a operação módulo 4 obtemos $\alpha^6=3\alpha+3$. Considerando $f=(010000)=\alpha$, todos os elementos não nulos e invertíveis do grupo cíclico do grupo $GR^*(4,6)$ são determinados, como mostra a Tabela 11.

Tabela 11 – Elementos do grupo cíclico $GR^*(4,6)$

$GR^*(4,6)$	Vetorial	$GR^*(4,6)$	Vetorial	$GR^*(4,6)$	Vetorial
1	(100000)	$f^{43}=\alpha^{43}$	(131011)	$f^{86}=\alpha^{86}$	(100303)
$f=\alpha$	(010000)	$f^{44}=\alpha^{44}$	(303101)	$f^{87}=\alpha^{87}$	(120030)
$f^2=\alpha^2$	(001000)	$f^{45}=\alpha^{45}$	(320310)	$f^{88}=\alpha^{88}$	(012003)
$f^3=\alpha^3$	(000100)	$f^{46}=\alpha^{46}$	(032031)	$f^{89}=\alpha^{89}$	(111200)
$f^4=\alpha^4$	(000010)	$f^{47}=\alpha^{47}$	(333203)	$f^{90}=\alpha^{90}$	(011120)
$f^5=\alpha^5$	(000001)	$f^{48}=\alpha^{48}$	(103320)	$f^{91}=\alpha^{91}$	(001112)
$f^6=\alpha^6$	(330000)	$f^{49}=\alpha^{49}$	(010332)	$f^{92}=\alpha^{92}$	(220111)
$f^7=\alpha^7$	(033000)	$f^{50}=\alpha^{50}$	(221033)	$f^{93}=\alpha^{93}$	(312011)

Continua

Tabela 11 Elementos do grupo cíclico $GR^*(4,6)$

$GR^*(4,6)$	**Vetorial**	$GR^*(4,6)$	**Vetorial**	$GR^*(4,6)$	**Vetorial**
$f^8 = \alpha^8$	(003300)	$f^{51} = \alpha^{51}$	(132103))	$f^{94} = \alpha^{94}$	(321201)
$f^9 = \alpha^9$	(000330)	$f^{52} = \alpha^{52}$	(123210)	$f^{95} = \alpha^{95}$	(322120)
$f^{10} = \alpha^{10}$	(000033)	$f^{53} = \alpha^{53}$	(012321)	$f^{96} = \alpha^{96}$	(032212)
$f^{11} = \alpha^{11}$	(110003)	$f^{54} = \alpha^{54}$	(331232)	$f^{97} = \alpha^{97}$	(222221)
$f^{12} = \alpha^{12}$	(121000)	$f^{55} = \alpha^{55}$	(213123)	$f^{98} = \alpha^{98}$	(312322)
$f^{13} = \alpha^{13}$	(012100)	$f^{56} = \alpha^{56}$	(131312)	$f^{99} = \alpha^{99}$	(211232)
$f^{14} = \alpha^{14}$	(001210)	$f^{57} = \alpha^{57}$	(233131)	$f^{100} = \alpha^{100}$	(201123)
$f^{15} = \alpha^{15}$	(000121)	$f^{58} = \alpha^{58}$	(313313)	$f^{101} = \alpha^{101}$	(130112)
$f^{16} = \alpha^{16}$	(330012)	$f^{59} = \alpha^{59}$	(101331)	$f^{102} = \alpha^{102}$	(232011)
$f^{17} = \alpha^{17}$	(213001)	$f^{60} = \alpha^{60}$	(300133)	$f^{103} = \alpha^{103}$	(313301)
$f^{18} = \alpha^{18}$	(311300)	$f^{61} = \alpha^{61}$	(100013)	$f^{104} = \alpha^{104}$	(321330)
$f^{19} = \alpha^{19}$	(031130)	$f^{62} = \alpha^{62}$	(120001)	$f^{105} = \alpha^{105}$	(032133)
$f^{20} = \alpha^{20}$	(003113)	$f^{63} = \alpha^{63}$	(302000)	$f^{106} = \alpha^{106}$	(113213)
$f^{21} = \alpha^{21}$	(110311)	$f^{64} = \alpha^{64}$	(030200)	$f^{107} = \alpha^{107}$	(121321)
$f^{22} = \alpha^{22}$	(301031)	$f^{65} = \alpha^{65}$	(003020)	$f^{108} = \alpha^{108}$	(302132)
$f^{23} = \alpha^{23}$	(320103)	$f^{66} = \alpha^{66}$	(000302)	$f^{109} = \alpha^{109}$	(210213)
$f^{24} = \alpha^{24}$	(102010)	$f^{67} = \alpha^{67}$	(220030)	$f^{110} = \alpha^{110}$	(131021)
$f^{25} = \alpha^{25}$	(010201)	$f^{68} = \alpha^{68}$	(022003)	$f^{111} = \alpha^{111}$	(303102)
$f^{26} = \alpha^{26}$	(331020)	$f^{69} = \alpha^{69}$	(112200)	$f^{112} = \alpha^{112}$	(210310)
$f^{27} = \alpha^{27}$	(033102)	$f^{70} = \alpha^{70}$	(011220)	$f^{113} = \alpha^{113}$	(021031)
$f^{28} = \alpha^{28}$	(223310)	$f^{71} = \alpha^{71}$	(001122)	$f^{114} = \alpha^{114}$	(332103)
$f^{29} = \alpha^{29}$	(022331)	$f^{72} = \alpha^{72}$	(220112)	$f^{115} = \alpha^{115}$	(103210)
$f^{30} = \alpha^{30}$	(332233)	$f^{73} = \alpha^{73}$	(202011)	$f^{116} = \alpha^{116}$	(010321)
$f^{31} = \alpha^{31}$	(103223)	$f^{74} = \alpha^{74}$	(310201)	$f^{117} = \alpha^{117}$	(331032)
$f^{32} = \alpha^{32}$	(120322)	$f^{75} = \alpha^{75}$	(321020)	$f^{118} = \alpha^{118}$	(213103)
$f^{33} = \alpha^{33}$	(232032)	$f^{76} = \alpha^{76}$	(032102)	$f^{119} = \alpha^{119}$	(131310)
$f^{34} = \alpha^{34}$	(203203)	$f^{77} = \alpha^{77}$	(223210)	$f^{120} = \alpha^{120}$	(013131)
$f^{35} = \alpha^{35}$	(130320)	$f^{78} = \alpha^{78}$	(022321)	$f^{121} = \alpha^{121}$	(331313)
$f^{36} = \alpha^{36}$	(013032)	$f^{79} = \alpha^{79}$	(332232)	$f^2 = \alpha^{122}$	(103131)
$f^{37} = \alpha^{37}$	(221303)	$f^{80} = \alpha^{80}$	(213223)	$f^{123} = \alpha^{123}$	(300313)
$f^{38} = \alpha^{38}$	(132130)	$f^{81} = \alpha^{81}$	(131322)	$f^{124} = \alpha^{124}$	(100031)
$f^{39} = \alpha^{39}$	(0132213)	$f^{82} = \alpha^{82}$	(231132)	$f^{125} = \alpha^{125}$	(300003)
$f^{40} = \alpha^{40}$	(111321)	$f^{83} = \alpha^{83}$	(203313)	$f^{126} = \alpha^{126}$	(010000)
$f^{41} = \alpha^{41}$	(301132)	$f^{84} = \alpha^{84}$	(130331)		
$f^{42} = \alpha^{42}$	(210113)	$f^{85} = \alpha^{85}$	(303033)		

Fonte: Da autora.

Ao determinar a extensão do anel $\mathbb{Z}_4$, o elemento f^{126} é igual ao elemento f^1. Logo, f gera um grupo cíclico de ordem 126.

Passo 6 - Determinar o grupo das unidades

Do Passo 5, temos que f gera um grupo cíclico de ordem $n.d$ em $GR^*(4,6)$, com $d \geq 1 \in \mathbb{Z}$ e f^d gera um subgrupo cíclico cuja ordem é 63 em $GR^*(4,6)$. Como f gera um grupo de ordem $n \cdot d$, onde $n = 63$ (comprimento da sequência), temos que $n \cdot d = 63 \cdot d = 126$, implicando em $d = 2$.

Logo, $f^2 = (001000) = \alpha^2$ gera um subgrupo cíclico de ordem 63 em $GR^*(4,6)$, como apresentado na Tabela 12. Considere $\beta = \alpha^2$ o elemento primitivo que gera esse subgrupo cíclico. Esse elemento primitivo será utilizado na construção do código BCH de comprimento 63 sobre $\mathbb{Z}_4$.

Tabela 12 – Elementos de G_{63}

$GR^*(4,6)$	**Vetorial**	$GR^*(4,6)$	**Vetorial**	$GR^*(4,6)$	**Vetorial**
$\beta = \alpha^2$	(001000)	$\beta^{22} = \alpha^{44}$	(303101)	$\beta^{43} = \alpha^{86}$	(100303)
$\beta^2 = \alpha^4$	(000010)	$\beta^{23} = \alpha^{46}$	(032031)	$\beta^{44} = \alpha^{88}$	(012003)
$\beta^3 = \alpha^6$	(330000)	$\beta^{24} = \alpha^{48}$	(103320)	$\beta^{45} = \alpha^{90}$	(011120)
$\beta^4 = \alpha^8$	(003300)	$\beta^{25} = \alpha^{50}$	(221033)	$\beta^{46} = \alpha^{92}$	(220111)
$\beta^5 = \alpha^{10}$	(000033)	$\beta^{26} = \alpha^{52}$	(123210)	$\beta^{47} = \alpha^{94}$	(321201)
$\beta^6 = \alpha^{12}$	(121000)	$\beta^{27} = \alpha^{54}$	(331232)	$\beta^{48} = \alpha^{96}$	(032212)
$\beta^7 = \alpha^{14}$	(001210)	$\beta^{28} = \alpha^{56}$	(131312)	$\beta^{49} = \alpha^{98}$	(312322)
$\beta^8 = \alpha^{16}$	(330012)	$\beta^{29} = \alpha^{58}$	(313313)	$\beta^{50} = \alpha^{100}$	(201123)
$\beta^9 = \alpha^{18}$	(311300)	$\beta^{30} = \alpha^{60}$	(300133)	$\beta^{51} = \alpha^{102}$	(232011)
$\beta^{10} = \alpha^{20}$	(003113)	$\beta^{31} = \alpha^{62}$	(120001)	$\beta^{52} = \alpha^{104}$	(321330)
$\beta^{11} = \alpha^{22}$	(301031)	$\beta^{32} = \alpha^{64}$	(030200)	$\beta^{53} = \alpha^{106}$	(113213)
$\beta^{14} = \alpha^{24}$	(102010)	$\beta^{33} = \alpha^{66}$	(000302)	$\beta^{54} = \alpha^{108}$	(302132)
$\beta^{13} = \alpha^{26}$	(331020)	$\beta^{34} = \alpha^{68}$	(022003)	$\beta^{55} = \alpha^{110}$	(131021)
$\beta^{14} = \alpha^{28}$	(223310)	$\beta^{35} = \alpha^{70}$	(011220)	$\beta^{56} = \alpha^{112}$	(210310)
$\beta^{15} = \alpha^{30}$	(332233)	$\beta^{36} = \alpha^{72}$	(220112)	$\beta^{57} = \alpha^{114}$	(332103)
$\beta^{16} = \alpha^{32}$	(120322)	$\beta^{37} = \alpha^{74}$	(310201)	$\beta^{58} = \alpha^{116}$	(010321)
$\beta^{17} = \alpha^{34}$	(203203)	$\beta^{38} = \alpha^{76}$	(032102)	$\beta^{59} = \alpha^{118}$	(213103)
$\beta^{18} = \alpha^{36}$	(013032)	$\beta^{39} = \alpha^{78}$	(022321)	$\beta^{60} = \alpha^{120}$	(013131)
$\beta^{19} = \alpha^{38}$	(132130)	$\beta^{40} = \alpha^{80}$	(213223)	$\beta^{61} = \alpha^{122}$	(103131)
$\beta^{20} = \alpha^{40}$	(111321)	$\beta^{41} = \alpha^{82}$	(231132)	$\beta^{62} = \alpha^{124}$	(100031)
$\beta^{21} = \alpha^{42}$	(210113)	$\beta^{42} = \alpha^{84}$	(130331	$\beta^{63} = \alpha^{126}$	(010000)

Fonte: Da autora.

Passo 7 - Determinar o polinômio gerador da matriz G, $g(x)$

Depois de realizar o Passo 6, podemos construir códigos BCH de comprimento n sobre $GR(4,6)$. Considerando que a distância mínima de um código é menor ou igual ao comprimento da sequência, o algoritmo irá analisar todas as possibilidades para a distância mínima, $d_H \leq 2t+1$, que estão relacionadas com a capacidade de correção de erros. Assim, todos os valores de $1 \leq t \leq (n-1)/2$ serão examinados. Para cada valor de t, teremos um polinômio gerador $g(x)$ diferente e, consequentemente, um novo código.

O polinômio gerador do código BCH de comprimento n tem como raízes os elementos $\left\{(\beta^i), (\beta^i)^p, \ldots, (\beta^i)^{p^{r-1}(mod n)}\right\}$, e é dado por:

$$g(x) = mmc(M_1(x), M_2(x), \ldots, M_{2t}(x)),$$

em que $M_i(x)$ é o polinômio minimal associado ao elemento primitivo β^i, $i = 1, 2, \ldots, 2t$, e mmc denota o mínimo múltiplo comum.

No caso da palavra código que estamos considerando, a qual possui comprimento de $n = 63$ nucleotídeos, os valores de $1 \leq t \leq 31$ serão examinados. Para cada valor de t, teremos uma distância equivalente e seus referentes polinômios minimais envolvidos no cálculo dos polinômios geradores. O polinômio $g(x)$ é calculado por meio de três etapas:

1º) Cálculo das raízes dos polinômios minimais:

Para cada polinômio minimal $M_i(x) = M_i$, com $i = 1, 2, \ldots, 62$, temos:

$$\begin{aligned}
M_1(x) &= \left\{(\beta^1)(\beta^1)^2 \ldots, (\beta^1)^{2^{6-1}(\text{mod } 63)}\right\} \rightarrow M_1(x) = \left\{(\beta), (\beta^2), (\beta^4), (\beta^8), (\beta^{16}), (\beta^{32})\right\}, \\
M_2(x) &= \left\{(\beta^2), (\beta^2)^2, \ldots, (\beta^2)^{2^{6-1}(\text{mod } 63)}\right\} \rightarrow M_2(x) = \left\{(\beta^2), (\beta^4), (\beta^8), (\beta^{16}), (\beta^{32}), (\beta)\right\}, \\
\vdots &= \vdots \\
M_{62}(x) &= \left\{(\beta^{62}), (\beta^{62})^2, \ldots, (\beta^{62})^{2^{6-1}(\text{mod } 63)}\right\} \rightarrow M_{62}(x) = \left\{(\beta^{62}), (\beta^{61}), (\beta^{59}), (\beta^{55}), (\beta^{47}), (\beta^{31})\right\}.
\end{aligned}$$

2º) Cálculo dos polinômios minimais $M_i(x)$, para todo $i = 1, 2, \ddots, 62$:

Os polinômios minimais $M_i(x)$, para todo $i = 1, 2, \ldots, 62$, são calculados realizando-se o produtório das suas respectivas raízes minimais. Desse modo, o polinômio minimal $M_1(x)$ é obtido da seguinte maneira:

$$\begin{aligned}
M_1(x) &= \left\{(x-\beta)(x-\beta^2)(x-\beta^4)(x-\beta^8)(x-\beta^{16})(x-\beta^{32})\right\} \\
M_1(x) &= x^6 + 2x^3 + 3x + 1.
\end{aligned}$$

De maneira análoga, os outros polinômios minimais são calculados. No entanto, observe que não será preciso calcular os 62 polinômios minimais, pois alguns desses polinômios possuem as mesmas raízes e, consequentemente, são iguais.

3º) Cálculo dos polinômios geradores para $1 \leq t \leq 31$:

O polinômio gerador $g(x)$ para cada valor de t é dado pelo minímo múltiplo comum formado pelos polinômios minimais diferentes entre si.

Considerando que a distância mínima do código seja $d_H = 3$, então o polinômio gerador do código é dado por $g_1(x) = x^6+2x^3+3x+1$ que está relacionado com a matriz geradora G do código BCH sobre $\mathbb{Z}_4$, com parâmetros $(n, k, d_H) = (63, 57, 3)$. Da mesma forma, os demais polinômios geradores para as outras distâncias mínimas são determinados.

Passo 8 - Determinar o polinômio gerador da matriz H, $h(x)$

O polinômio gerador da matriz verificação de paridade H é obtido por meio da relação:

$$h(x) = \frac{x^n - 1}{g(x)} = \frac{x^{63} - 1}{x^6 + 2x^3 + 3x + 1},$$

$h(x) = x^{57} + 2x^{54} + x^{52} + 3x^{51} + x^{47} + 2x^{46} + x^{45} + 2x^{44} + 3x^{42} + x^{41} + 3x^{40} + 3x^{39} + x^{37} + 2x^{35} + 2x^{34} + x^{33} + x^{32} + 3x^{31} + 2x^{29} + x^{28} + 2x^{26} + 3x^{25} + 2x^{24} + 3x^{23} + x^{22} + 2x^{21} + 3x^{20} + x^{19} + x^{18} + 3x^{16} + 3x^{15} + 2x^{14} + 2x^{13} + x^{12} + 3x^{11} + x^9 + 2x^8 + 3x^7 + x^5 + 3x^4 + x^3 + 3x^2 + 3x + 3$, onde os coeficientes do polinômio $h(x)$ pertencem a $\mathbb{Z}_4$.

Passo 9 - Determinar a matriz G

Determinado o polinômio gerador no Passo 7, podemos construir a matriz geradora G relacionada ao mesmo. Seja o polinômio:

$$g(x) = g_0 + g_1x + g_2x^2 + \cdots + g_{r-1}x^{r-1} + x^r,$$

então a matriz geradora do código é dada por:

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & \cdots & g_{n-k-1} & g_{n-k} & 0 & 0 & \ldots & 0 \\ 0 & g_0 & g_1 & g_2 & \cdots & g_{n-k-1} & g_{n-k} & 0 & \ldots & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & \cdots & g_{n-k-1} & g_{n-k} & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & \ldots & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} \end{bmatrix}.$$

Considere o polinômio gerador $g_1(x) = x^6+2x^3+3x+1$. Realizando os deslocamentos dos coeficientes do polinômio $g(x)$ da esquerda para a direita, obtemos a matriz geradora G com dimensão 57×63 :

$$G = \begin{bmatrix} 130200100000000000000000000000000000000000000000000000000000000 \\ 013020010000000000000000000000000000000000000000000000000000000 \\ 001302001000000000000000000000000000000000000000000000000000000 \\ 000130200100000000000000000000000000000000000000000000000000000 \\ \vdots \\ 000000000000000000000000000000000000000000000000000001302001000 \\ 000000000000000000000000000000000000000000000000000000130200100 \\ 000000000000000000000000000000000000000000000000000000013020010 \\ 000000000000000000000000000000000000000000000000000000001302001 \end{bmatrix}.$$

Passo 10 - Determinar a matriz H e a sua transposta H^T

Dado o polinômio de verificação de paridade $h(x)$, $h(x) = h_0 + h_1x + \cdots + h_kx^k$, temos a matriz H:

$$H = \begin{bmatrix} h_k & h_{k-1} & h_{k-2} & \dots & h_0 & 0 & 0 & 0 & \dots & 0 \\ 0 & h_k & h_{k-1} & h_{k-2} & \dots & h_0 & 0 & 0 & \dots & 0 \\ 0 & 0 & h_k & h_{k-1} & h_{k-2} & \dots & h_0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & h_k & h_{k-1} & \dots & h_1 & h_0 \end{bmatrix}.$$

Determinado o polinômio $h(x)$ no **Passo 8**, realizamos os deslocamentos dos coeficientes do polinômio gerador $h(x)$ da direita para a esquerda e obtemos a matriz geradora H com dimensão 6×63 :

$$H = \begin{bmatrix} 100201300012120313301022113021023231231103322130123013133300000 \\ 010020130001212031330102211302102323123110332213012301313330000 \\ 001002013000121203133010221130210232312311033221301230131333000 \\ 000100201300012120313301022113021023231231103322130123013133300 \\ 000010020130001212031330102211302102323123110332213012301313330 \\ 000001002013000121203133010221130210232312311033221301230131333 \end{bmatrix}.$$

A matriz H^T de dimensão 63×6 é obtida realizando-se a troca dos elementos das linhas pelos elementos das colunas.

Passo 11 - Rotular a sequência de DNA utilizando o Passo 1

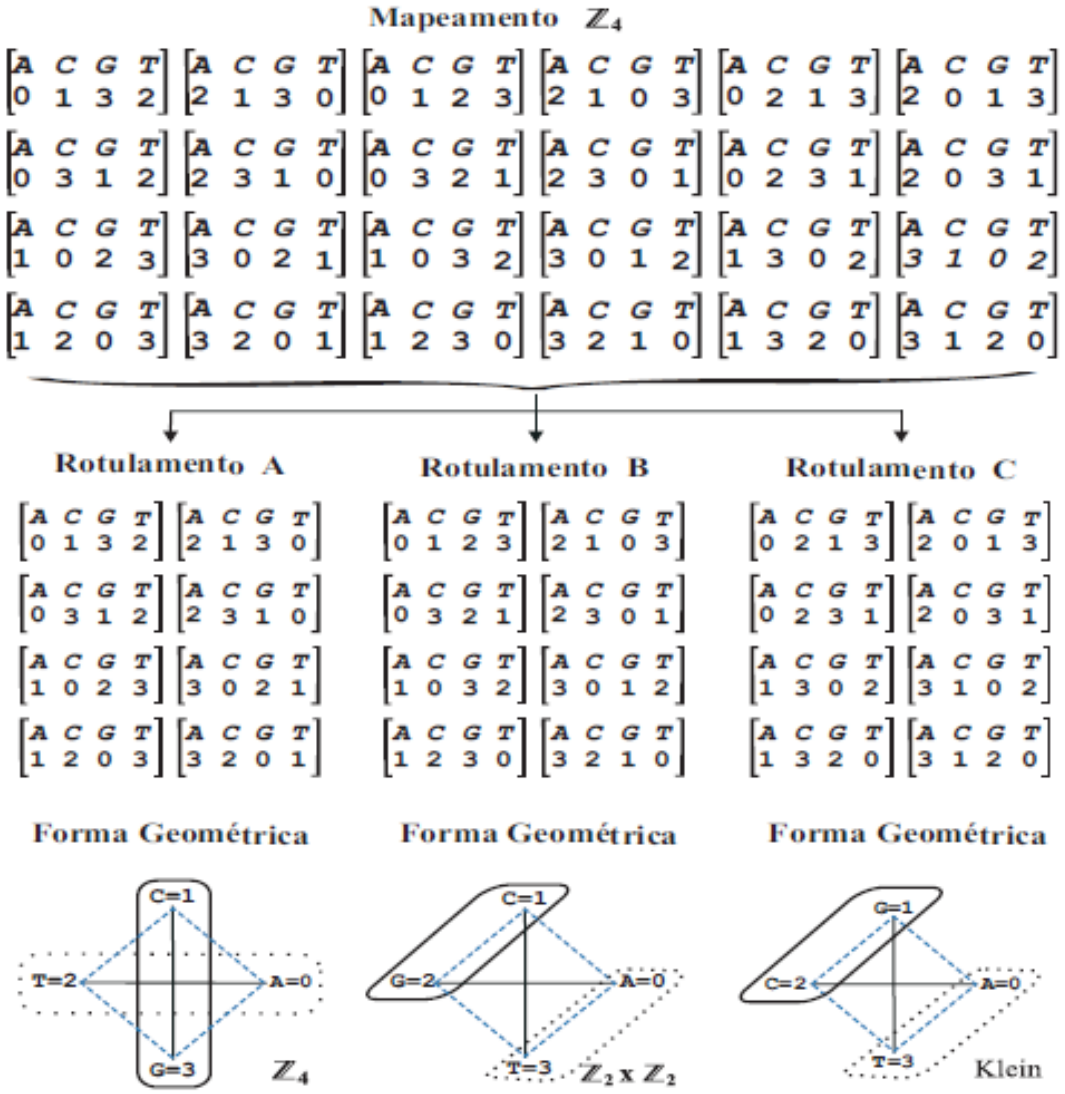
Neste exemplo, analisaremos se o código BCH primitivo sobre $GR(4, 6)$ é capaz de reproduzir a sequência de direcionamento (SD) de uma proteína mitocondrial.

No Passo 1 especificamos o alfabeto utilizado no código, em que foi determinada uma analogia entre o alfabeto 4-ário do conjunto de nucleotídeos denotado por $N = \{A, C, G, T/U\}$ e o alfabeto 4-ário matemático, denotado por $\mathbb{Z}_4 = \{0, 1, 2, 3\}$. Essa analogia é necessária para realizar o mapeamento entre o conjunto $N = \{A, C, G, T/U\}$ e o conjunto $\mathbb{Z}_4 = \{0, 1, 2, 3\}$ ou vice-versa. No entanto, o melhor mapeamento entre esses dois conjuntos ainda é desconhecido, dessa maneira todas as possibilidades de associação dos elementos serão analisadas para verificar qual delas é a melhor.

Assim, toda sequência de DNA será considerada como uma das 24 permutações entre $N \to \mathbb{Z}_4$. Essas permutações podem ser divididas em três grupos contendo oito permutações cada. Esses grupos são chamados de Rotulamento A, Rotulamento B e Rotulamento C, como apresentado na Figura 10.

Para analisarmos estes rotulamentos, é necessário considerar a complementaridade das bases nitrogenadas, de modo que, a adenina (A) se liga com a timina (T) ou a uracila (U) (ou vice-versa) e a guanina (G) se liga a citosina (C) (ou vice-versa). No caso do Rotulamento A, para que qualquer um dos nucleotídeos alcance o seu complementar é necessário caminhar duas arestas, enquanto que no outros rotulamentos basta deslocar em uma aresta. Todas as permutações contidas no rotulamento A caracterizam o mapeamento em $\mathbb{Z}_4$ - linear, ou seja, toda sequência de DNA reproduzida por esse código será não linear,

Figura 10 – Rotulamentos A, B e C



Fonte: ROCHA, 2010, p. 74.

pois o mapeamento $\mathbb{Z}_4$ - linear é não linear, enquanto que o código BCH sobre estrutura de anel é linear. Já para os mapeamentos $\mathbb{Z}_2 \times \mathbb{Z}_2$ e $Klein$ - linear, são lineares e o código BCH também é linear, então toda sequência de DNA reproduzida por eles também será linear.

Seja a sequência de direcionamento do $NCBI$, a qual analisaremos, igual a:

GCCGTTCATGTTTACTCTGGGTTGCCTTGGTGGGGAACTATCGCGGCCACCACCATCCTCATT

As 24 linhas da matriz P correspondem às 24 permutações da sequência de direciona-

mento.

$$P = \begin{bmatrix}
211233103233301313222332113322322220013031212211011011031131033\\
311322102322201212333223112233233330012021313311011011021121022\\
122133203133302323111331223311311110023032121122022022032232033\\
322311201311102121333113221133133330021012323322022022012212011\\
233211301211103131222112331122122220031013232233033033013313011\\
133122302122203232111221332211211110032023131133033033023323022\\
200233013233310303222332003322322221103130202200100100130030133\\
300322012322210202333223002233233331102120303300100100120020122\\
022033213033312323000330223300300001123132020022122122132232133\\
322300210300012020333003220033033331120102323322122122102202100\\
033022312022213232000220332200200001132123030033133133123323122\\
233200310200013030222002330022022221130103232233133133103303100\\
100133023133320303111331003311311112203230101100200200230030233\\
300311021311120101333113001133133332201210303300200200210010211\\
011033123033321313000330113300300002213231010011211211231131233\\
311300120300021010333003110033033332210201313311211211201101200\\
033011321011123131000110331100100002231213030033233233213313211\\
133100320100023030111001330011011112230203131133233233203303200\\
100122032122230202111221002211211113302320101100300300320020322\\
200211031211130101222112001122122223301310202200300300310010311\\
011022132022231212000220112200200003312321010011311311321121322\\
211200130200031010222002110022022223310301212211311311301101300\\
022011231011132121000110221100100003321312020022322322312212311\\
122100230100032020111001220011011113320302121122322322302202300
\end{bmatrix}.$$

No Quadro 3 mostramos como as linhas da matriz P estão relacionadas com as 24 permutações $N \to \mathbb{Z}_4$, onde cada uma das 24 permutações foi definida como um caso.

Quadro 3 – Relação entre as linhas da matriz P e as 24 permutações

Linha = Caso	$N \to \mathbb{Z}_4$	**Linha = Caso**	$N \to \mathbb{Z}_4$
L 1 = Caso 01	(A, C, G, T) = (0, 1, 2, 3)	L 13 = Caso 13	(A, C, G, T) = (2, 0, 1, 3)
L 2 = Caso 02	(A, C, G, T) = (0, 1, 3, 2)	L 14 = Caso 14	(A, C, G, T) = (2, 0, 3, 1)
L 3 = Caso 03	(A, C, G, T) = (0, 2, 1, 3)	L 15 = Caso 15	(A, C, G, T) = (2, 1, 0, 3)
L 4 = Caso 04	(A, C, G, T) = (0, 2, 3, 1)	L 16 = Caso 16	(A, C, G, T) = (2, 1, 3, 0)
L 5 = Caso 05	(A, C, G, T) = (0, 3, 2, 1)	L 17 = Caso 17	(A, C, G, T) = (2, 3, 0, 1)
L 6 = Caso 06	(A, C, G, T) = (0, 3, 1, 2)	L 18 = Caso 18	(A, C, G, T) = (2, 3, 1, 0)
L 7 = Caso 07	(A, C, G, T) = (1, 0, 2, 3)	L 19 = Caso 19	(A, C, G, T) = (3, 0, 1, 2)
L 8 = Caso 08	(A, C, G, T) = (1, 0, 3, 2)	L 20 = Caso 20	(A, C, G, T) = (3, 0, 2, 1)
L 9 = Caso 09	(A, C, G, T) = (1, 2, 0, 3)	L 21 = Caso 21	(A, C, G, T) = (3, 1, 0, 2)
L 10 = Caso 10	(A, C, G, T) = (1, 2, 3, 0)	L 22 = Caso 22	(A, C, G, T) = (3, 1, 2, 0)
L 11 = Caso 11	(A, C, G, T) = (1, 3, 0, 2)	L 23 = Caso 23	(A, C, G, T) = (3, 2, 0, 1)
L 12 = Caso 12	(A, C, G, T) = (1, 3, 2, 0)	L 24 = Caso 24	(A, C, G, T) = (3, 2, 1, 0)

Fonte: Da autora.

Passo 12 -Verificar se a sequência de DNA é palavra-código de acordo com os padrões de erros estabelecidos: D(a, b) = 0, D(a, b) = 1 e D(a, b) = 2

O procedimento usado para determinar quais das sequências são palavras-código dos códigos $(63, k, d_H)$ é o seguinte:

a) para analisarmos as sequências de DNA com até 1 nucleotídeo de diferença da sequência de DNA do $NCBI$, $D(a, b) = 1$, consideramos as 4356 possíveis palavras código para cada sequência de DNA analisada e então usamos a relação $v \cdot H^T = 0$. As palavras código encontradas são armazenadas. Essa quantidade de possíveis palavras código foi obtida considerando as sequências de DNA diferindo em um nucleotídeo, analisando as 3 outras possibilidades de nucleotídeos em cada posição na sequência para as 24 permutações.

b) já para analisarmos as sequências de DNA com até 2 nucleotídeos de diferença da sequência de DNA do $NCBI$, $D(a,b) = 2$, consideramos todas as combinações 2 a 2 dos n nucleotídeos de comprimento da sequência para as 24 permutações, resultando em 17577 possíveis palavras código para cada sequência de DNA analisada e, então usamos a relação $v \cdot H^T = 0$. As palavras código encontradas são armazenadas.

Passo 13 - Comparar todas as palavras código armazenadas no Passo 12 com a sequência de DNA original e mostrar onde os erros ocorreram

Neste passo, todas as palavras código armazenadas no passo anterior estão rotuladas na forma alfabeto do código $\mathbb{Z}_4 = \{0, 1, 2, 3\}$, e serão convertidas em nucleotídeos usando o alfabeto do código genético $N = \{A, C, G, T\}$. Em seguida, as palavras código são comparadas, uma a uma, com a sequência de DNA original, mostrando onde os nucleotídeos diferem.

Passo 14 - Voltar para o Passo 7 e determinar outro $g(x)$

Neste passo, determinamos outro valor da distância mínima, e utilizamos o mesmo procedimento, apresentado no Passo 7, para calcular o polinômio gerador relativo a esta distância.

Passo 15 - Repetir os Passos 8 ao 12 para o $g(x)$ obtido no Passo 14, até que se esgote todas as possibilidades de $g(x)$

Neste passo, o algoritmo determina todas as palavras código encontradas com 0, 1 e 2 nucleotídeos de diferença, com todos os polinômios geradores.

Passo 16 - Voltar para o Passo 3 e escolher outro $p(x)$ e então, repetir os Passos 4 ao 14 até esgotar todos os $p(x)$ do Passo 3

Passo 17 - Fim.

Como resultado para o algoritmo de geração de sequência de DNA para $D(a,b) = 1$ nucleotídeo de diferença da sequência do $NCBI$, foram obtidos 8 palavras códigos, pertencentes ao Rotulamento C. Essas palavras código eram diferentes no alfabeto do código $\mathbb{Z}_4 = \{0, 1, 2, 3\}$, no entanto são iguais quando realizado o rotulamento para o alfabeto do código genético $N = \{A, C, G, T/U\}$. Assim, tem-se apenas uma única sequência de DNA gerada pelo algoritmo, a ser analisada para $D(a,b) = 1$, como mostra a Figura 11.

A Figura 11 mostra a sequência reproduzida pelo código Klein-linearidade com parâmetros (63,57,3), por meio do polinômio primitivo $p_1(x) = x^6 + x + 1$ e do polinômio gerador $g_1(x) = x^6 + 2x^3 + 3x + 1$. Pode-se observar que na posição da décima trinca ocorreu uma troca de nucletídeo ocasionando a troca do aminoácido Triptofano (W) pelo aminoácido Glicina (G).

Como resultado do algoritmo de geração da sequência de DNA para $D(a,b) = 2$ nucleotídeos de diferença da sequência do $NCBI$, foram obtidos 5 palavras código em

Figura 11 – Sequência de DNA com 63 nucleotídeos e $D(a,b) = 1$

Seq.36 | *S. cerevisiae* - **OXA 1 – Sinal interno – GI número 832917**

Código klein-linearidade((63,57,3) BCH primitivo sobre *GR*(4,6), rotulamento C)

$p_1(x)=x^6+x+1$ - $g_1(x)=x^6+2x^3+3x+1$ - Caso 3-(A,C,G,T)=(0,2,1,3)

```
aaO:  A   V   H   V   Y   S   G   L   P   W   W   G   T   I   A   A   T   T   I   L   I
ntO: GCC GTT CAT GTT TAC TCT GGG TTG CCT TGG TGG GGA ACT ATC GCG GCC ACC ACC ATC CTC ATT
RtO: 122 133 203 133 302 323 111 331 223 311 311 110 023 032 121 122 022 022 032 232 033
RtG: 122 133 203 133 302 323 111 331 223 111 311 110 023 032 121 122 022 022 032 232 033
ntG: GCC GTT CAT GTT TAC TCT GGG TTG CCT GGG TGG GGA ACT ATC GCG GCC ACC ACC ATC CTC ATT
aaG:  A   V   H   V   Y   S   G   L   P   G   W   G   T   I   A   A   T   T   I   L   I
```

Fonte: ROCHA, 2010, p. 157.

cada caso de permutação para os Rotulamentos A e B, e 93 palavras código em cada caso de permutação no Rotulamento C. Essas palavras código são diferentes nos 24 casos de permutações para o alfabeto do código $\mathbb{Z}_4 = \{0, 1, 2, 3\}$, mas quando rotuladas usando o alfabeto $= \{A, C, G, T/U\}$, as palavras códigos são iguais nos 8 casos dos Rotulamentos A, B e C, respectivamente.

Figura 12 – Sequência de DNA com $n = 63$ nucleotídeos e $D(a,b) = 2$

Código $\mathbb{Z}_4$-linearidade((63,57,3) BCH primitivo sobre *GR*(4,6), rotulamento A)

$p_1(x)=x^6+x+1$ - $g_1(x)=x^6+2x^3+3x+1$ - Caso 2-(A,C,G,T)=(0,1,3,2)

```
aaO:  A   V   H   V   Y   S   G   L   P   W   W   G   T   I   A   A   T   T   I   L   I
ntO: GCC GTT CAT GTT TAC TCT GGG TTG CCT TGG TGG GGA ACT ATC GCG GCC ACC ACC ATC CTC ATT
RtO: 211 233 103 233 301 313 222 332 113 322 322 220 013 031 212 211 011 011 031 131 033
RtG: 211 233 103 233 311 333 222 332 113 322 322 220 013 031 212 211 011 011 031 131 033
ntG: GCC GTT CAT GTT TCC TTT GGG TTG CCT TGG TGG GGA ACT ATC GCG GCC ACC ACC ATC CTC ATT
aaG:  A   V   H   V   S   F   G   L   P   W   W   G   T   I   A   A   T   T   I   L   I
```

Código $\mathbb{Z}_2 \times \mathbb{Z}_2$-linearidade((63,57,3) BCH primitivo sobre *GR*(4,6), rotulamento B)

$p_1(x)=x^6+x+1$ - $g_1(x)=x^6+2x^3+3x+1$ - Caso 1-(A,C,G,T)=(0,1,2,3)

```
aaO:  A   V   H   V   Y   S   G   L   P   W   W   G   T   I   A   A   T   T   I   L   I
ntO: GCC GTT CAT GTT TAC TCT GGG TTG CCT TGG TGG GGA ACT ATC GCG GCC ACC ACC ATC CTC ATT
RtO: 311 322 102 322 201 212 333 223 112 233 233 330 012 021 313 311 011 011 021 121 022
RtG: 311 321 102 322 201 212 333 223 112 233 233 330 012 021 313 311 011 010 021 121 022
ntG: GCC GTC CAT GTT TAC TCT GGG TTG CCT TGG TGG GGA ACT ATC GCG GCC ACC ACA ATC CTC ATT
aaG:  A   V   H   V   Y   S   G   L   P   W   W   G   T   I   A   A   T   T   I   L   I
```

Código Klein-linearidade((63,57,3) BCH primitivo sobre *GR*(4,6), rotulamento C)

$p_1(x)=x^6+x+1$ - $g_1(x)=x^6+2x^3+3x+1$ - Caso 3-(A,C,G,T)=(0,2,1,3)

```
aaO:  A   V   H   V   Y   S   G   L   P   W   W   G   T   I   A   A   T   T   I   L   I
ntO: GCC GTT CAT GTT TAC TCT GGG TTG CCT TGG TGG GGA ACT ATC GCG GCC ACC ACC ATC CTC ATT
RtO: 122 133 203 133 302 323 111 331 223 311 311 110 023 032 121 122 022 022 032 232 033
RtG: 122 133 203 131 302 323 111 331 223 311 311 110 023 032 123 122 022 022 032 232 033
ntG: GCC GTT CAT GTG TAC TCT GGG TTG CCT TGG TGG GGA ACT ATC GCT GCC ACC ACC ATC CTC ATT
aaG:  A   V   H   V   Y   S   G   L   P   W   W   G   T   I   A   A   T   T   I   L   I
```

Fonte: ROCHA, 2010, p. 159.

A Figura 12 mostra a sequência reproduzida pelos códigos: $\mathbb{Z}_4$-linearidade ((63,57,3) BCH primitivo sobre $GR(4,6)$, rotulamento A); $\mathbb{Z}_2 \times \mathbb{Z}_2$-linearidade ((63,57,3) BCH primitivo sobre $GR(4,6)$, rotulamento B) e Klein-linearidade ((63,57,3) BCH primitivo sobre $GR(4,6)$, rotulamento C), por meio do polinômio primitivo $p_1(x) = x^6 + x + 1$ e do polinômio gerador $g_1(x) = x^6 + 2x^3 + 3x + 1$ com D(a, b) = 2 nucleotídeos diferindo da

sequência do NCBI.

Nas posições das trincas cinco e seis da sequência reproduzida no rotulamento A foram alterados, ocasionando a troca dos aminoácidos nestas posições. No entanto, as mudanças nas posições dois e dezoito da sequência reproduzida no rotulamento B, bem como as alterações de nucleotídeos nas posições quatro e quinze da sequência reproduzida no rotulamento C, não ocasionaram as trocas dos aminoácidos nessas trincas.

3.3 ANÁLISE DAS SEQUÊNCIAS REPRODUZIDAS

O exemplo exposto na Seção 3.2, referente a reprodução de uma proteína mitocondrial foi apenas uma das 92 sequências reproduzidas e analisadas por Rocha (2010) e Faria (2011) por meio do algoritmo de geração de proteínas. Dentre as sequências reproduzidas pelos códigos BCH estão sequências com os seguintes comprimentos: 21, 39, 45, 51, 63, 93, 105, 195, 255, 511, 1023 e 2047 nucleotídeos e com as seguintes características biológicas: sequências de direcionamento, sequências de direcionamento ambígua, enzima, sinal interno, hormônio, íntron, DNA repetitivo, miRNA, proteínas de vírus e de bactérias, proteínas, gene e genoma procariótico. Com isso, foi possível verificar alguns padrões existentes na reprodução dessas sequências, além de averiguar padrões biológicos que se mantiveram no algoritmo de geração de proteínas.

Na reprodução das sequências foi possível verificar que todas elas possuem uma estrutura matemática, e portanto, podem ser identificadas e classificadas através de códigos corretores de erros. Um outro ponto observado, foi de que na reprodução de sequências de DNA por códigos corretores de erros, apesar dos códigos BCH serem construídos em todas as distâncias mínimas, somente alguns dos códigos com $d_H = 3$ foram capazes de reproduzir as sequências correspondentes, implicando assim, que o grau do polinômio primitivo e do polinômio gerador devem ser iguais. Como consequência desse fato, a redundância está associada com o grau desses polinômios. Então, uma pequena redundância resulta em um código de taxa alta.

Já em relação a quantidade de palavras códigos reproduzidas pelos rotulamentos A, B e C para $D(a, b) = 1$ e $D(a, b) = 2$, observou-se que no caso das palavras códigos reproduzidas pelos códigos BCH primitivos com $D(a, b) = 1$, para cada um dos 3 rotulamentos existem 8 palavras códigos diferentes, no entanto, quando empregado o rotulamento genético essas 8 palavras códigos são iguais em termos dos nucleotídeos. E para $D(a, b) = 2$, sempre existem várias palavras código para cada uma das 24 permutações e quando aplicado, nessas sequências, o rotulamento recíproco do código genético o resultado final são várias sequências de DNA reproduzidas nos rotulamentos A, B e C.

Sob o ponto de visto algébrico, em sistema de comunicação digital, a construção de um código corretor de erro não depende do polinômio primitivo do grau r usado na extensão de Galois. No entanto, na reprodução das sequências de DNA verificou-se que

existe uma dependência na existência de alguns códigos corretores de erros com alguns polinômios primitivos. Para $D(a, b) = 1$ apenas alguns polinômios primitivos associados a alguns rotulamentos foram capazes de reproduzir algumas sequências de DNA, assim, a escolha da estrutura algébrica, o alfabeto, o rotulamento, o mapeamento, o polinômio primitivo e o polinômio gerador são fundamentais. Já para $D(a, b) = 2$ não exite relação de dependência entre os códigos corretores de erros e os polinômios primitivos, pois não importa o polinômio primitivo de grau r a ser usado no processo de geração do código, todo e qualquer polinômio primitivo será capaz de identificar e reproduzir as sequências de DNA.

Além disso, alguns aspectos biológicos puderam ser observados nas sequências de DNA geradas pelo algoritmo de geração de proteínas. Apesar de no Passo 12, cada uma das posições do códon foram consideradas com probabilidades iguais de terem erros, a maioria das sequências de DNA apresentaram trocas de nucleotídeos na primeira posição ou na terceira posição. Isto infere então, que a segunda posição foi mais protegida contra erros durante a reprodução das sequências de DNA pelos códigos BCH, o que faz sentido biologicamente, já que a troca de nucleotídeo nessa posição acarreta na troca de aminoácido e, consequentemente, há uma maior possibilidade de suceder em uma mutação não silenciosa.

4 ANÁLISE MUTACIONAL DA ENZIMA MITOCONDRIAL ATP6 POR MEIO DO ALGORITMO DE GERAÇÃO DE PROTEÍNAS

Neste capítulo serão apresentadas a reprodução e a análise da sequência de DNA relacionada à enzima mitocondrial ATP6, por meio do algoritmo de geração de proteínas.

Segundo Rocha (2010), existe um grande número de sequências de DNA que podem ser geradas por esse algoritmo. Contudo, há uma dificuldade em encontrar sequências que atendam às suas restrições. Após procurar uma sequência de nucleotídeos relacionada a patologias importantes, a enzima mitocondrial ATP6 foi escolhida para ser reproduzida. Essa escolha se deve ao fato dela possuir comprimento de $n = 63$ nucleotídeos, sendo este o comprimento mínimo que atende a restrição e, biologicamente, mutações no ATP6 podem estar associadas a doença de Leber, a doença de NARP e a necrose estriada bilateral.

Apesar de algumas doenças mitocondriais estarem relacionadas com mutações no ATP6, os mecanismos pelos quais as alterações nessa proteína interferem na síntese do ATP ainda precisam ser elucidados (DUNO *et al.*, 2013; SENIOR; WEBER, 2003). Como o algoritmo de geração de proteínas possibilita a realização de estudos mutacionais, futuramente, esses estudos poderão contribuir para uma melhor compreensão dos processos naturais que envolvem doenças mitocondriais na proteína ATP6. Com isso, consideramos de vital importância a análise dessa sequência por meio dos códigos corretores de erros.

Na Seção 4.1 será apresentada uma breve introdução sobre as mitocôndrias e a sua importância para as atividades metabólicas que ocorrem dentro das células. Além disso, apresentamos o complexo proteico ATP sintase, responsável pela sintetização da maior parte de ATP, e em especial a subunidade proteica ATP6. Na Seção 4.3 foi gerada e reproduzida a sequência de DNA relacionada a proteína ATP6, por meio do algoritmo de geração de proteínas. Nessa Seção apresentamos alguns passos da execução do algoritmo para essa sequência. Na Seção 4.4 foram analisados os resultados obtidos nas simulações da Seção 4.3.

Os conceitos apresentados a seguir podem ser encontrados em Afonso *et al.* (2006), Ballmoos; Dimroth e Meier (2006), Blanco-Grau *et al.* (2013), Duno *et al.* (2013), Komulainen *et al.* (2016), Lyra *et al.* (2006), Nasser *et al.* (2001), Sgarbi *et al.* (2006), Senior e Weber (2003) e Tuset *et al.* (2006).

4.1 MITOCÔNDRIAS

A mitocôndria é uma das mais importantes organelas celulares e está presente em todas as células eucarióticas dos animais e plantas, e também em alguns micro-organismos. A principal função atribuída à mitocôndria é a de transformar energia química encontrada no citoplasma em energia acessível à célula. Essa energia é acumulada principalmente em

forma de ATP (adenina trifosfato), que será utilizada para atividades que ocorrem dentro da célula.

Estruturalmente, essas organelas possuem formato esférico ou alongado e são envoltas por duas membranas: uma mais interna e outra mais externa. A membrana externa é semelhante à de outras organelas, ou seja, possui textura lisa e é composta por lipídeos e proteínas, que controlam a entrada de moléculas. E a membrana interna apresenta numerosas dobras, chamadas de cristas mitocondriais.

Juntas, essas duas membranas definem dois compartimentos mitocondriais separados: o espaço intermembranoso e a matriz mitocondrial. O espaço intermembranoso está localizado na região entre a membrana interna e a membrana externa e possui permeabilidade a íons e a outras moléculas menores. A matriz mitocondrial é o espaço situado dentro das duas membranas e é preenchida por uma substância viscosa onde se encontram as enzimas respiratórias que participam da respiração celular.

A respiração celular é um processo que tem como objetivo a geração de energia em forma de ATP. Esse processo é dividido em três etapas: glicólise, Ciclo de Krebs e cadeia respiratória. A primeira etapa acontece no citoplasma das células, já as outras duas etapas são realizadas dentro da mitocôndria de forma separada, isto é, o Ciclo de Krebs é realizado na matriz mitocondrial e a cadeia respiratória ocorre nas cristas mitocondriais.

A glicólise é o processo em que a glicose é quebrada em moléculas menores de ácido pirúvico ou piruvato, liberando energia. Após a glicólise, inicia-se a segunda etapa da respiração celular denominada de reações do Ciclo de Krebs em que é promovida a degradação de produtos finais dos carboidratos, lipídeos e de aminoácidos. A última etapa da respiração celular, denominada por cadeia respiratória, é responsável pela maior parte do ATP produzido.

A cadeia respiratória mitocondrial é composta por cinco complexos enzimáticos: Complexo I (NADH coenzima Q oxido-redutase), Complexo II (succinato-ubiquinona oxirredutase), Complexo III (ubiquinona-citocromo-c oxirredutase), Complexo IV (citocromo-c oxidase- COX) e o Complexo V (ATP sintase). O Complexo I oxida os elétrons do nicotinamina adenina dimunucleotídeo reduzida (NADH) gerados na glicólise e no Ciclo de Krebs e, os transfere para a ubiquinona, que é um transportador de elétrons e prótons. O Complexo II catalisa a oxidação de elétrons do succinato, levando elétrons também à ubiquinona. O Complexo III recebe elétrons do ubiquinona e os transfere para o Complexo IV que realiza a redução de moléculas de O_2 para H_2O. Depois disso, esse fluxo de prótons entra no Complexo V, e ao passarem por esse complexo, os prótons são capazes de mudar sua conformação, produzindo moléculas ATP, de ADP e fosfato.

Esse mecanismo de produção de ATP também é chamado de fosforilação oxidativa. Assim, esses cinco complexos devem funcionar em perfeita harmonia para que a geração de energia celular supra a demanda de energia dos mais diversos órgãos e tecidos do corpo. A quantidade de enzimas mitocondriais em cada órgão é variável, sendo geralmente maior

naquelas com maior atividade metabólica. Os tecidos que mais demandam requerimento energético são os do cérebro, músculo esquelético e o cardíaco.

Em 1966, foi descoberto por Van Bruggen, Sinclair e Stevens Nass que as mitocôndrias possuem sua própria carga genética, denominado como DNA mitocondrial (mtDNA). No entanto, foi apenas em 1981 que Anderson et. al sequenciou totalmente o genoma mitocondrial. Com isso, foi possível identificar alguns padrões gerais no mtDNA, como por exemplo, que os principais rRNAs são sempre codificados pelo genoma mitocondrial e que o mtDNA é de herança materna, pois durante a fecundação as mitocôndrias do espermatozoide são degradadas, restando apenas as do óvulo. Além disso, verificou-se também que existem diferenças entre o mtDNA das leveduras e o mtDNA dos mamíferos.

Nos mamíferos, o mtDNA é uma molécula compacta de formato circular fechado, sendo que a região codificante é constituída de 15.447 bases pareadas e 37 genes codificadores de 13 subunidades proteicas, de 22 RNA transportadores e de dois genes para o RNA ribossômico. Já a região não codificante denominada de *D-Loop*, apresenta aproximadamente 1.122 bases pareadas. A região *D-Loop* apresenta grande variabilidade entre os indivíduos e por ser a região onde se inicia o processo de duplicação do mtDNA, é a região mais suscetível de ocorrer mutações. Além disso, se compararmos o mtDNA com o ácido desoxirribonucleico nuclear (nDNA), o primeiro apresenta maior probabilidade de acontecer mudança nas bases, pois não possui enzima reparadora da DNA polimerase mitocondrial.

Cada célula do corpo possui de 5 a 10 mitocôndrias, cada uma com o seu próprio material genético. Assim, quando existe uma mutação no mtDNA, a célula pode apresentar 100% de mtDNA mutado, 100% de mtDNA normal ou a mistura de mtDNA mutado e mtDNA normal. A mutação do mtDNA gera um defeito na produção de energia, havendo uma grande probabilidade de ocorrer um comprometimento no funcionamento do órgão que está presente. No entanto, o que define se o tecido ou célula apresentará deficiência em sua função é a proporção de mutante e o limiar do tecido ou célula. A função anormal de proteínas acarretada por mutações no mtDNA podem resultar em doenças mitocondriais que comprometem principalmente os processos de fosforilação oxidativa, reduzindo assim a taxa de produção de ATP.

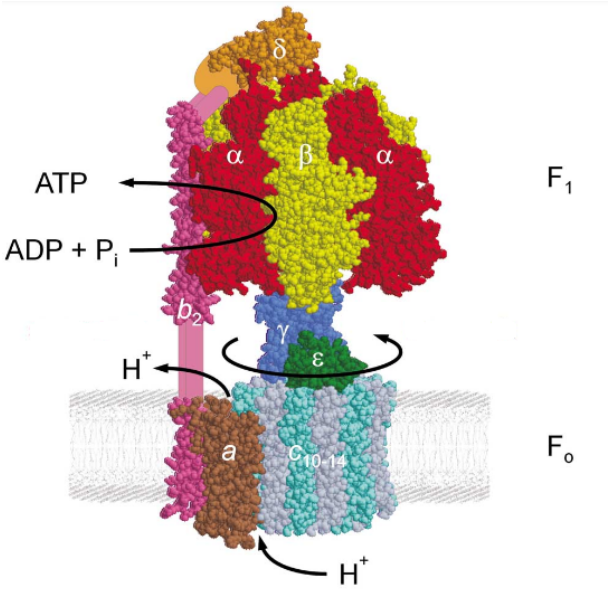
O estudo da mitocôndria bem como do DNA mitocondrial, atualmente, possui grande aplicabilidade no contexto científico, pois estão relacionados a variados processos que acontecem nas células, como o envelhecimento celular e a deterioração programada de células, além da relação de defeitos no mtDNA com doenças em órgãos que exijam alta capacidade energética e em doenças metabólicas.

4.2 ATP SINTASE E ATP6

O complexo V, também conhecido como ATP sintase, é a enzima final da fosforilação oxidativa responsável por sintetizar ATP. A ATP sintase é um complexo proteico de grandes dimensões, sendo composta por duas regiões rotativas, sendo cada uma delas constituída de várias cadeias polipeptídicas.

A primeira região conhecida como fator de acoplamento 1, F_1, é uma porção esférica e hidrofílica, responsável pela síntese de ATP e localizada em associação com a face matricial da membrana interna mitocondrial. A segunda região, F_o possui característica hidrofóbica e é um canal localizado dentro da membrana mitocondrial por onde passa o fluxo de prótons. Para que a enzima de ATP sintase funcione, as regiões F_1 e F_o devem estar adequadamente associadas. Essa estrutura pode ser vista na Figura 13.

Figura 13 – Estrutura do ATP sintase



Fonte: SENIOR; WEBER, 2003, p. 3.

A região F_1 é constituída das subunidades $\alpha, \gamma, \beta, \delta$ e ϵ. Já a região F_o é composta pelas subunidades a, b e c. As subunidades α e β de F_1 contém o núcleo catalítico que sintetiza o ATP. Já a subunidade γ de F_1 consiste em um eixo proteico que penetra nas subunidades α e β. A subunidade δ faz parte do eixo periférico que sustenta o centro catalítico e liga a subunidade a da região F_o com a região F_1, e a subunidade ϵ forma os talos do centro catalítico. A subunidade c possui forma de anel e é o compartimento onde estão presentes os prótons. A subunidade a é um conjunto de quatro hélices mais uma hélice transmembranar, onde essa subunidade se associa a duas hélices transmembranares da subunidade b que conecta F_o às subunidades α e β de F_1.

Conforme os prótons atravessam a membrana por meio da ATP sintase, F_o entra em movimento de rotação. O anel c é cercado pelas subunidades a e b e interage com a

subunidade γ, formando o conjunto do motor que gira os prótons. Esta rotação é causada por mudanças no estado de ionização de aminoácidos na subunidade c. Essa subunidade entra em movimento de rotação, e por sua vez, força a rotação da subunidade γ. É o movimento da subunidade γ que providencia a energia necessária para que as subunidades α e β sofram modificações, e produzam ATP.

As subunidades a e c da região F_o também conhecidas como ATP6 (ou MT-ATP6) e ATP8 (ou MT-ATP8), são codificadas pelo mtDNA, enquanto que as outras subunidades são codificadas pelo nDNA. Uma característica incomum na proteína MT-ATP6 é a sobreposição de 46 nucleotídeos de seus primeiros códons com o final da proteína MT-ATP8. Nos seres humanos, mutações na subunidade ATP6 geram distúrbios complexos com expressão e gravidade diferenciadas, afetando da infância até a vida adulta, com grau de manifestação diferenciadas. A primeira mutação ATP6 relatada foi a troca de um nucleotídeo timina por uma guanina, em que se faz a substituição do aminoácido leucina por arginina, acarretando na doença de NARP.

As manifestações clínicas dessa mutação estão associadas ao atraso do desenvolvimento neuropsicomotor, hipotonia, crises convulsivas, ataxia, sinais piramidais, cardiopatia hipertrófica, níveis elevados de lactato de alanina no sangue e/ou urina e retinite pigmentosa.

Porém, apesar de alguns distúrbios neurodegenerativos e cardiovasculares estarem relacionados com mutações no ATP6, os mecanismos pelos quais as alterações nessa proteína interferem na síntese do ATP ainda precisam ser elucidados. As principais suposições são de que mutações no ATP6 poderiam impedir a translocação de prótons na região F_o, impedindo a rotação da subunidade ATP6. Outra alternativa é a de que as mutações podem causar mudanças estruturais no ATP6, resultando em acoplamento ineficiente entre o transporte de prótons e o ATP síntese

4.3 GERAÇÃO DA ENZIMA MITOCONDRIAL ATP6

A sequência de DNA associada ao ATP6 foi escolhida para ser reproduzida pelo algoritmo por possuir uma região codificante com comprimento de 63 nucleotídeos, sendo este o menor comprimento de sequência encontrado que satisfaz $n = 2^r - 1$. Escolhemos essa sequência também, devido ao fato de que mutações nessa região podem acarretar em doenças genéticas.

Nesta seção foi executado o algoritmo de geração de proteínas, a fim de identificar se a enzima mitocondrial ATP6 poderia ser identificada e reproduzida por meio dos códigos corretores de erros. Uma vez observada e determinada a estrutura algébrica nessa sequência, pode-se realizar análises mutacionais identificando possíveis mutações que podem ocorrer na sequência ATP6 e estudar como essas alterações podem modificar propriedades físico-químicas de um aminoácido.

Os passos do algoritmo são determinados de maneira análoga ao procedimento adotado no exemplo da construção de códigos BCH sobre anel $\mathbb{Z}_4$, apresentado na Seção 3.2, do Capítulo 3. Considere a construção do código BCH primitivo sobre a estrutura de anel com parâmetros $(n, k, d_H) = (63,\ k, d_H\ )$ capaz de gerar e reproduzir sequências de DNA com comprimentos n = 63 nucleotídeos.

Passo 1 - Especificar a estrutura matemática e o alfabeto do código

O alfabeto 4-ário do código genético está relacionado ao conjunto formado pelos nucleotídeos denotados por $N = \{A, C, G, T/U\}$ que corresponde, respectivamente, à adenina, citosina, guanina e timina/uracila. Adotaremos o alfabeto 4-ário denotado por $\mathbb{Z}_4 = \{0, 1, 2, 3\}$. Todas as operações algébricas obedecerão às operações de adição e multiplicação módulo 4, Tabelas 13 e 14, respectivamente.

Tabela 13 – Adição módulo 4

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Fonte: Da autora.

Tabela 14 – Multiplicação módulo 4

·	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Fonte: Da autora.

Passo 2 - Determinar a extensão de Galois

Aqui, utilizaremos uma sequência de DNA formada por 63 nucleotídeos. Logo, o grau dos polinômios primitivos a ser usado na extensão de Galois do corpo GF (2^r) é $r = 6$, pois $n = 2^r - 1 = 2^6 - 1 = 63$.

Passo 3 - Determinar todos os polinômios primitivos $p(x)$, relacionados à extensão de Galois

Para o grau da extensão de Galois $r = 6$, são obtidos seis polinômios primitivos, listados no Quadro 4 a seguir:

Quadro 4 – Polinômios primitivos da extensão de Galois de grau $r = 6$

Polinômios primitivos $p(x)$	
$p_1(x) = x^6 + x + 1$	$p_4(x) = x^6 + x^5 + x^2 + x + 1$
$p_2(x) = x^6 + x^4 + x^3 + x + 1$	$p_5(x) = x^6 + x^5 + x^3 + x^2 + 1$
$p_3(x) = x^6 + x^5 + 1$	$p_6(x) = x^6 + x^5 + x^4 + x + 1$

Fonte: Da autora.

Passo 4 - Determinar a extensão do corpo $GF(2)$

Seja α um elemento primitivo em $\mathbb{F}_{64}$, α é uma raíz de $x^6 + x^4 + x^3 + x^1 + 1 = 0$, então temos $\alpha^6 + \alpha^4 + \alpha^3 + \alpha^1 + 1 = 0$ ou $\alpha^6 = -\alpha^4 - \alpha^3 - \alpha^1 - 1$. Porém, como os

coeficientes dos polinômios que formam o conjunto dos elementos de $\mathbb{F}_{64}$ pertencem a $\mathbb{F}_2$ devemos fazer a redução módulo 2. Portanto, $\alpha^6 = \alpha^4 + \alpha^3 + \alpha^1 + 1$. Os elementos de $\mathbb{F}_{64}$ são mostrados na Tabela 15.

Tabela 15 – Elementos de $\mathbb{F}_{64}$

Potência	**Vetorial**	**Potência**	**Vetorial**	**Potência**	**Vetorial**
0	(000000)	1	(100000)	α^1	(010000)
α^2	(001000)	α^3	(000100)	α^4	(000010)
α^5	(000001)	α^6	(110110)	α^7	(011011)
α^8	(111011)	α^9	(101011)	α^{10}	(100011)
α^{11}	(100111)	α^{12}	(100101)	α^{13}	(100100)
α^{14}	(010010)	α^{15}	(001001)	α^{16}	(110010)
α^{17}	(011001)	α^{18}	(111010)	α^{19}	(011101)
α^{20}	(111000)	α^{21}	(011100)	α^{22}	(001110)
α^{23}	(000111)	α^{24}	(110101)	α^{25}	(101100)
α^{26}	(010110)	α^{27}	(001011)	α^{28}	(110011)
α^{29}	(101111)	α^{30}	(100001)	α^{31}	(100110)
α^{32}	(010011)	α^{33}	(111111)	α^{34}	(101001)
α^{35}	(100010)	α^{36}	(010001)	α^{37}	(111110)
α^{38}	(011111)	α^{39}	(111001)	α^{40}	(101010)
α^{41}	(010101)	α^{42}	(111100)	α^{43}	(011110)
α^{44}	(001111)	α^{45}	(110001)	α^{46}	(101110)
α^{47}	(010111)	α^{48}	(111101)	α^{49}	(101000)
α^{50}	(010100)	α^{51}	(001010)	α^{52}	(000101)
α^{53}	(110100)	α^{54}	(011010)	α^{55}	(001101)
α^{56}	(110000)	α^{57}	(011000)	α^{58}	(001100)
α^{59}	(000110)	α^{60}	(000011)	α^{61}	(110111)
α^{62}	(101101)	α^{63}	(100000)		

Fonte: Da autora.

Passo 5 - Determinar a extensão do anel $\mathbb{Z}_4$

Consideremos agora o anel $GR\left(p^k, r\right) = GR^*\left(4, 6\right)$, dado por:

$$\frac{\mathbb{Z}_4\left[x\right]}{\langle p\left(x\right)\rangle} \cong \frac{\mathbb{Z}_4\left[x\right]}{\langle 1x^6 + 1x^4 + 1x^3 + 1x^1 + 1\rangle} = \left\{b_0 + b_1x + b_2x^2 + \cdots + b_5x^5 : \ b_i's \in \mathbb{Z}_4\right\},$$

Seja β uma raiz de $p\left(x\right)$. Então, $\beta^6 + \beta^4 + \beta^3 + \beta^1 + 1 = 0$ levando a $\beta^6 = -\beta^4 - \beta^3 - \beta^1 - 1$. Porém, note que agora estamos trabalhando em $\mathbb{Z}_4$. Sendo assim, $\beta^6 = 3\beta^4 + 3\beta^3 + 3\beta^1 + 3$. A Tabela 16 apresenta todos os elementos não nulos e inversíveis do grupo cíclico do grupo $\mathrm{GR}^*\left(4, 6\right)$:

Tabela 16 – Elementos do grupo cíclico $\mathrm{GR}^*(4,6)$

$GR^*(4,6)$	**Vetorial**	$GR^*(4,6)$	**Vetorial**	$GR^*(4,6)$	**Vetorial**
1	(100000)	β^1	(010000)	β^2	(001000)
β^3	(000100)	β^4	(000010)	β^5	(000001)
β^6	(330330)	β^7	(033033)	β^8	(113013)
β^9	(121011)	β^{10}	(302031)	β^{11}	(320133)
β^{12}	(102123)	β^{13}	(120322)	β^{14}	(232212)
β^{15}	(203001)	β^{16}	(310230)	β^{17}	(031023)
β^{18}	(113212)	β^{19}	(231101)	β^{20}	(313000)
β^{21}	(031300)	β^{22}	(003130)	β^{23}	(000313)
β^{24}	(110101)	β^{25}	(301300)	β^{26}	(030130)
β^{27}	(003013)	β^{28}	(110011)	β^{29}	(301331)
β^{30}	(320023)	β^{31}	(102112)	β^{32}	(230031)
β^{33}	(313333)	β^{34}	(101003)	β^{35}	(120210)
β^{36}	(012021)	β^{37}	(331132)	β^{38}	(213333)
β^{39}	(131003)	β^{40}	(123210)	β^{41}	(012321)
β^{42}	(331122)	β^{43}	(213332)	β^{44}	(201113)
β^{45}	(130221)	β^{46}	(303312)	β^{47}	(210111)
β^{48}	(311301)	β^{49}	(321020)	β^{50}	(032102)
β^{51}	(223030)	β^{52}	(022303)	β^{53}	(112300)
β^{54}	(011230)	β^{55}	(001123)	β^{56}	(110222)
β^{57}	(231202)	β^{58}	(203300)	β^{59}	(020330)
β^{60}	(002033)	β^{61}	(110313)	β^{62}	(121101)
β^{63}	(302000)	β^{64}	(030200)	β^{65}	(003020)
β^{66}	(000302)	β^{67}	(220210)	β^{68}	(022021)
β^{69}	(332132)	β^{70}	(213033)	β^{71}	(131013)
β^{72}	(123211)	β^{73}	(302211)	β^{74}	(320111)
β^{75}	(322301)	β^{76}	(322120)	β^{77}	(032212)
β^{78}	(223001)	β^{79}	(312230)	β^{80}	(031223)
β^{81}	(113232)	β^{82}	(231103)	β^{83}	(133220)
β^{84}	(013322)	β^{85}	(221112)	β^{86}	(202331)
β^{87}	(310123)	β^{88}	(101122)	β^{89}	(230332)
β^{90}	(203213)	β^{91}	(130031)	β^{92}	(303333)
β^{93}	(100003)	β^{94}	(120110)	β^{95}	(012011)
β^{96}	(331131)	β^{97}	(323003)	β^{98}	(102010)
β^{99}	(010201)	β^{100}	(331310)	β^{101}	(033131)
β^{102}	(333203)	β^{103}	(103030)	β^{104}	(010303)

Continua

Tabela 16 – Elementos do grupo cíclico $\mathbf{GR}^*(4,6)$

$GR^*(4,6)$	**Vetorial**	$GR^*(4,6)$	**Vetorial**	$GR^*(4,6)$	**Vetorial**
β^{105}	(111100)	β^{106}	(011110)	β^{107}	(001111)
β^{108}	(330001)	β^{109}	(323330)	β^{110}	(032333)
β^{111}	(113303)	β^{112}	(121000)	β^{113}	(012100)
β^{114}	(001210)	β^{115}	(000121)	β^{116}	(330302)
β^{117}	(213210)	β^{118}	(021321)	β^{119}	(332022)
β^{120}	(213022)	β^{121}	(201122)	β^{122}	(200332)
β^{123}	(200213)	β^{124}	(130131)	β^{125}	(303303)
β^{126}	(100000)				

Fonte: Da autora.

Passo 6 - Determinar o grupo das unidades

Nesta etapa do algoritmo iremos construir o subgrupo cíclico, sendo formado por elementos da extensão do anel e será baseado no parâmetro d. Como f gera um subgrupo de ordem $63.d = 126$, temos que $d = 2$.

Como $d = 2$, teremos um subgrupo formado por 63 elementos com $f^2 \rightarrow (001000)$, sendo este considerado como o elemento primitivo que gera o subgrupo cíclico G_{63}. A Tabela 17 apresenta os elementos constituintes do subgrupo:

Tabela 17 – Elementos de G_{63}

$GR^*(4,6)$	**Vetorial**	$GR^*(4,6)$	**Vetorial**	$GR^*(4,6)$	**Vetorial**
$(f^2)^1 = \beta^2$	(001000)	$(f^2)^2 = \beta^4$	(000010)	$(f^2)^3 = \beta^6$	(330330)
$(f^2)^4 = \beta^8$	(113013)	$(f^2)^5 = \beta^{10}$	(302031)	$(f^2)^6 = \beta^{12}$	(102123)
$(f^2)^7 = \beta^{14}$	(232212)	$(f^2)^8 = \beta^{16}$	(310230)	$(f^2)^9 = \beta^{18}$	(113212)
$(f^2)^{10} = \beta^{20}$	(313000)	$(f^2)^{11} = \beta^{22}$	(003130)	$(f^2)^{12} = \beta^{24}$	(110101)
$(f^2)^{13} = \beta^{26}$	(030130)	$(f^2)^{14} = \beta^{28}$	(110011)	$(f^2)^{15} = \beta^{30}$	(320023)
$(f^2)^{16} = \beta^{32}$	(230031)	$(f^2)^{17} = \beta^{34}$	(101003)	$(f^2)^{18} = \beta^{36}$	(012021)
$(f^2)^{19} = \beta^{38}$	(213333)	$(f^2)^{20} = \beta^{40}$	(123210)	$(f^2)^{21} = \beta^{42}$	(331122)
$(f^2)^{22} = \beta^{44}$	(201113)	$(f^2)^{23} = \beta^{46}$	(303312)	$(f^2)^{24} = \beta^{48}$	(311301)
$(f^2)^{25} = \beta^{50}$	(032102)	$(f^2)^{26} = \beta^{52}$	(022303)	$(f^2)^{27} = \beta^{54}$	(011230)
$(f^2)^{28} = \beta^{56}$	(110222)	$(f^2)^{29} = \beta^{58}$	(203300)	$(f^2)^{30} = \beta^{60}$	(002033)
$(f^2)^{31} = \beta^{62}$	(121101)	$(f^2)^{32} = \beta^{64}$	(030200)	$(f^2)^{33} = \beta^{66}$	(000302)
$(f^2)^{34} = \beta^{68}$	(022021)	$(f^2)^{35} = \beta^{70}$	(213033)	$(f^2)^{36} = \beta^{72}$	(123211)
$(f^2)^{37} = \beta^{74}$	(320111)	$(f^2)^{38} = \beta^{76}$	(322120)	$(f^2)^{39} = \beta^{78}$	(223001)
$(f^2)^{40} = \beta^{80}$	(031223)	$(f^2)^{41} = \beta^{82}$	(231103)	$(f^2)^{42} = \beta^{84}$	(013322)
$(f^2)^{43} = \beta^{86}$	(202331)	$(f^2)^{44} = \beta^{88}$	(101122)	$(f^2)^{45} = \beta^{90}$	(203213)
$(f^2)^{46} = \beta^{92}$	(303333)	$(f^2)^{47} = \beta^{94}$	(120110)	$(f^2)^{48} = \beta^{96}$	(331131)

Continua

Tabela 17 – Elementos de G_{63}

$GR^*(4,6)$	**Vetorial**	$GR^*(4,6)$	**Vetorial**	$GR^*(4,6)$	**Vetorial**
$(f^2)^{49} = \beta^{98}$	(102010)	$(f^2)^{50} = \beta^{100}$	(331310)	$(f^2)^{51} = \beta^{102}$	(333203)
$(f^2)^{52} = \beta^{104}$	(010303)	$(f^2)^{53} = \beta^{106}$	(011110)	$(f^2)^{54} = \beta^{108}$	(330001)
$(f^2)^{55} = \beta^{110}$	(032333)	$(f^2)^{56} = \beta^{112}$	(121000)	$(f^2)^{57} = \beta^{114}$	(001210)
$(f^2)^{58} = \beta^{116}$	(330302)	$(f^2)^{59} = \beta^{118}$	(021321)	$(f^2)^{60} = \beta^{120}$	(213022)
$(f^2)^{61} = \beta^{122}$	(200332)	$(f^2)^{62} = \beta^{124}$	(130131)	$(f^2)^{63} = \beta^{126}$	(100000)

Fonte: Da autora.

Passo 7 - Determinar o polinômio gerador da matriz G, $g(x)$

Neste passo, iremos obter o polinômio gerador $g(x)$. O polinômio gerador do código BCH de comprimento n tem como raízes os elementos $\left\{(\beta^i), (\beta^i)^p, \ldots, (\beta^i)^{p^{r-1}(modn)}\right\}$, e é representado por:

$$g(x) = mmc(M_1(x), M_2(x), \ldots, M_{2t}(x)),$$

em que $M_i(x)$ é o polinômio minimal associado ao elemento primitivo β^i, $i = 1, 2, \ldots, 2t$, e mmc denota o mínimo múltiplo comum.

O polinômio $g(x)$ é calculado por meio de três etapas:

1. Cálculo das raízes dos polinômios minimais;
2. Cálculo dos polinômios minimais $M_i(x)$, para todo $i = 1, 2, \cdots, 62$;
3. Cálculo dos polinômios geradores para $1 \leq t \leq 31$.

O polinômio gerador $g(x)$ para cada valor de t é dado pelo mínimo múltiplo comum formado pelos polinômios minimais diferentes entre si. Considerando que a distância mínima do código seja $d_H = 3$, então o polinômio gerador do código é dado por $g(x) = x^6 + 2x^5 + x^4 + x^3 + 3x^1 + 1$.

Passo 8 - Determinar o polinômio gerador da matriz H, $h(x)$

Para a obtenção do polinômio $h(x)$ realizamos o cálculo da seguinte relação:

$$h(x) = \frac{x^n - 1}{g(x)} = \frac{x^{63} - 1}{x^6 + 2x^5 + x^4 + x^3 + 3x^1 + 1}$$

$h(x) = 1x^{57} + 2x^{56} + 3x^{55} + 3x^{54} + 1x^{53} + 1x^{52} + 3x^{51} + 1x^{50} + 2x^{49} + 2x^{48} + 1x^{47} + 3x^{45} + 2x^{44} + 1x^{43} + 2x^{40} + 3x^{39} + 3x^{38} + 2x^{36} + 3x^{35} + 1x^{34} + 1x^{33} + 3x^{32} + 2x^{31} + 1x^{30} + 3x^{29} + 3x^{28} + 1x^{26} + 2x^{25} + 1x^{24} + 3x^{23} + 1x^{21} + 2x^{19} + 1x^{18} + 1x^{17} + 1x^{15} + 3x^{14} + 1x^{10} + 3x^7 + 2x^6 + 2x^5 + 2x^4 + 3x^2 + 3x^1 + 3.$

Passo 9 - Determinar a matriz G

Considerando o polinômio gerador $g_1(x) = x^6 + 2x^5 + x^4 + x^3 + 3x^1 + 1$, realizando os deslocamentos dos coeficientes do polinômio $g(x)$ da esquerda para a direita, obtemos a matriz geradora G, com dimensão 57×63 :

$$G = \begin{bmatrix} 130112100000000000000000000000000000000000000000000000000000000 \\ 013011210000000000000000000000000000000000000000000000000000000 \\ 001301121000000000000000000000000000000000000000000000000000000 \\ 000130112100000000000000000000000000000000000000000000000000000 \\ \vdots \\ 000000000000000000000000000000000000000000000000000001301121000 \\ 000000000000000000000000000000000000000000000000000000130112100 \\ 000000000000000000000000000000000000000000000000000000013011210 \\ 000000000000000000000000000000000000000000000000000001301121 \end{bmatrix}.$$

Passo 10 - Determinar a matriz H e a sua transposta H^T

Neste passo, teremos a construção da chamada matriz verificação de paridade H por meio do polinômio $h\,(x)$. A primeira linha da matriz será constituída pelos coeficientes do referido polinômio. Para a composição das linhas subsequentes deve-se fazer um deslocamento da direita para esquerda dos elementos formadores da primeira linha.

$$H = \begin{bmatrix} 123311312210321002330231132133012130102110130001003222033300000 \\ 012331131221032100233023113213301213010211013000100322203330000 \\ 001233113122103210023302311321330121301021101300010032220333000 \\ 000123311312210321002330231132133012130102110130001003222033300 \\ 000012331131221032100233023113213301213010211013000100322203330 \\ 000001233113122103210023302311321330121301021101300010032220333 \end{bmatrix}.$$

Passo 11 - Rotular a sequência de DNA utilizando o Passo 1

Vamos verificar se o código BCH primitivo sobre anel é capaz de reproduzir a sequência de DNA Homo sapiens [P30049 - 12587], ATP synthase subunit delta, mitochondrial precursor, inner membrane 6 com 63 nucleotídeos. Essa sequência deve ser rotulada com as 24 permutações apresentadas no Quadro 5, resultando na matriz P.

Quadro 5 – Relação entre as linhas da matriz P e as 24 permutações

Linha = Caso	$N \to \mathbb{Z}_4$	**Linha = Caso**	$N \to \mathbb{Z}_4$
L 1 = Caso 01	(A, C, G, T) = (0, 1, 2, 3)	L 13 = Caso 13	(A, C, G, T) = (2, 0, 1, 3)
L 2 = Caso 02	(A, C, G, T) = (0, 1, 3, 2)	L 14 = Caso 14	(A, C, G, T) = (2, 0, 3, 1)
L 3 = Caso 03	(A, C, G, T) = (0, 2, 1, 3)	L 15 = Caso 15	(A, C, G, T) = (2, 1, 0, 3)
L 4 = Caso 04	(A, C, G, T) = (0, 2, 3, 1)	L 16 = Caso 16	(A, C, G, T) = (2, 1, 3, 0)
L 5 = Caso 05	(A, C, G, T) = (0, 3, 2, 1)	L 17 = Caso 17	(A, C, G, T) = (2, 3, 0, 1)
L 6 = Caso 06	(A, C, G, T) = (0, 3, 1, 2)	L 18 = Caso 18	(A, C, G, T) = (2, 3, 1, 0)
L 7 = Caso 07	(A, C, G, T) = (1, 0, 2, 3)	L 19 = Caso 19	(A, C, G, T) = (3, 0, 1, 2)
L 8 = Caso 08	(A, C, G, T) = (1, 0, 3, 2)	L 20 = Caso 20	(A, C, G, T) = (3, 0, 2, 1)
L 9 = Caso 09	(A, C, G, T) = (1, 2, 0, 3)	L 21 = Caso 21	(A, C, G, T) = (3, 1, 0, 2)
L 10 = Caso 10	(A, C, G, T) = (1, 2, 3, 0)	L 22 = Caso 22	(A, C, G, T) = (3, 1, 2, 0)
L 11 = Caso 11	(A, C, G, T) = (1, 3, 0, 2)	L 23 = Caso 23	(A, C, G, T) = (3, 2, 0, 1)
L 12 = Caso 12	(A, C, G, T) = (1, 3, 2, 0)	L 24 = Caso 24	(A, C, G, T) = (3, 2, 1, 0)

Fonte: Da autora.

Seja a sequência do $NCBI$ igual a:

CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCACGCCCGTGCCTAT

então obtemos a matriz P:

$$P=\begin{bmatrix}
1321112112121321311211211122201332211211312311211012111232113 03\\
1231113113131231211311311133301223311311213211311013111321122 02\\
2312221221212312322122122211102331122122321322122021222131223 03\\
2132223223232132122322322233302113322322123122322023222312210 1\\
3123332332323123133233233322031122332331321332330323332123310 1\\
3213331331313213233133133311103221133133231233133033133121332 02\\
0320002002020320300200200022210332200200302300200102000232003 13\\
0230003003030230200300300033310223300300203200300103000323002 12\\
2302220220202302322022022000123300220223203220221202220302231 3\\
2032223223232032022322322233312003322322023022322123222303220 10\\
3203330330303203233033033300013220033033230233033130333020332 12\\
3023332332323023033233233322213002233233032033233132333202330 10\\
0310001001010310300100100011120331100100301300100201000131003 23\\
0130003003030130100300300033320113300300103100300203000313001 21\\
1301110110101301311011011100021330011011310311011210111030113 23\\
1031113113131031011311311133321003311311013011311213111303110 20\\
3103330330303103133033033300023110033033130133033233033010331 21\\
3013331331313013033133133311123001133133031033133231333101330 20\\
0210001001010210200100100011130221100100201200100301000121002 32\\
0120002002020120100200200022230112200200102100200302000212001 31\\
1201110110101201211011011100031220011011210211011310111020112 32\\
1021112112121021011211211122231002211211012011211312111202110 30\\
2102220220202102122022022000321100220221201220223202220102213 1\\
2012221221212012022122122211132001122122021022122321222101220 30
\end{bmatrix}$$

Passo 12 -Verificar se a sequência de DNA é palavra-código de acordo com os padrões de erros estabelecidos: D(a, b) = 0, D(a, b) = 1 e D(a, b) = 2

Vamos analisar se as sequências (sem diferença, com 1 ou com 2 nucleotídeos de diferença da sequência postada no NCBI) são palavras código dos códigos (n, k, d_H) usando a equação $v.H^T = 0$, da seguinte maneira:

a) para analisarmos as sequências de DNA com até 1 nucleotídeo de diferença da sequência de DNA original, consideramos as 4356 possíveis palavras código para cada sequência de DNA analisada. As palavras código encontradas são armazenadas.

b) já para analisarmos as sequências de DNA com até 2 nucleotídeos de diferença da sequência de DNA original, consideramos todas as combinações 2 a 2 dos n nucleotídeos de comprimento da sequência para as 24 permutações. As palavras código encontradas são armazenadas.

Como os demais passos do algoritmo são análogos ao apresentado no exemplo do Capítulo 3, passemos aos resultados desse algoritmo.

Como resultado das simulações da enzima ATP6 para $D(a, b) = 1$, diferindo em um nucleotídeo da sequência do $NCBI$, foram obtidas 8 palavras código, listadas a seguir:

Palavra código: 3 de rotulamento C :: (A = 0, G = 1, C = 2, T = 3)

```
Oaa:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..H..A..R..A..Y.
Ont:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCACGCCCGTGCCTAT
OLb:  2312221221212312322122122211102331122122321322122021222131223 03
GLb:  2312221221212312322122122211102331122122321322122011222131223 03
```

```
Gnt:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCAGGCCCGTGCCTAT
Goo:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..Q..A..R..A..Y.
```

Palavra código: 3 de rotulamento C :: (A = 0, T = 1, C = 2, G = 3)

```
Oaa:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..H..A..R..A..Y.
Ont:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCACGCCCGTGCCTAT
OLb:  213222322323213212232322223330211332232212312232202322231322101
GLb:  213222322323213212232322223330211332232212312232203322231322101
Gnt:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCAGGCCCGTGCCTAT
Goo:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..Q..A..R..A..Y.
```

Palavra código: 3 de rotulamento C :: (G = 0, A = 1, T = 2, C = 3)

```
Oaa:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..H..A..R..A..Y.
Ont:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCACGCCCGTGCCTAT
OLb:  320333033030320323303303330001322003303323023303313033302033212
GLb:  320333033030320323303303330001322003303323023303310033302033212
Gnt:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCAGGCCCGTGCCTAT
Goo:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..Q..A..R..A..Y.
```

Palavra código: 3 de rotulamento C :: (T = 0, A = 1, G = 2, C = 3)

```
Oaa:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..H..A..R..A..Y.
Ont:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCACGCCCGTGCCTAT
OLb:  302333233232302303323323332221300223323303203323313233320233010
GLb:  302333233232302303323323332221300223323303203323312233320233010
Gnt:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCAGGCCCGTGCCTAT
Goo:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..Q..A..R..A..Y.
```

Palavra código: 3 de rotulamento C :: (C = 0, G = 1, A = 2, T = 3)

```
Oaa:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..H..A..R..A..Y.
Ont:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCACGCCCGTGCCTAT
OLb:  031000100101031030010010001112033110010030130010020100013100323
GLb:  031000100101031030010010001112033110010030130010021100013100323
Gnt:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCAGGCCCGTGCCTAT
Goo:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..Q..A..R..A..Y.
```

Palavra código: 3 de rotulamento C :: (C = 0, T = 1, A = 2, G = 3)

```
Oaa:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..H..A..R..A..Y.
Ont:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCACGCCCGTGCCTAT
OLb:  013000300303013010030030003332011330030010310030020300031300121
GLb:  013000300303013010030030003332011330030010310030023300031300121
Gnt:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCAGGCCCGTGCCTAT
Goo:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..Q..A..R..A..Y.
```

Palavra código: 3 de rotulamento C :: (G = 0, C = 1, T = 2, A = 3)

```
Oaa:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..H..A..R..A..Y.
Ont:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCACGCCCGTGCCTAT
OLb:  120111011010120121101101110003122001101121021101131011102011232
GLb:  120111011010120121101101110003122001101121021101130011102011232
Gnt:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCAGGCCCGTGCCTAT
Goo:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..Q..A..R..A..Y.
```

Palavra código: 3 de rotulamento C :: (T = 0, C = 1, G = 2, A = 3)

```
Oaa:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..H..A..R..A..Y.
Ont:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCACGCCCGTGCCTAT
OLb:  102111211212102101121121112223100221121101201121131211120211030
GLb:  102111211212102101121121112223100221121101201121132211120211030
Gnt:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCAGGCCCGTGCCTAT
Goo:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..Q..A..R..A..Y.
```

Essas palavras código são diferentes em termo do alfabeto do código $\mathbb{Z}_4 = \{0, 1, 2, 3\}$, no entanto quando as rotulamos para o alfabeto do código genético $N = \{A, C, G, T/U\}$ elas são iguais, resultando em única sequência de DNA. Como as sequências de DNA reproduzidas são iguais, é suficiente analisar apenas um dos casos (Figura 14). Ressaltamos que a diferença do dígito nessas palavras código sempre ocorre na mesma posição.

A Figura 14 , mostra a sequência de direcionamento da enzima ATP6 reproduzida pelo código Klein-linearidade ((63, 57, 3), BCH primitivo sobre $GR(4, 6)$, rotulamento C), através do polinômio primitivo $p(x) = x^6 + x^5 + 1$ e do polinômio gerador $g(x) = x^6 + 3x^5 + 2x^3 + 1$.

Observe que, na posição da trinca 17 houve uma troca do nucleotídeo citosina (CAC) por um nucleotídeo guanina (CAG), ocasionando a troca de aminoácido nesta posição, sendo a histidina (H) substituída pela glutamina (Q).

E, como resultado do algoritmo para $D(a, b) = 2$, diferindo em dois nucleotídeos da sequência do $NCBI$, não encontramos palavras código com 2 dígitos de diferença da sequência original.

Figura 14 – Sequência gerada com $n = 63$ nucleotídeos

Palavra código: 3 de rotulamento C :: (T = 0, C = 1, G = 2, A = 3)

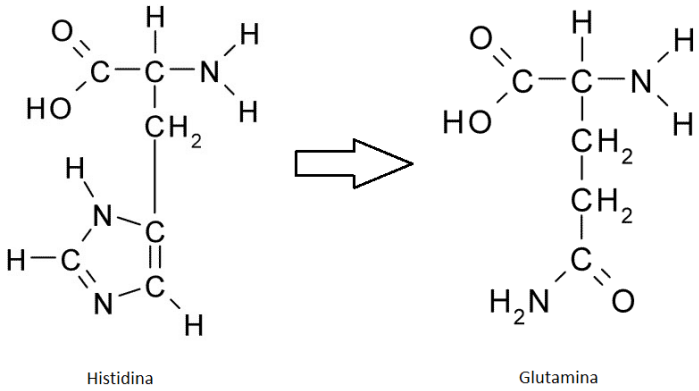
```
Oaa:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..H..A..R..A..Y.
Ont:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCACGCCCGTGCCTAT
OLb:  10211121121210210111211211122231002211211012011211312111202110 30
GLb:  10211121121210210111211211122231002211211012011211322111202110 30
Gnt:  CTGCCCGCCGCGCTGCTCCGCCGCCCGGGACTTGGCCGCCTCGTCCGCCAGGCCCGTGCCTAT
Goo:  .L..P..A..A..L..L..R..R..P..G..L..G..R..L..V..R..Q..A..R..A..Y.
```

Fonte: Da autora.

4.4 ANÁLISE DAS SIMULAÇÕES DA PROTEÍNA MITOCONDRIAL ATP6

Do ponto de vista biológico, a troca do aminoácido histidina pelo aminoácido glutamina, identificado pelo algoritmo de geração de proteínas, não implica na mudança de polaridade do aminoácido, pois essa modificação ocorre dentro da mesma classe hidrofílica. Com isso, a princípio, os efeitos fenotípicos são menos drásticos, pois a diferença na natureza química das cadeias laterais desses aminoácidos é menor, já que no aminoácido histidina as aminas estão presentes em sua cadeia lateral, enquanto que a glutamina tem cadeias laterais eletricamente neutras em pH neutro, como pode ser visto na Figura 15.

Figura 15 – Estrutura química da histidina e da glutamina



Fonte: Da autora.

A alteração reproduzida pelo algoritmo é classificada como uma mutação pontual missense, pois apenas um único par de bases é mudado, levando a troca de um aminoácido por outro. Disfunção da ATP sintase devido a uma mutação missense na ATP6 é uma causa frequente de doenças mitocondriais graves.

Mutações na proteína ATP6 estão principalmente associadas a doença hereditária da neuropatia óptica de Leber (LHON), indicada pela substituição do nucleotídeo timina pelo nucleotídeo citosina ou o nucleotídeo adenina por um nucleotídeo guanina; necrose estriada bilateral, geralmente identificados pela substituição do nucleotídeo timina por nucleotídeo guanina e a doença de NARP, que na maioria dos casos está relacionada à

substituição do nucleotídeo timina pelo nucleotídeo citosina ou guanina.

No entanto, uma nova mutação na região sobreposta do ATP6 e ATP8 causadas por alteração do nucleotídeo da citosina (C) por um nucleotídeo de guanina (G) foi identificada, conforme apontam Blanco-Grau *et al.* (2013), Duno *et al.* (2013) e Komulainen *et al.* (2016). Essa mutação causa defeito na estrutura e produção do ATP, resultando clinicamente em um fenótipo com características típicas de ataxia cerebelar, neuropatia periférica e retinite pigmentosa (doença NARP), diabetes mellitus e hipogonadismo hipergonadotrópico.

Essas condições genéticas afetam principalmente o sistema nervoso e o sistema reprodutivo, ou seja, tecidos que demandam uma alta carga de energia proveniente de ATP. A doença de NARP possui manifestações clínicas que incluem neuropatia sensorial, ataxia, convulsões, demência e retinite pigmentosa (quando há perda da visão ocular). Diabetes mellitus é uma doença na qual altos níveis de glicose no sangue são encontrados por um longo período de tempo. Enquanto que o hipogonadismo hipergonadotrópico é uma doença na qual os testículos em homens e ovários nas mulheres não produzem quantidades adequadas de hormônios sexuais.

Com isso, comparando o resultado matemático obtido com dados biológicos, é possível avaliar que existe aplicabilidade do uso da teoria da codificação no contexto biológico, uma vez que a alteração reproduzida pelo algoritmo no ATP6, que implica na troca do nucleotídeo citosina por um nucleotídeo guanina foi comprovado por análises laboratoriais, mostrando que a alteração do nucleotídeo interferem na função da sequência analisada. Assim, mutações na região no gene ATP6 são passíveis de serem estudadas por meio desse algoritmo. Portanto, os códigos corretores de erros são eficientes para localizar mutações em uma molécula de DNA.

5 CONCLUSÕES E SUGESTÕES DE TRABALHOS FUTUROS

Um dos maiores desafios para os pesquisadores que utilizam os conceitos da teoria da informação, da codificação e da comunicação para análises de dados genéticos é mostrar a existência de uma estrutura de códigos corretores de erros inerentes à estrutura do DNA. Historicamente, a aplicação da teoria de informação nos sistemas biológicos se iniciou na década de 70 e se impulsionou na década de 80, com o aumento dos dados genéticos. Desde então, as pesquisas que utilizam essa aplicação buscam encontrar analogias entre o fluxo de informação biológica e o sistema de comunicação digital, a fim de obter uma melhor compreensão dos padrões biológicos existentes. Rocha (2010) e Faria (2010) mostraram pela primeira vez em seus estudos as semelhanças entre os sistemas de comunicação e os sistemas biológicos, apontando para a existência de uma estrutura matemática nas fitas simples do RNA e na dupla hélice do DNA associada aos códigos corretores de erros e, com base nessas semelhanças propuseram o modelo de codificação genética e o modelo de codificação genômica, capazes de identificar, reproduzir e classificar matematicamente diferentes sequências de DNA. Para a reprodução de tais sequências, Rocha (2010) e Faria (2010) desenvolveram algoritmos computacionais, denominados de algoritmos de geração de proteínas, responsáveis pela identificação de sequências de DNA, em que descrevem a construção dos códigos BCH sobre corpos e códigos BCH sobre anéis. Esses algoritmos permitem a identificação dessas sequências com até dois nucleotídeos diferindo das sequências postadas no $NCBI$, sendo então uma ferramenta que pode ser aplicada no estudo de análises mutacionais.

Fundamentado nas pesquisas citadas anteriormente, o objetivo deste trabalho foi efetuar um estudo da estrutura algébrica dos códigos BCH e gerar uma sequência de DNA, por meio do algoritmo de geração de proteínas, que possuísse uma função biológica conhecida. Dentre as sequências selecionadas, a sequência da proteína mitocondrial ATP6 foi escolhida para ser reproduzida pelo algoritmo por atender as restrições impostas pelo mesmo, além de que erros na enzima ATP6 acarretam em doenças genéticas que afetam principalmente tecidos do sistema nervoso central, cardiovascular e muscular. Mediante a simulação foi possível verificar a ocorrência de uma troca de nucleotídeo da posição da trinca 17 da enzima ATP6, em que o nucleotídeo citosina é modificado pelo nucleotídeo guanina, levando a substituição do aminoácido histidina pelo aminoácido glutamina.

A substituição dos nucleotídeos gerada pela simulação está relacionada com uma mutação atípica encontrada na enzima mitocondrial ATP6 que acarreta a doença de NARP, a diabetes mellitus e o hipogonadismo hipergonadotrópico. Esta mutação foi verificada e comprovada por meio de análises laboratoriais apenas em 2013, até então era conhecido que apenas as mutações que levam a substituição do nucleotídeo timina pelo nucleotídeo guanina ou o nucleotídeo timina pelo nucleotídeo citosina que poderiam estar relacionados

com patologias clínicas ocasionadas por mutações no ATP6. Sendo assim, o algoritmo de geração de proteínas se mostrou um mecanismo eficaz para prever uma mutação baseada na sequência de DNA original, nos levando a questionar se as alterações genéticas acontecem de forma aleatória ou são resultado de uma sequência algébrica, já que a mutação reproduzida por este trabalho pode ser encontrada em pesquisas biomédicas.

Consequentemente, o algoritmo de geração de proteínas pode se tornar uma ferramenta mais barata e eficaz, se compararmos com análises clínicas, para o estudo de análises mutacionais, pois através dele é possível investigar possíveis mutações que podem ocorrer em sequências de DNA. Com isso, os códigos corretores de erros são eficientes para localizar uma mutação em uma sequência de DNA, podendo ser utilizado em aplicações terapeuticas, pois consegue reproduzir alguns mecanismos existentes no sistema biológico.

Este trabalho leva a várias questões a serem respondidas acerca da reprodução de sequências de DNA, por meio do algoritmo de geração de proteínas. A fim de dar prosseguimento a este estudo, apresentamos algumas sugestões, tais como: gerar e reproduzir sequências de DNA com comprimento maior do que 63 nucleotídeos; realizar análises bioquímicas das sequências de DNA geradas pelo algoritmo de geração de proteínas; gerar sequências de DNA utilizando os códigos BCH sobre a extensão de corpos e comparar os resultados obtidos nas simulações com situações práticas de laboratório.

De modo a dar continuidade a esta pesquisa, o próximo passo consiste em reproduzir, mediante o algoritmo de geração de proteínas, uma sequência de comprimento 127 nucleotídeos. Visando atigir esse obetivo, foi realizado um levantamento prévio no banco de dados do NCBI a procura de sequências de DNA que possuíssem comprimento de 127 nucleotídeos. Nessa busca inicial foram excluídas todas as sequências de DNA que não estavam relacionadas com sequências humanas. Com isso, desse levantamento foram selecionadas 13 sequências que atendiam as restrições impostas, e posteriormente foi verificado se o algoritmo de geração de proteínas conseguiria reproduzir essas sequências. Das 13 sequências, o algortimo foi capaz de reproduz apenas 8. Por meio da geração dessas 8 sequências foi analisada a troca de nucleotídeo gerada pelo algoritmo e verificado se trabalhos biomédicos já haviam comprovado laboratorialmente essas trocas de nucleotídeos. Após essa verificação, a sequência de 127 nucleotídeos escolhida pra ser estudada foi a **Sequence = Homo sapiens TMPRSS3 gene, DNA variation found in Japanese hearing loss patients**. Por meio da geração dessa sequência será realizada uma análise físico-química da mesma, verificando as possíveis trocas de nucleotídeos que podem acarretam na perda auditiva na população chinesa. Essa análise mais detalhada será feita em um trabalho futuro.

REFERÊNCIAS

AFONSO, A. *et al.* Malaria parasites can develop stable resistance to artemisinin but lack mutations in candidate genes atp6 (encoding the sarcoplasmic and endoplasmic reticulum ca ATPase), tctp, mdr1, and cg10. *Antimicrobial Agents and Chemotherapy*, v. 50, p. 480–489, 2006.

ALBERTS, B. *et al. Biologia molecular da célula.* 5. ed. Porto Alegre: ArtMed, 2010.

BALLMOOS, C.; DIMROTH, P.; MEIER, T. Catalytic and mechanical cycles in F-ATP synthases: fourth in the cycles review series. *EMBO Reports*, v. 3, p. 276–282, 2006.

BARBOSA, P. R. *Construção de códigos $\mathbb{Z}_{2^k}$-pseudolineares através de aplicações isométricas e extensões de Galois sobre anéis locais.* 2000. 107 f. Dissertação (Mestrado em engenharia elétrica) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2000.

BLANCO-GRAU, A. *et al.* Identification and biochemical characterization of the novel mutation m.8839G>C in the mitochondrial ATP6 gene associated with NARP syndrome. *Genes, Brain and Behavior*, v. 12, p. 812–820, 2013.

COSTELO JR, D. J.; LIN, S. *Error control coding: fundamentals and applications.* Englewood Cliffs: Prentice Hall, 1983.

DOMINGUES H. H.; IEZZI, G. *Álgebra moderna.* 4. ed. São Paulo: Atual, 2003.

DUNO, M. *et al.* A novel mitochondrial mutation m.8989 G>C associated with neuropathy, ataxia, retinitis pigmentosa - The NARP syndrome. *Gene*, v. 515, p. 372–375, 2013.

FARIA, L. C. B. et. al. DNA Sequences Generated by BCH Codes over GF (4). *Electronics Letters*, v. 46, p. 202–203, 2010.

GONZALEZ, M. E. D. *Modelagem da síntese de proteínas e sua estrutura organizacional através de Códigos Corretores de Erros.* 2017. 174 f. Tese (Doutorado em Engenharia Elétrica) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2017.

GRIFFITHS, A. J. F. *et al. Introdução à genética.* Rio de Janeiro: Guanabara Koogan, 2006.

HIB, J.; ROBERTIS, E. M. F. *Bases da biologia molecular e celular.* 4. ed. Rio de Janeiro: Guanabara Koogan, 2006.

INTERLANDO, J. C. *Uma contribuição à construção e decodificação de códigos lineares sobre grupos abelianos via concatenação de códigos sobre anéis de inteiros residuais.* 1994. 145 f. Tese (Doutorado em Engenharia Elétrica) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 1994.

KOMULAINEN, T. *et al.* A novel mutation m.8561C>G in MT-ATP6/8 causing a mitochondrial syndrome with ataxia, peripheral neuropathy, diabetes mellitus, and hypergonadotropic hypogonadism. *J Neurol*, v. 263, p. 2188–2195, 2016.

LYRA, R. *et al.* Clinical and molecular aspects of hypogonadotropic hypogonadism congenital isolated. *Brazilian Archive Metabolic Endocrinology*, v. 50, p. 239, 2006.

NASSEH, I. E. *et al.* Mitochondrial diseases. *Neuroscience Magazine*, v. 92, p. 60–69, 2001.

OLIVEIRA, A. J. *Análise algébrica dos rotulamentos associados ao mapeamento do código genético.* 2012. 105 f. Dissertação (Mestrado em Engenharia Elétrica) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2012.

PEREIRA, D. G. *Uma abordagem computacional para a análise de sequências de DNA por meio dos códigos corretores de Erros.* 2014. 186 f. Dissertação (Mestrado em Engenharia Elétrica) — Faculdade de Engenharia Elétrica e de Computação, Univerisidade Estadual de Campinas, Campinas, 2014.

ROCHA, A. S. L. et. al. DNA Sequences Generated by $\mathbb{Z}_4$-linear Codes. *IEEE International Symposium on Information Theory, ISIT2010*, v. 1, p. 1320–1324, 2010.

SENIOR, A. E.; WEBER, J. ATP synthesis driven by proton transport in F1F0-ATP synthase. *FEB SLetters*, v. 545, p. 61–70, 2003.

SGARBI, G. *et al.* Inefficient coupling between proton transport and ATP synthesis may be the pathogenic mechanism for NARP and Leigh syndrome resulting from the T8993G mutation in mtDNA. *Biochemical Journal*, v. 395, p. 493–500, 2006.

TRANCOSO, M. C. R. *Sestematização da codificação e discodificação de códigos BCH.* 1995. 128 f. Dissertação (Mestrado em Engenharia Eletrotécnica e de Computadores) — Departamento de Engenharia Eletrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, 1995.

TUSET, C. *et al.* Clinical and molecular aspects of hypogonadotropic hypogonadism congenital isolated. *Brazilian Archive Metabolic Endocrinology*, v. 8, p. 501, 2006.