UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
CAMPUS DE SÃO JOÃO DA BOA VISTA

MARIANA VENEZIAN MUSTO BASSI

Análise de sequências de DNA através de Códigos Corretores de Erros

São João da Boa Vista
2019

MARIANA VENEZIAN MUSTO BASSI

Análise de sequências de DNA através de Códigos Corretores de Erros

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia de Telecomunicações do Campus de São João da Boa Vista, Universidade Estatual Paulista "Júlio de Mesquita Filho", como parte dos requisitos para obtenção do diploma de Graduação em Engenharia de Telecomunicações .

Orientador: Profa. Dra. Cintya Wink de Oliveira Benedito

São João da Boa Vista
2019

Bassi, Mariana Venezian Musto

Análise de sequências de DNA através de códigos corretores de erros / Mariana Venezian Musto Bassi. -- São João da Boa Vista, 2019.
84 p. : il. color.

Trabalho de Conclusão de Curso – Câmpus Experimental de São João da Boa Vista – Universidade Estadual Paulista "Júlio de Mesquita Filho".
Orientadora: Profa. Dra. Cintya Wink de Oliveira Benedito
Bibliografia
1. Código genético 2. Códigos corretores de erros (Teoria da informação) 3. DNA Análise 4. Telecomunicações

CDD 23. ed. – 621.382

Ficha catalográfica elaborada pela Biblioteca-BJB
Bibliotecário responsável: João Pedro Alves Cardoso – CRB-8/9717

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
CÂMPUS EXPERIMENTAL DE SÃO JOÃO DA BOA VISTA
GRADUAÇÃO EM ENGENHARIA DE TELECOMUNICAÇÕES

TRABALHO DE CONCLUSÃO DE CURSO

ANÁLISE DE SEQUÊNCIAS DE DNA ATRAVÉS DE CÓDIGOS CORRETORES DE ERROS

Aluno: Mariana Venezian Musto Bassi

Orientador: Prof.ª Dr.ª Cintya Wink de Oliveira Benedito

Banca Examinadora:

- Cintya Wink de Oliveira Benedito (Orientador)
- Edgar Eduardo Benitez Olivo (Examinador)
- José Augusto de Oliveira (Examinador)

A ata da defesa com as respectivas assinaturas dos membros encontra-se no prontuário do aluno (Expediente nº 43/2018)

São João da Boa Vista, 28 de janeiro de 2019

Aos meus segundos pais, carinhosamente chamados de vô Bassi (*in memoriam*) e Tchô (*in memoriam*), por todo amor e dedicação a nossa família.

AGRADECIMENTOS

À Deus pelo seu amor incondicional, que me ensina a ter fé e força para trilhar cada dia e superar cada dificuldade encontrada.

Aos meus avôs, que mesmo já fazendo morada no céu, são minhas inspirações. Guardo todo amor e ensinamento em meu coração.

À minha família por todo amor e carinho, estando sempre ao meu lado, me incentivando e apoiando em todas as etapas da minha vida.

À minha orientadora, Profª. Dra. Cintya Wink de Oliveira Benedito, por toda disposição, conhecimento, paciência e compreensão durante os meses de desenvolvimento deste trabalho.

Aos membros da banca examinadora pela disponibilidade e atenção dispensada ao trabalho, assim como, por suas valiosas sugestões.

Ao Prof. Dr. Reginaldo Palazzo Júnior, Dra. Luzinete Cristina Bonani de Faria e Mestre Diogo Guilherme Pereira por compartilharem todos os seus conhecimentos sobre este trabalho, permitindo sua realização. Além de toda a atenção e disponibilidade em ajudar.

À todos os professores e funcionários da Unesp SJBV pelos conselhos e acolhimento, me fazendo sentir querida por todos.

Aos meus amigos, por todos os momentos vividos. Gratidão pelo apoio, amizade e compreensão quando precisei me ausentar.

Aos meus colegas que estiveram ao meu lado em algum momento dessa caminhada, me trazendo grandes reflexões e aprendizado.

Gratidão!

“Leve na sua memória para o resto de sua vida as coisas boas que surgiram no meio das dificuldades. Elas serão uma prova de sua capacidade em vencer as provas e lhe darão confiança na presença divina, que nos auxilia em qualquer situação, em qualquer tempo, diante de qualquer obstáculo.”

(CHICO XAVIER)

RESUMO

A teoria da informação e codificação, bem como, a genética preocupam-se com a transferência e armazenamento de informações. Há décadas, os cientistas estudam o casamento dessas teorias, porém, há uma grande dificuldade em determinar uma estrutura matemática relacionada à estrutura do DNA (ácido desoxirribonucleico). No presente trabalho, baseado em um modelo de sistema para importação genética proposto em [ROCHA 2010] através de códigos BCH (Bose-Chaudhuri-Hocquenghem) sobre a extensão de anel de Galois, implementamos um algoritmo capaz de identificar e reproduzir duas sequências de DNA, com funções biológicas distintas e comprimento de 63 nucleotídeos, utilizando para ambas os mesmos seis polinômios primitivos e geradores de grau 6. Para isso, precisamos associar as bases nitrogenadas do DNA (adenina, timina, guanina e citosina) aos elementos do alfabeto do anel finito $\mathbb{Z}_4 = \{0, 1, 2, 3\}$. Esse processo denomina-se rotulamento e, nas duas sequências analisadas de comprimento 63, aplicando um mesmo polinômio gerador, encontramos 8 palavras-código ambas com o mesmo tipo de rotulamento. Essas palavras-código distam um nucleotídeo da sequência original, onde as trocas de base nitrogenada ocorreram em posições distintas, ocasionando diferentes bases, códons e aminoácidos. O algoritmo também é capaz de analisar mutações em sequências de DNA. Para exemplificar esta aplicação, utilizamos a sequência relacionada ao éxon 14 do gene BRCA1 (*Breast Cancer* 1) com comprimento 127 analisando mutações pontuais *nonsense* e *missense* através de um polinômio gerador de grau 7. A partir das funções de identificação e reprodução, encontramos palavras-código para serem utilizadas como referência nessas análises. Posteriormente, aplicando cada mutação pontualmente, observamos que o código é capaz de recuperar a sequência original. Apontando uma estrutura matemática associada aos códigos corretores de erros para a fita simples de DNA, esse algoritmo pode contribuir para o desenvolvimento de uma metodologia que poderá reduzir o tempo e custos laboratoriais, auxiliar no diagnóstico de doenças, análises de mutações, produção de novos fármacos e melhoramento genético.

PALAVRAS-CHAVE: Códigos corretores de erros. Código genético. Análise mutacional. Código BCH sobre anéis.

ABSTRACT

Information and coding theory as well as genetics are concerned with the transfer and storage of information. For decades, scientists have studied the integration of these theories, but there is a great difficulty in determining a mathematical structure related to the structure of DNA (deoxyribonucleic acid). In the present work, based on a genetic import system model proposed in [ROCHA 2010] through BCH codes (Bose-Chaudhuri-Hocquenghem) on the Galois ring extension, we implemented an algorithm capable of identifying and reproducing two sequences of DNA, with different biological functions and length of 63 nucleotides, using for both the same six primitive polynomials and generators of degree 6. For this, we need to associate the nitrogen bases of the DNA (adenine, thymine, guanine and cytosine) to the elements of the alphabet of the finite ring $\mathbb{Z}_4 = \{0, 1, 2, 3\}$. This process is called labeling and, in the two results obtained, applying the same generator polynomial, we find 8 codewords with the same labeling. These codewords differ a nucleotide from the original sequence, where the exchanges of nitrogen base occurred in different positions, causing different bases, codons and amino acids. The algorithm is also capable of analyzing mutations in DNA sequences. To exemplify this application, we used the sequence related to exon 14 of the BRCA1 gene (Breast Cancer1) with length 127 analyzing nonsense and missense point mutations through a generation polynomial of degree 7. From the identification and reproduction functions, we find codewords to be used as reference in these analysis. Subsequently, applying each mutation punctually, we observe that the code is able to retrieve the original sequence. Pointing to a mathematical structure associated with error-correcting codes for single strand of DNA, this algorithm can contribute to the development of a methodology that can reduce laboratory time and costs, assist in disease diagnosis, mutation analysis, new drug production, and genetical improvement.

KEYWORDS: Error Correcting Codes. Genetic code. Mutational analysis. BCH code on rings.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

A	Adenina
aa	Aminoácido
ASK	*Amplitude Shift Keying*
BCH	Bose-Chaudhuri-Hocquenghem
BIC	*Breast Cancer Information Core*
BRCA1	*Breast Cancer 1*
C	Citosina
CCE	Código Corretor de Erro
DNA	Ácido desoxirribonucleico
FSK	*Frequency Shift Keying*
G	Guanina
Gaa	Sequência de aminoácidos gerada
Glb	Rotulamento gerado
GmR	Sequência de RNA mensageiro gerada
Gnt	Sequência de nucleotídeos gerada
mRNA	RNA mensageiro
NCBI	*National Center for Biotechnology Information*
Oaa	Sequência de aminoácidos
Olb	Rotulamento original
OmR	Sequência de RNA mensageiro
Ont	Sequência de nucleotídeos original
PSK	*Phase Shift Keying*
QAM	*Quadrature Amplitude Modulation*
Ref	Sequência de referência
RNA	Ácido ribonucleico

rRNA	RNA ribossômico
T	Timina
tRNA	RNA transportador

LISTA DE SÍMBOLOS

$\mathbb{A}$	Anel
α	Raíz do polinômio primitivo
β	Raíz do polinômio minimal e do polinômio gerador
$\mathbb{C}$	Conjunto dos números complexos
C	Código de bloco linear
$\mathbf{c}$	Palavra-código
$c(x)$	Polinômio código
d	Distância de projeto de um código BCH
d_{min}	Distância mínima do código
$\mathbf{e}$	Vetor erro
$e(x)$	Polinômio erro
$\mathbb{G}$	Grupo
$GF(p)$	Corpo finito de ordem p
$GF(p^m)$	Extensão do corpo de Galois
$GF(q)$	Corpo de Galois
$g(x)$	Polinômio gerador
G	Matriz geradora
$GR(p^k, m)$	Extensão do anel de Galois
$GR^*(p^k, m)$	Grupo multiplicativo
G_n	Subgrupo cíclico com n elementos
$h(x)$	Polinômio verificação de paridade
H	Matriz verificação de paridade
H^t	Matriz transposta verificação de paridade
$\mathbb{K}$	Corpo
k	Dimensão do código de bloco linear

m	Grau da extensão de Galois e grau do polinômio primitivo
$m(x)$	Polinômio mensagem
n	Comprimento da sequência ou comprimento do código
N	Alfabeto genético
p	Número primo
$p(x)$	Polinômio primitivo
P	Matriz dos rotulamentos
$\phi(x)$	Polinômio minimal
$\mathbb{Q}$	Conjunto dos números racionais
q	Potência de um número primo
$\mathbb{R}$	Conjunto dos números reais
$\mathbf{r}$	Vetor recebido
$r(x)$	Polinômio recebido
$\mathbf{s}$	Síndrome
$S(x)$	Polinômio síndrome
$\mathbb{Z}$	Conjunto dos números inteiros
$\mathbb{Z}_p$	Corpo finito, para p um número primo
$\mathbb{Z}_q$	Anel finito, para todo q
$*$	Operação binária

SUMÁRIO

1 INTRODUÇÃO

A teoria dos códigos corretores de erros surgiu na década de 1940 com o trabalho de Shannon [SHANNON 1948]. Esse estudo partiu da necessidade de detectar e recuperar uma mensagem recebida, dada a possibilidade desta ser diferente da mensagem transmitida pela fonte. Um código corretor de erro (CCE) é um mecanismo aplicado no codificador de canal de um sistema de comunicação digital (Figura 1), em que uma sequência de *bits* de informação tem seu comprimento acrescido em um número pré-determinado, chamado redundância para que quando essa informação for transmitida ou armazenada ao ser recuperada, seja possível detectar e corrigir possíveis erros. Essa ferramenta é muito utilizada em nosso cotidiano, quando há informação digitalizada presente, como ao assistir um programa de televisão ou navegar na Internet.

Nos sistemas de comunicação e armazenamento, há dois tipos de códigos corretores de erros muito utilizados, conhecidos como códigos de bloco e códigos convolucionais. Os códigos de bloco são classificados em lineares e não lineares. Este último não é muito utilizado em aplicações práticas, por isso não é estudado para este trabalho, onde o foco está no linear, o qual possui a classe dos códigos cíclicos de suma importância para a introdução dos códigos BCH.

A elucidação da estrutura do DNA e dos processos de replicação, transcrição e tradução possibilitaram avanços na biologia molecular. Essas descobertas proporcionaram o desenvolvimento em tecnologias de DNA recombinante e estimularam o surgimento das indústrias biotecnológicas. Além disso, enconrajaram o desenvolvimento de estudos interdisciplinares, como o presente trabalho que envolve a telecomunicação (códigos corretores de erros), a matemática (álbegra abstrata) e a biologia (molecular). Dessa forma, buscamos analogias entre um sistema biológico e um sistema de comunicação, visto que tanto a genética quanto a teoria de comunicação se preocupam com a transferência da informação.

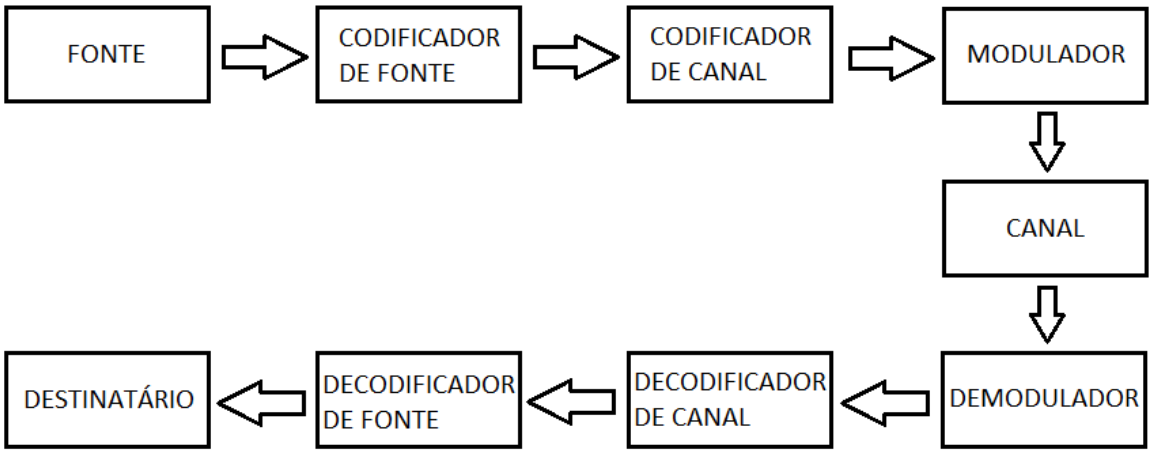
Primeiramente, consideremos um sistema de comunicação como sendo um conjunto de componentes que tem por objetivo transmitir uma informação gerada pela fonte a um receptor, através de um canal de comunicação onde erros poderão ser introduzidos à mensagem. Há dois tipos de sistemas: *i)* **analógico**, que conserva a forma do sinal durante toda sua transmissão e *ii)* **digital**, em que a forma do sinal pode ser diferente da original, variando em frequência e/ou amplitude e/ou fase em intervalos fixos de tempo.

A Figura 1 ilustra um diagrama de blocos de um modelo de sistema de comunicação digital.

Os componentes deste sistema são definidos da seguinte maneira:

- **Fonte**: gerador da informação a ser transmitida;
- **Codificador de fonte**: realiza a associação dos símbolos da informação gerada pela fonte com um determinado alfabeto, com o objetivo de melhorar a eficiência do sistema. Essa sequência é representada por *bit*s ou caso se utilize q sinais, por um alfabeto q-ário;
- **Codificador de canal**: adiciona redundâncias a sequência de saída do codificador de fonte, transformando-a em uma sequência codificada;

Figura 1 – Sistema de Comunicação Digital.

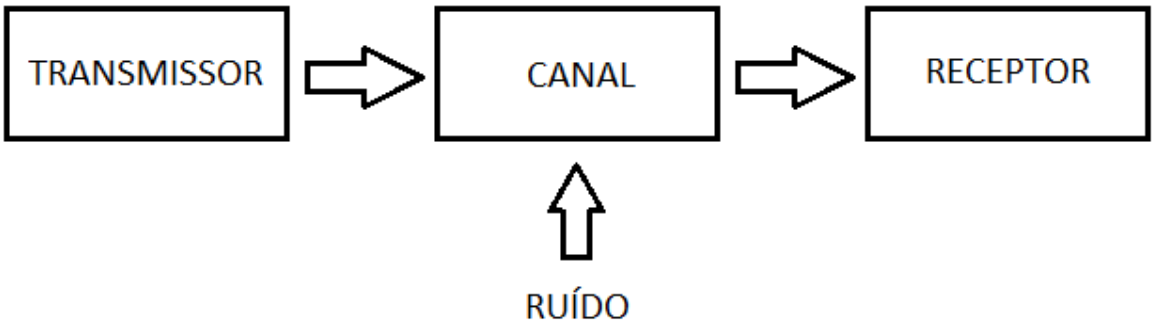


Fonte: Produção do próprio autor.

- **Modulador**: converte o sinal da saída do codificador de canal em uma forma de onda adequada para a transmissão através do canal. Algumas técnicas são: *i)* **ASK** (*Amplitude Shift Keying*), modulação em amplitude; *ii)* **FSK** (*Frequency Shift Keying*), modulação em frequência; *iii)* **PSK** (*Phase Shift Keying*), modulação em fase; *iv)* **QAM** (*Quadrature Amplitude Modulation*), modulação em amplitude e fase;
- **Canal**: meio físico que transmitirá a informação;
- **Demodulador**: a partir do sinal recebido do canal, estima-se o sinal transmitido e envia sua versão digital correspondente para o decodificador de canal;
- **Decodificador de canal**: produz uma estimativa do sinal enviado pelo demodulador, corrigindo possíveis erros;
- **Decodificador de fonte**: a partir da sequência de saída do decodificador de canal, estima-se uma sequência na saída da fonte.
- **Destinatário**: receptor da informação transmitida.

Analisando o diagrama da Figura 1, podemos representá-lo apenas considerando três blocos principais (transmissor, canal e receptor), como ilustrado na Figura 2. Essa simplificação é conhecida como o **dogma central da teoria de comunicação**.

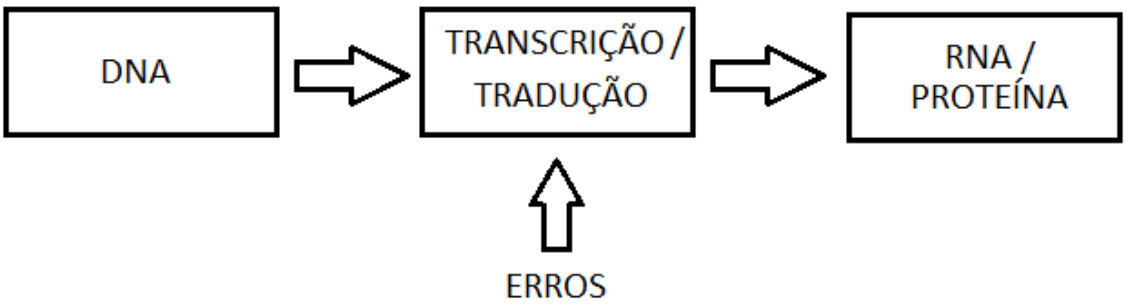
Figura 2 – Dogma central da teoria de comunicações.



Fonte: Produção do próprio autor.

Analogamente, como ilustrado na Figura 3, o dogma central da biologia molecular aborda que a partir da sequência de nucleotideos do DNA ocorrem os processos de transcrição e tradução que respectivamente, geram o RNA (ácido ribonucleico) e a proteína. Durante estes processos, há possibilidade de ocorrer erros que podem interferir tanto na formação das sequências de nucleotídeos do RNA quanto na leitura dos códons que formarão as proteínas (tais conceitos e processos serão apresentados na Seção 2.1). Portanto, o sistema biológico também transmite e armazena informações, podendo ser caracterizado por um diagrama de blocos.

Figura 3 – Dogma central da biologia molecular.



Fonte: Produção do próprio autor.

Dessa maneira, fazendo a analogia entre os sistemas, temos:

- **Transmissor**: DNA, porque para ocorrer o processo de transcrição, uma das fitas do DNA é utilizada como molde;
- **Canal**: processos de transcrição e tradução, porque nestas etapas, erros podem ocorrer e alterar a sequência de nucleotídeos;
- **Receptor**: compartimento intra ou extracelular, onde a proteína (informação gerada após a tradução) será transportada.

O estudo da aplicação de códigos corretores de erros para identificar e reproduzir sequências de DNA utilizando um código BCH sobre a extensão de anel de Galois foi iniciado em [ROCHA

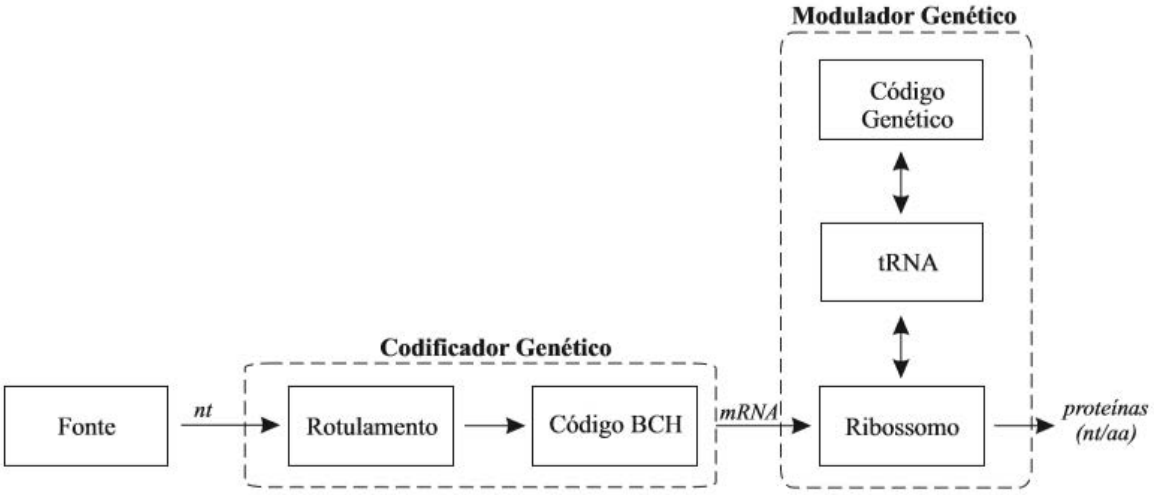
2010], onde focou-se na análise de sequências (do tipo direcionamento) com comprimento 63 e com a capacidade de correção de um erro. [ROCHA 2010] identificou que dentre as 24 possibilidades de associação (rotulamento) do alfabeto genético (adenina, timina, guanina e citosina) com o anel $\mathbb{Z}_4 = \{0, 1, 2, 3\}$, devido suas características geométricas distintas, os rotulamentos podes ser divididos em três tipos: os casos 2, 6, 7, 9, 16, 18, 20 e 23 foram denominados rotulamento A; os casos 1, 5, 8, 10, 15, 17, 19 e 24 rotulamento B e por fim, 3, 4, 11, 12, 13, 14, 21 e 22 rotulamento C. Os resultados fortaleceram a hipótese de existência de códigos concatenados na estrutura do DNA, sugerindo propostas de análises para diferentes sequências com comprimentos variados, assim como, outras estruturas matemáticas como os corpos.

Baseado nisso, [FARIA 2011] avançou com os estudos, analisando diferentes sequências de DNA (enzima, sinal interno, hormônio, íntron, DNA repetitivo, gene, entre outros) com comprimentos variados, utilizando códigos BCH sobre a extensão de corpo e de anel de Galois com diferentes graus, onde as sequências geradas diferiam em um e dois nucleotídeos sequência original. Além disso, caracterizou matematicamente o modelo de codificação genômico para identificação de sequências da dupla hélice do DNA, uma vez que em [ROCHA 2010] era apenas considerado a codificação genética (fita simples). Seus resultados também apontaram a identificação da existência de códigos concatenados na estrutura do DNA, em algumas sequências. Por fim, conhecido que a estrutura de corpo é mais inflexível que a estrutura de anel foi sugerido em [FARIA 2011] uma análise laboratorial detalhada das sequências de DNA reproduzidas sobre essas estruturas.

Já, em [PEREIRA 2014] assim como em [ROCHA 2010], para o código BCH foi utilizada a estrutura de anel para análise de sequências de DNA, mas com comprimentos maiores que 2047 nucleotídeos. Como verificou em [FARIA 2011] que a execução do programa computacional desenvolvido demandava muito tempo para sequências de DNA muito longas, [PEREIRA 2014] aprimorou o algoritmo introduzindo laços de repetição que envolve todos os polinômios primitivos; desenvolveu uma base de dados para que todas as informações geradas úteis fossem armazenadas e pudessem ser utilizadas quando necessário, sem ser preciso executar o programa novamente; assim como, aprimorou para que fosse possível a identificação de sequências de DNA resgatadas diretamente do repositório NBCI (*National Center for Biotechnology Information*). Além disso, parte de seu trabalho foi uma análise da sequência relacionada ao éxon 14 do gene supressor de câncer BRCA1 apresentado em [FARIA 2012], onde o desenvolvimento dos algoritmos dos programas de cálculo do polinômios geradores e do programa de análise de sequências de DNA foi apresentado em [PEREIRA 2013].

A base de nosso trabalho será um modelo de um sistema de comunicação para importação de proteínas organelares como proposto em [ROCHA 2010], conforme mostrado na Figura 4. Para demonstrar esse sistema, implementamos um algoritmo baseado em [FARIA 2011, PEREIRA 2014] que, por meio da utilização dos códigos BCH sobre a extensão de anel de Galois, tem como objetivo identificar e reproduzir sequências de nucleotídeos do DNA geradas pela fonte, permitindo análises de mutações. Para a análise biológica, reproduzimos as sequências de RNA mensageiro traduzindo os aminoácidos que geram a proteína, com o auxílio do código genético.

Figura 4 – Modelo de um sistema de comunicação para importação de proteínas organelares.



Fonte: [ROCHA 2010].

O presente trabalho está organizado da seguinte maneira: no Capítulo 2, elucidaremos os principais conceitos da biologia, álgebra e códigos corretores de erros. No Capítulo 3, descreveremos o algoritmo e suas etapas, exemplificando duas sequências de DNA com comprimento 63 nucleotídeos para compreender seu funcionamento. No Capítulo 4, mostraremos a aplicação deste algoritmo para análises mutacionais, utilizando a sequência de DNA com comprimento 127 relacionada ao éxon 14 do gene BRCA1. Por fim, no Capítulo, 5 apresentaremos as conclusões finais.

2 CONCEITOS INICIAIS

Como visto na introdução, o presente trabalho tem sua interdisciplinaridade na biologia molecular, álgebra abstrata e códigos corretores de erros. Este capítulo elucidará um pouco estes assuntos, com o objetivo de obter melhor entendimento para o desenvolvimento do trabalho. Na Seção 2.1, faremos uma breve revisão sobre a biologia molecular. Na Seção 2.2 apresentaremos ferramentas algébricas e por fim, na Seção 2.3 apresentaremos um estudo sobre os códigos corretores de erros.

2.1 BIOLOGIA MOLECULAR

Esta seção introduz os principais conceitos da biologia molecular abordados neste trabalho, os quais são indispensáveis para o seu desenvolvimento. Na Subseção 2.1.1 fazemos um breve resumo sobre os ácidos nucleicos e suas principais características. Na Subseção 2.1.2, elucidamos o dogma central da biologia molecular, com o objetivo de compreender os processos de replicação, transcrição e tradução. Na Subseção 2.1.3, abordamos sobre unidade básica da vida, a célula e sua conexão com as proteínas. Por fim, na subseção 2.1.4, apresentamos os tipos de mutações que as sequências de DNA podem sofrer.

Os conceitos apresentados nesta seção podem ser encontrados em [LODISH B.; MATSUDAIRA e ZIPURSKY 2005, ALBERTS B.; JOHNSON e WALTER 2005, FARIA 2011, ROCHA 2010, PEREIRA 2014].

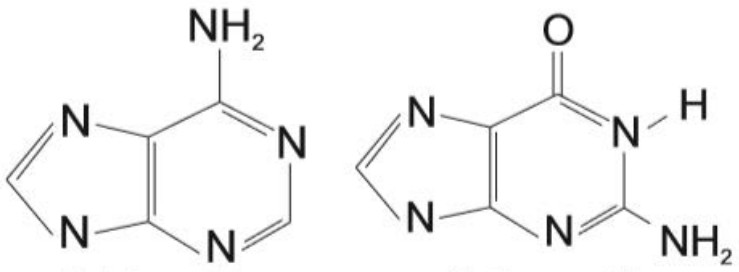
2.1.1 Ácidos Nucleicos

Os ácidos nucléicos são macromoléculas que contem o material genético responsável por inúmeras informações celulares como a **síntese proteica**, com a produção das proteínas no local e momento adequados, suas diferentes funções e sequências de aminoácidos, multiplicação celular, entre outras. Os dois principais tipos são: **DNA** e **RNA**. Essas moléculas são formadas por monômeros, denominados **nucleotídeos**, compostos por um grupo fosfato ligado através de uma ligação fosfodiéster a uma pentose (desoxirribose no DNA e ribose no RNA) que, por sua vez, está ligada a um anel (conhecido como base nitrogenada) contendo nitrogênio e carbono.

As bases nitrogenadas são estruturas cíclicas e são divididas em dois grupos:

- **Purina**: representado pela Adenina (A) e a Guanina (G). Estrutura com um anel fusionado, como ilustrado na Figura 5.
- **Pirimidina**: representado pela Timina (T), Citosina (C) e Uracila (U). Estrutura com um anel simples, como ilustrado na Figura 6

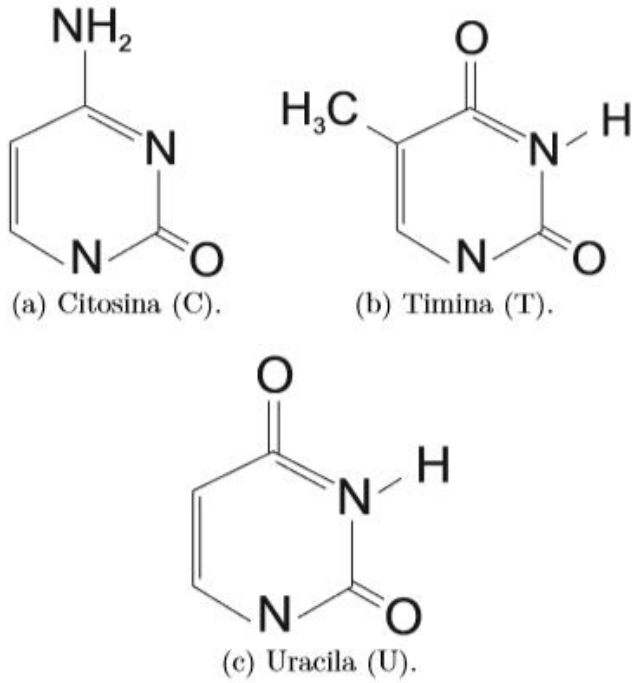
Figura 5 – Estrutura das purinas.



(a) Adenina (A). (b) Guanina (G).

Fonte: [PEREIRA 2014]

Figura 6 – Estrutura das pirimidinas.



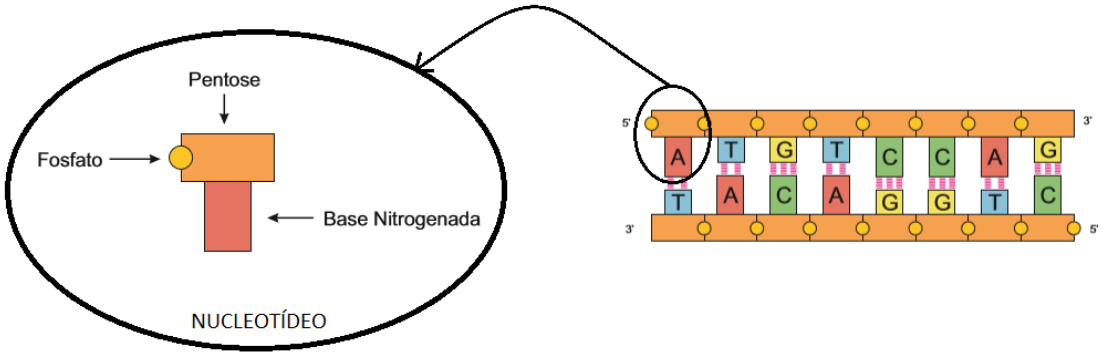
(a) Citosina (C). (b) Timina (T).

(c) Uracila (U).

Fonte: [PEREIRA 2014]

O DNA é formado por uma fita dupla em forma de espiral (ou dupla hélice) composta por unidades de nucleotídeos, como mostrado na Figura 7. Segundo o modelo de Watson e Crick, as duas fitas são antiparalelas e as bases de uma fita se ligam às bases de outra através de pontes de hidrogênio, respeitando a complementariedade biológica. Isto é, A se liga com T e, C com G.

Figura 7 – DNA.



Fonte: [PEREIRA 2014].

As parcelas das sequências de DNA que possuem informações genéticas são divididas em unidades funcionais denominadas **genes**. Os genes que possuem informações para a produção de proteínas podem ser divididos em duas regiões:

i) **codificante**, que determina a sequência de aminoácidos da proteína e;

ii) **reguladora**, que controla quando a proteína será produzida e em que tipo de célula.

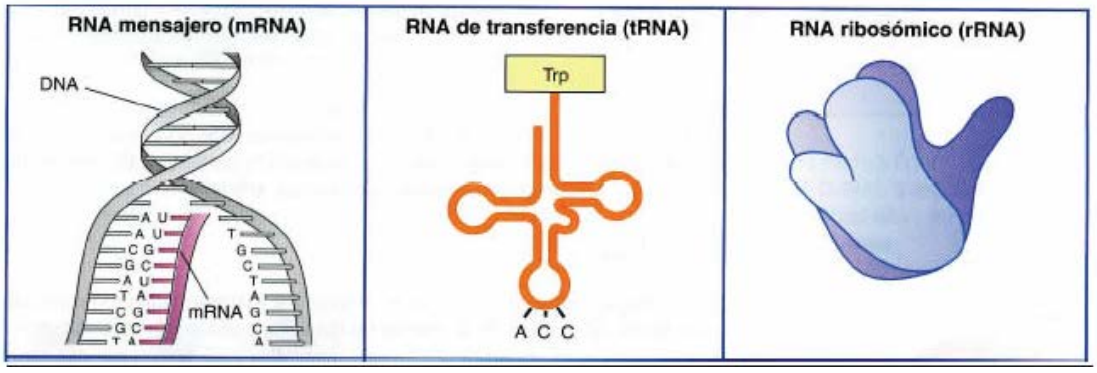
A partir do DNA, temos a formação do RNA que é composto por uma fita simples de nucleotídeos, em que a base nitrogenada adenina passará a se ligar com a uracila ao invés da timina, como ilustrado na Figura 8. Ou seja, a base complementar da adenina é a uracila e da guanina é a citosina. Há três tipos dessa molécula:

i) **RNA ribossômico (rRNA)**, principal constituinte dos ribossomos;

ii) **RNA mensageiro (mRNA)**, sequência de RNA resultante do processo de transcrição (explicado na próxima seção) que será transportado do núcleo da célula para o citoplasma para acontecer o processo de tradução (explicado posteriormente) junto ao ribossomo e;

iii) **RNA transportador (tRNA)**, transporta os aminoácidos referentes a sequência de bases do mRNA para ser formada a proteína.

Figura 8 – Tipos de RNA.

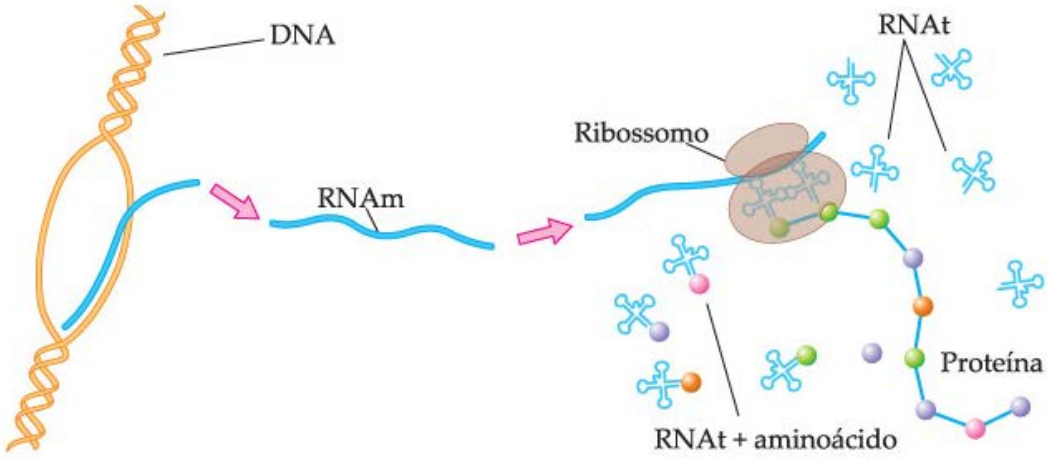


Fonte: [Biologia 2019].

2.1.2 Dogma Central da Biologia Molecular

Em 1958, Francis Crick descreveu o dogma central da biologia molecular explicando o fluxo da informação genética, como uma relação entre o DNA, RNA e as proteínas. Este dogma define o paradigma de que a informação é perpetuada através de três etapas: **replicação** do DNA, **transcrição** e **tradução**. Tais etapas são ilustradas na Figura 9 e descritas, de maneira sucinta, a seguir.

Figura 9 – Síntese Proteica.



Fonte: [Petrin 2018].

- **Replicação**:

  Neste processo é onde acontece a duplicação do DNA através dos seguintes passos:

  1. Rompimento das pontes de hidrogênio que unem as bases nitrogenadas da dupla hélice do DNA, auxiliado pela enzima **DNA helicase**. Essas fitas se separam e servem de moldes para a formação das moléculas filhas;
  2. Em cada uma das fitas separadas, há o encaixe de novos nucleotídeos de DNA livres existentes no núcleo celular, de acordo com sua complementariedade (A-T ou C-G). Para isso ocorrer, é necessária a presença da enzima **DNA polimerase**;
  3. Ao final do processo anterior, quando ambas as fitas estiverem completadas, há a formação de duas moléculas de DNA idênticas entre si.

  Ressalta-se que em cada molécula resultante existe um filamento pertencente à molécula-mãe, sendo o processo denominado **semiconservativo**. Além disso, as polimerases possuem a capacidade de verificarem erros resultantes do Passo 2, evitando ocasionalmente, mutações no genoma.

- **Transcrição**:

  Nesta etapa acontecerá a produção das moléculas de mRNA a partir do DNA. Essas moléculas migram para o citoplasma e controlam a síntese das proteínas. Esse processo segue os seguintes passos:

1. Rompimento das pontes de hidrogênio que unem as bases nitrogenadas da dupla hélice do DNA. Essas fitas se separam e **apenas uma** servirá de molde para a formação do mRNA;
2. Há o encaixe de nucleotídeos de RNA de acordo com sua complementariedade (A-U ou C-G) em uma única fita, denominada **fita ativa**. Assim como na replicação, é necessária a presença da enzima **RNA polimerase**;
3. A molécula de RNA se desprende da fita de DNA e migra para o citoplasma;
4. As duas fitas de DNA tornam-se a parear e reconstitui a molécula original.

Ressalta-se que essa enzima polimerase diferente do processo de replicação, não possui função revisora. Portanto, possíveis erros poderão acarretar ou não em mutações. Além disso, há o fenômeno de corte (*splicing*) alternativo que ocorre ainda no núcleo. Nesse processo, partes da sequência de bases não importantes (íntrons) para a proteína a ser formada são retiradas e há a combinação das partes restantes (éxons). Isto é, formam-se diferentes proteínas dependendo da combinação de íntrons a serem eliminados.

- **Tradução**:

Por fim, após a transcrição, ocorrerá a síntese da proteína localizada no ribossomo. A partir da sequência do mRNA, a cada três bases da fita única, um aminoácido é codificado. A correspondência entre as trincas de bases do DNA, RNA e os aminoácidos chamamos de código genético. Cada trinca é denominada códon e existem 64 códons que correspondem a 20 aminoácidos, apresentados na quarta coluna da Tabela 2. Portanto, mais de um códon pode corresponder ao mesmo aminoácido e o código é dito degenerado. Há o códon de iniciação (AUG - Metionina), que indica que a sequência de aminoácidos da proteína deve ser inicializada e, códons de finalização (UAA, UAG e UGA – stop), que indicam que a sequência deve ser finalizada. Os aminoácidos podem ser polares, não polares, básicos ou ácidos, de acordo com a Tabela 1.

Característica do aminoácido	**Aminoácidos**
Polar	Serina, treonina, tirosina, glutamina, asparagina, cisteína e glicina;
Não polar	Fenilalanina, leucina, isoleucina, metionina, valina, prolina, alanina e triptofano;
Básico	Histidina, lisina e arginina;
Ácido	Ácido aspártico e ácido glutâmico.

Tabela 1 – Tipos de Aminoácidos.

A Tabela 2 lista todos os aminoácidos e seus respectivos símbolos e abreviações, os quais serão utilizados no decorrer do trabalho.

Nome	Símbolo	Abreviação	Códons
Glicina	Gly, Gli	G	GGU, GGC, GGA e GGG
Alanina	Ala	A	GCU, GCC, GCA e GCG
Leucina	Leu	L	UUA, UUG, CUU, CUC, CUA e CUG
Valina	Val	V	GUU, GUC, GUA e GUG
Isoleucina	Ile	I	AUU, AUC e AUA
Prolina	Pro	P	CCU, CCC, CCA e CCG
Fenilalanina	Phe, Fen	F	UUU e UUC
Serina	Ser	S	UCU, UCC, UCA, UCG, AGU e AGC
Treonina	Thr, The	T	ACU, ACC, ACA e ACG
Cisteina	Cys, Cis	C	UGU e UGC
Tirosina	Tyr, Tir	Y	UAU e UAC
Asparagina	Asn	N	AAU e AAC
Glutamina	Gln	Q	CAA e CAG
Ácido aspártico	Asp	D	AAG e GAU
Ácido glutâmico	Glu	E	GAA e GAG
Arginina	Arg	R	CGU, CGC, CGA, CGG, AGA e AGG
Lisina	Lys, Lis	K	AAA e AAG
Histidina	His	H	CAU e CAC
Triptofano	Trp, Tri	W	UGG
Metionina	Met	M	AUG
Parada (*stop*)			UAA, UAG e UGA

Tabela 2 – Lista dos Aminoácidos.

Exemplo 2.1.1 *Considere a seguinte sequência de DNA "Mouse mRNA for T-cell receptor beta-chain V(beta)14-J(beta)2.2.", com 63 nucleotídeos, obtida no repositório NCBI (National Center for Biotechnology Information) disponível online, [NCBI 2018]. A seguir apresentamos a sequência de nucleotídeos original (**Ont**), obtida no NCBI e, as correspondentes sequências de RNA mensageiro (**OmR**) e de aminoácidos (**Oaa**):*

Ont: $TGT\ GCC\ TGG\ AGT\ CTA\ GCG\ GGG\ GAG\ CAG\ CTC\ TAC\ TTT\ GGT\ GAA\ GGC\ TCA\ AAG\ CTG\ ACA\ GTG\ CTG$

OmR: $ACA\ CGG\ ACC\ UCA\ GAU\ CGC\ CCC\ CUC\ GUC\ GAG\ AUG\ AAA\ CCA\ CUU\ CCG\ AGU\ UUC\ GAC\ UGU\ CAC\ GAC$

Oaa : $C\ \ A\ \ W\ \ S\ \ L\ \ A\ \ G\ \ E\ \ Q\ \ L\ \ Y\ \ F\ \ G\ \ E\ \ G\ \ S\ \ K\ \ L\ \ T\ \ V\ \ L$

Esta sequência será utilizada no Capítulo 4 para exemplificar a identificação e reprodução de sequências de DNA utilizando códigos corretores de erros.

2.1.3 A célula e as proteínas

A **célula** é um compartimento protegido do exterior por uma membrana plasmática, com interior aquoso, denominado **citoplasma**. Há dois tipos de célula:

i) **Procariótica**, que não possui núcleo definido e sua organização interna é simples. Como exemplo temos as bactérias, e;

ii) **Eucariótica**, que contem núcleo definido e diversos compartimentos internos, chamados organelas. Como exemplos temos as células dos reinos animal e vegetal.

A maior parte das propriedades funcionais e estruturais das células depende das **proteínas**, que são macromoléculas compostas por uma repetição de 20 aminoácidos interligados por ligações peptídicas, apresentados na Tabela 2. De acordo com o tipo de aminoácido que possui, do tamanho da cadeia polipeptídica e sua configuração espacial, as proteínas podem ter quatro tipos de estrutura: primária, secundária, terciária e quaternária. Quando a cadeia de aminoácidos é formada, ela se dobra em uma estrutura tridimensional característica e confere uma determinada função para a proteína. Isto é, somente quando uma proteína está na conformação correta, é capaz de funcionar eficientemente. Estas podem atuar em locais intra ou extracelulares. Além disso, possuem variadas e importantes funções como:

- **Estrutural**: participam dos tecidos dando-lhes consistência, elasticidade e rigidez. *Exemplos:* colágeno, queratina, albumina e outras;
- **Hormonal**: exerce função específica sobre algum órgão ou estrutura;
- **Energética**: obtenção de energia;
- **Enzimática**: capazes de catalisar reações bioquímicas. *Exemplo:* lipases;
- **Defesa**: proteção do organismo. *Exemplos:* anticorpos, fibrinogênio, trombina e outras;
- **Condutora de gases**: transporte de gases. *Exemplos:* hemoglobina e hemocianina.

2.1.4 Mutação

As proteínas possuem diversas funções que são essenciais para o correto funcionamento das células. A sua formação depende da sequência de DNA e suas bases nitrogenadas que serão utilizadas no processo da síntese proteica. Ocasionalmente, erros podem ocorrer por diversos meios, alterando essa sequência e causando assim, uma **mutação**. O DNA possui mecanismos de reparo capazes de corrigir a maior parte dessas alterações antes que se tornem permanentes, e muitos organismos conseguem eliminar as células somáticas que sofreram essas mudanças.

As mutações geram variações no conjunto de genes da população, sendo elas desfavoráveis (ou deletérias), que podem desenvolver proteínas parciais ou não funcionais ou; favoráveis (benéficas), que levam a novas versões de proteínas que auxiliam novas gerações a adaptarem-se melhor a mudanças em seu ambiente, ou seja, mudanças evolutivas adaptativas. A sequência pode ser alterada de diversas maneiras tanto em pequena escala, que afetam um pequeno gene em um ou poucos nucleotídeos, quanto em grande escala. Dessa forma, as mutações são classificadas em:

- **Pequena escala**:

  - **Mutação de ponto**: troca de um nucleotídeo por outro. Geralmente são causadas por substâncias mutagênicas ou erros na replicação do DNA. Essa mudança pode ser de dois tipos: *i)* **transição**, em que há a troca de uma purina por outra (A-G), ou de uma pirimidina por outra (C-T) e; *ii)* **transversão**, com a troca de uma purina por uma pirimidina, ou vice e versa (C/T - A/G). Quando essas mutações ocorrem dentro de uma região codificadora da proteína, estas podem ser classificadas em três tipos:
    * **Silenciosa**: o códon codifica para o mesmo aminoácido;
    * ***Missense* (sentido trocado)**: o códon codifica um aminoácido diferente, que não seja o de parada;
    * ***Nonsense* (sem sentido)**: codifica para um códon de parada que interrompe a proteína antes de seu término;
  - **Inserção**: adição de um ou mais nucleotídeos na sequência de DNA. Acréscimos na região codificadora de um gene podem alterar o splicing do mRNA ou causar alteração no quadro de leitura dos códons;
  - **Deleção**: remoção de um ou mais nucleotídeos da sequência de DNA. Essa exclusão pode modificar o quadro de leitura do gene e geralmente são irreversíveis. Ressalta-se que uma deleção não é o oposto de uma inserção;

- **Grande escala**:

  - **Amplificação**: criação de várias cópias de uma região cromossômica, aumentando a dosagem dos genes dentro dela;
  - **Deleção de regiões cromossômicas**: perda dos genes nessas regiões;
  - **Inserção**: une partes do DNA anteriormente separadas, resultando em genes fundidos funcionalmente distintos;
  - **Perda de heterogozidade**: perda de um dos dois alelos de um organismo, por deleção ou recombinação.

Exemplo 2.1.2 *O BRCA1 (breast cancer 1, early onset) é um gene supressor do câncer encontrado na posição q21 do cromossomo 17. O gene codifica uma proteína de 183 aminoácidos desempenhando um importante papel na regulação do ciclo celular e no controle de reparação do DNA. Estudos apontam que este foi o primeiro gene de predisposição ao câncer de mama mapeado, em 1994, no cromossomo humano, [HALL J. M.; LEE e KING 1990, FUTREAL P.A.; LIU 1994, MIKI Y.; SWENSEN 1994, LARSON J. S.; TONKINSON e LAI 1997, SOMASUNDARAM K.; ZHANG 1997, SCULLY R.; CHEN 1997, PEREIRA 2014].*

O câncer de mama é o tipo da doença mais comum entre as mulheres. No Brasil, esse tipo de câncer corresponde a cerca de 29% dos novos casos a cada ano. Em 2018, foram esperados 59.700 novos casos. Já em 2015, ocorreram 15.593 mortes, entre mulheres e homens. A detecção precoce, em grande parte dos casos, aumenta a possibilidade de tratamentos menos agressivos e com taxas de sucesso satisfatórias. Por esse motivo, a busca por métodos que auxiliem nesse diagnostico é importante, [INCA 2019].

Há mais de 600 mutações do BRCA1 descritas pelo repositório Breast Cancer Information Core (BIC), [BIC 2018, BRODY e BIESECKER 1998, BOERI L.; CANZONIERI e DANESINO 2011]. Mais de 85% das mutações conhecidas são do tipo translocações, missense ou nonsense, [SZABO e KING 1997]. As Tabelas 3 e 4 apresentam as mutações do tipo missense e nonsense do éxon 14 do gene BRCA1, constantes no repositório BIC [BIC 2018]. No Capítulo 4, analisaremos algumas destas mutações utilizando códigos corretores de erros.

Nucleotídeo	**Códon**	**Mudança de base**	**Mudança de a.a**	**Designação**	**Clinicamente Importante**
4521	1468	A para C	Asn para His	N1468H	desconhecido
4524	1469	C para T	Pro para Ser	P1469S	desconhecido
4529	1470	A para T	Glu para Asp	E1470D	desconhecido
4569	1484	T para A	Ser para Thr	S1484T	desconhecido
4573	1485	C para T	Thr para Ile	T1485I	desconhecido
4579	1487	A para G	Lys para Arg	K1487R	desconhecido
4599	1494	G para A	Glu para Lys	E1494K	desconhecido
4603	1495	G para A	Arg para Lys	R1495K	sim
4603	1495	G para T	Arg para Met	R1495M	sim

Tabela 3 – Mutações *Missense*.

Nucleotídeo	**Códon**	**Mudança de base**	**Mudança de a.a**	**Designação**	**Clinicamente Importante**
4489	1457	C para G	Ser para *Stop*	S1457X	sim
4491	1458	C para T	*Stop* 1458	4491C>T	desconhecido
4491	1458	C para T	Gln para *Stop*	Q1458X	sim
4508	1463	C para A	Tyr para *Stop*	Y1463X	sim
4599	1494	G para T	Glu para *Stop*	E1494X	sim
4603	1495	G para T	*Stop* 1462	4603G>T	sim

Tabela 4 – Mutações *Nonsense*.

2.2 ESTRUTURAS ALGÉBRICAS

Nesta seção, abordaremos conceitos de estruturas algébricas para uma operação, denominados grupos, e para duas operações, denominados anéis e corpos. Além disso, apresentaremos alguns conceitos sobre estruturas vetoriais utilizando corpos finitos. Estes conceitos desempenham um papel importante para a teoria de códigos corretores de erros permitindo a construção de códigos de bloco lineares e cíclicos, em especial os códigos BCH que será o código utilizado nas análises das sequências de DNA neste trabalho. Na Subseção 2.2.1 apresentaremos os conceitos de grupos, anéis e corpos juntamente com suas principais propriedades. Na Subseção 2.2.2, apresentaremos as construções de corpo e anél de Galois.

Os conceitos e resultados apresentados nesta seção podem ser encontrados em [GARCIA e LEQUAIN 2003, GONÇALVES 1999, RYAN e LIN 2009, HERSTEIN 1975, BENEDITO 2010]

2.2.1 Grupos, Anéis e Corpos

Definição 2.2.1 *Seja $\mathbb{G}$ um conjunto não vazio e uma operação binária $*$ sobre ele. $\mathbb{G}$ será chamado* ***grupo*** *se satisfazer os seguintes axiomas:*

1. *Associativo:* $(a * b) * c = a * (b * c)$*, para todo a, b, c* $\in \mathbb{G}$*;*

2. *Existência do elemento identidade, e: existe um* $e \in \mathbb{G}$ *tal que,* $a * e = e * a = a$*, para todo a* $\in \mathbb{G}$*;*

3. *Inverso: para todo a* $\in \mathbb{G}$*, existe um elemento* $a_0 \in \mathbb{G}$ *tal que,* $a * a_0 = a_0 * a = e$*.*

Denotamos um grupo $\mathbb{G}$ *com operação* $*$*, por* $(\mathbb{G}, *)$*. Se* $a * b = b * a$*, para todo* $a, b \in \mathbb{G}$*, a operação binária * é dita comutativa e o grupo é chamado de* ***abeliano*** *ou* ***comutativo****.*

Definição 2.2.2 *Um grupo* $\mathbb{G}$ *é definido como* ***finito*** *ou* ***infinito*** *se contem finitos ou infinitos elementos, respectivamente. O número total de elementos em um grupo finito é sua* ***ordem****.*

Exemplo 2.2.1 *Vamos verificar se o conjunto dos números inteiros,* $\mathbb{Z}$*, com a operação adição usual é um grupo abeliano. Para isso, é necessário que se safistaça os axiomas apresentados na Definição 2.2.1. De fato:*

1. *Associativo:* $(a + b) + c = a + (b + c)$*, para todo* $a, b, c \in \mathbb{Z}$*;*

2. *Elemento identidade:* $0 \in \mathbb{Z}$ *tal que,* $a + 0 = 0 + a = a$*, para todo* $a \in \mathbb{Z}$*;*

3. *Inverso: para todo* $a \in \mathbb{Z}$*, existem* $-a \in \mathbb{Z}$ *tal que,* $a + (-a) = (-a) + a = 0$*;*

4. *Comutativo:* $a + b = b + a$*, para todo* $a, b \in \mathbb{Z}$*.*

Portanto, $\mathbb{Z}$ *é um grupo abeliano.*

Definição 2.2.3 *Seja* $(\mathbb{G}, *)$ *um grupo. Um subconjunto não vazio* M *de um grupo* $\mathbb{G}$ *é um* ***subgrupo*** *de* $\mathbb{G}$*, se* M *é um grupo com a operação de* $\mathbb{G}$ *restritas a* M*. Denotamos* $M \leq \mathbb{G}$*.*

Teorema 2.2.1 *Seja* $\mathbb{G}$ *um grupo e seja* $a \in \mathbb{G}$*. Então,*

$$M = \{a^n \mid n \in \mathbb{Z}\},$$

é um subgrupo de $\mathbb{G}$ *e é o menor subgrupo de* $\mathbb{G}$ *que contém* a*, ou seja, qualquer outro subgrupo que contém* a*, contém também* M*.*

Definição 2.2.4 *Dados um grupo* $\mathbb{G}$ *e um elemento* $a \in \mathbb{G}$*, se ocorrer que:*

$$\mathbb{G} = \{a^n \mid n \in \mathbb{Z}\},$$

então, $\mathbb{G} = \langle a \rangle$ *é chamado de* ***grupo cíclico*** *e* a *é dito um gerador de* $\mathbb{G}$*.*

Definição 2.2.5 *Seja* $\mathbb{A}$ *um conjunto não vazio e duas operações binárias, adição "+" e multiplicação "*$\cdot$*", sobre ele.* $\mathbb{A}$ *será chamado* ***anel*** *se satisfazer os seguintes axiomas:*

1. *O conjunto* $\mathbb{A}$ *é um grupo abeliano em relação à operação adição;*

2. *A operação multiplicativa é associativa:* $a \cdot (b \cdot c) = (a \cdot b) \cdot c$*, para todo a, b, c* $\in \mathbb{A}$*;*

3. *Distributiva:* $a \cdot (b + c) = a \cdot b + a \cdot c$ *e* $(a + b) \cdot c = a \cdot c + b \cdot c$*, para todo a,b,c* $\in \mathbb{A}$*.*

Denotamos um anel $\mathbb{A}$ *com as operações "+" e "·" por* $(\mathbb{A}, +, \cdot)$*.*

Definição 2.2.6 *Com as condições descritas na Definição 2.2.5 temos que*

1. $\mathbb{A}$ *é um* ***anel comutativo*** *quando a multiplicação do anel* $\mathbb{A}$ *satisfaz* $a \cdot b = b \cdot a$*, para todo a, b* $\in \mathbb{A}$*, e*

2. $\mathbb{A}$ *é um* ***anel com unidade*** *quando a multiplicação pode admitir um elemento neutro, 1, em que* $1 \in \mathbb{A}$*, tal que,* $a \cdot 1 = 1 \cdot a = a$*, para todo a* $\in \mathbb{A}$*;*

3. $\mathbb{A}$ *é um* ***anel comutativo com unidade*** *quando é um anel cuja multiplicação é comutativa e que possui unidade.*

Exemplo 2.2.2 *Vamos verificar se o conjunto dos números inteiros,* $\mathbb{Z}$*, com as operações adição e multiplicação usuais é um anel comutativo com unidade. Para isso é necessário que se safistaça as condições apresentadas nas Definições 2.2.5 e 2.2.6.*

1. $\mathbb{Z}$ *é um grupo abeliano em relação à operação adição, como demonstrado no Exemplo 1.2.1;*

2. *Associativa:* $a \cdot (b \cdot c) = (a \cdot b) \cdot c$*, para todo a, b, c* $\in \mathbb{Z}$*;*

3. *Distributiva:* $a \cdot (b + c) = a \cdot b + a \cdot c$ *e* $(a + b) \cdot c = a \cdot c + b \cdot c$*, para todo a,b,c* $\in \mathbb{Z}$*;*

4. *Comutativo:* $a \cdot b = b \cdot a$*, para todo* $a, b \in \mathbb{Z}$*;*

5. *Unidade:* $1 \in \mathbb{Z}$ *tal que,* $a \cdot 1 = 1 \cdot a = a$*, para todo* $a \in \mathbb{Z}$*.*

Portanto, $\mathbb{Z}$ *é um anel comutativo com unidade.*

Definição 2.2.7 *Seja* $\mathbb{K}$ *um conjunto não vazio e duas operações binárias, adição "+" e multiplicação "." sobre ele.* $\mathbb{K}$ *será chamado* ***corpo*** *se satisfazer os seguintes axiomas:*

1. $\mathbb{K}$ é um grupo comutativo em relação à operação adição;

2. Associativa para adição e multiplicação: $(a + b) + c = a + (b + c)$ e $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ para todo $a, b, c \in \mathbb{K}$;

3. Comutativa para adição e multiplicação: $a + b = b + a$ e $a \cdot b = b \cdot a$ para todo $a, b \in \mathbb{K}$;

4. Elemento neutro para adição e multiplicação: $0 \in \mathbb{K}$ tal que, $a + 0 = a = 0 + a$ e $1 \in \mathbb{K}$ tal que $1 \cdot a = a \cdot 1 = a$ para todo $a \in \mathbb{K}$;

5. Inverso: $-a \in \mathbb{K}$ tal que, $a + (-a) = 0$ e $a^{-1} \in \mathbb{K}$ tal que, $a^{-1} \cdot a = a \cdot a^{-1} = 1$ para todo $a \in \mathbb{K}$;

6. Distributiva: $a \cdot (b + c) = a \cdot b + a \cdot c$ e $(a + b) \cdot c = a \cdot c + b \cdot c$ para todo $a, b, c \in \mathbb{K}$.

Denotamos um corpo $\mathbb{K}$ com as operações "+" e "$\cdot$" por $(\mathbb{K}, +, \cdot)$.

Definição 2.2.8 *A **ordem** do corpo é o número de elementos desse corpo. Quando a ordem é finita, o corpo é dito **finito**. Caso contrário o corpo é dito infinito.*

Exemplo 2.2.3 *Os conjuntos dos números racionais $\mathbb{Q}$, reais $\mathbb{R}$ e complexos $\mathbb{C}$, são exemplos de corpos.*

Exemplo 2.2.4 *O conjunto dos números inteiros $\mathbb{Z}$ não é um corpo, porque $2 \in \mathbb{Z}$, mas seu inverso $\frac{1}{2}$ não pertence a esse o conjunto.*

Exemplo 2.2.5 *Seja p um número primo. O conjunto $\mathbb{Z}_p = \{0, 1, 2, \ldots, p-1\}$ é um corpo finito. A seguir exemplificamos os corpos finitos $\mathbb{Z}_2$ e $\mathbb{Z}_3$, apresentamos suas tábuas da adição e multiplicação.*

- *$\mathbb{Z}_2$ = {0,1}: primeiramente, devemos lembrar que as operações são fechadas sobre o conjunto, isto é, o resultado tem que ser um elemento dele mesmo. Para isso, em $\mathbb{Z}_2$, utilizaremos as operações módulo-2 ($\mod 2$).*

  Para montar a tabela da adição, observe que $1 + 1 = 2$ e $2 \equiv 0 \mod 2$.

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

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

  Observamos que 1 é o elemento inverso tanto para adição quanto para multiplicação. Portanto, $\mathbb{Z}_2$ é um corpo.

- *$\mathbb{Z}_3$ = {0, 1, 2}*

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

·	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Tabela 6 – Tábua da adição e multiplicação de $\mathbb{Z}_3$.

  Observamos que todos os elementos tem inverso aditivo e multiplicativo. Portanto, $\mathbb{Z}_3$ também é um corpo.

Exemplo 2.2.6 *Observe através da Tabela 7 que para $\mathbb{Z}_4 = \{0, 1, 2, 3\}$, o elemento 2 não possui elemento inverso para a multiplicação, uma vez que na terceira linha ou coluna não há um resultado que seja o elemento identidade. Portanto, $\mathbb{Z}_4$ é um anel mas não é um corpo.*

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

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

Tabela 7 – Tábua da adição e multiplicação de $\mathbb{Z}_4$.

2.2.2 Corpo de Galois

Os corpos finitos são comumente chamados de corpos de Galois, em homenagem à Évariste Galois, seu descobridor. Primeiramente, seja p um número primo. O conjunto de inteiros $\{0, 1, \ldots, p-1\}$ forma um corpo finito, denotado por $GF(p)$, que tem ordem *p*, as operações de adição e multiplicação são realizadas $\mod p$ e, os elementos neutro da adição e multiplicação do corpo são 0 e 1, respectivamente. Além disso, para qualquer inteiro positivo *m*, pode-se construir um outro corpo finito $GF(p^m)$, denominado extensão do corpo de Galois, que contem $GF(p)$ como um subcorpo. Considerando $q = p^m$, iremos utilizar nas definições e resultados a seguir, $GF(q)$ como um corpo de Galois com *q* elementos, em que *q* é uma potência de um número primo.

Observação 2.2.1 *Seja a não nulo e pertencente a $GF(q)$, então {a^1, a^2, a^3 ...} também são elementos de $GF(q)$. Contudo, como este é um corpo finito, nem todas as potências de a podem ser distintas. Isto é, em um determinado ponto, há a repetição das potências, $a^m = a^k$, para $m > k$. Expressamos $a^m = a^k$ como $a^k \cdot a^{m-k} = a^k \cdot 1$. Usando a regra do cancelamento, obtemos: $a^{m-k} = 1$. Logo, para qualquer elemento não nulo de GF(q), existe pelo menos um inteiro positivo n que $a^n = 1$.*

Definição 2.2.9 *Seja a um elemento não nulo pertencente à GF(q). O menor inteiro positivo n tal que, $a^n = 1$, é dito como a ordem de $GF(q)$.*

Teorema 2.2.2 *Seja a um elemento não nulo de ordem n pertencente à $GF(q)$. Então, as potências de a:*

$$a^n = \{1, a, a^2, a^3, \ldots, a^{n-1}\},$$

constitui um subgrupo cíclico de um grupo multiplicativo $GF(q)$.

Definição 2.2.10 ***Elemento primitivo*** *é um elemento não nulo a, pertencente à $GF(q)$ de ordem $q-1$, dado que $\{a^{q-1}, a, a^2, \ldots, a^{q-2}\}$ forme todos os elementos não nulos de $GF(q)$.*

Agora, considere o polinômio

$$p(X) = p_0 + p_1 X + \cdots + p_n x^n,$$

onde $p_i \in GF(q)$, $0 \leq i \leq n$, com n inteiro não negativo e x uma variável sobre $GF(q)$. O grau do polinômio será definido como a maior potência de x com coeficiente *p* não nulo. Caso este coeficiente seja 1, o polinômio é dito **mônico**.

Definição 2.2.11 *Um polinômio $p(X)$ com grau m sobre $GF(q)$ é chamado de **irredutível** sobre $GF(q)$, se não for divisível por qualquer polinômio sobre $GF(q)$ que tenha grau maior que zero e menor que m.*

Teorema 2.2.3 *Qualquer polinômio irredutível $p(X)$ sobre $GF(q)$ com grau m divide $X^{q^m-1}-1$.*

Definição 2.2.12 *Um polinômio mônico e irredutível $p(X)$ sobre $GF(q)$ com grau m é dito **primitivo**, se $n = q^m - 1$ é o menor inteiro positivo tal que $p(x)$ divida $X^n - 1$.*

Para todo inteiro positivo m, existe um polinômio primitivo $p(X)$ de grau m sobre $GF(q)$. Os polinômios primitivos são fundamentais para a construção das extensões de Galois que veremos a seguir e dos códigos BCH que serão apresentados na Seção 2.3.2.1.

Primeiramente, a construção da extensão de Galois de um corpo $GF(p) = \{0, 1, \ldots, p-1\}$ se inicia com um polinômio primitivo de grau m sobre $GF(p)$

$$p(X) = p_0 + p_1 X + \cdots + p_{m-1}X^{m-1} + X^m.$$

Este polinômio terá m raízes e como $p(X)$ é irredutível, tais raízes não pertencem a $GF(p)$ e sim devem pertencer a uma extensão de $GF(p)$, como um subcorpo.

Agora, seja α uma raiz de $p(X)$ e, considerando 0 e 1 os elementos neutro da adição e multiplicação de $GF(p)$, respectivamente. Definida a operação de multiplicação, a sequência de potências se forma da seguinte maneira:

$$\begin{aligned}
&0 \cdot 0 = 0,\\
&0 \cdot 1 = 1 \cdot 0 = 0,\\
&0 \cdot \alpha = \alpha \cdot 0 = 0,\\
&1 \cdot 1 = 1,\\
&1 \cdot \alpha = \alpha \cdot 1 = \alpha,\\
&\alpha^2 = \alpha \cdot \alpha,\\
&\alpha^3 = \alpha \cdot \alpha \cdot \alpha,\\
&\vdots\\
&\alpha^j = \alpha \cdot \alpha \ldots \alpha \text{ (j vezes)}.
\end{aligned}$$

A partir dessa sequência concluímos que,

$$\begin{aligned}
&0 \cdot \alpha^j = \alpha^j \cdot 0 = 0,\\
&1 \cdot \alpha^j = \alpha^j \cdot 1 = \alpha^j,\\
&\alpha^i \cdot \alpha^j = \alpha^j \cdot \alpha^i = \alpha^{i+j}.
\end{aligned}$$

Baseado nisso, considerando a raiz de $p(X)$ e aplicando o Teorema 2.2.3, obtemos:

$$p(\alpha) = p_0 + p_1\alpha + \cdots + p_{m-1}\alpha^{m-1} + \alpha^m = 0 \implies \alpha^{p^m-1} - 1 = q(\alpha) \cdot p(\alpha) = q(\alpha) \cdot 0.$$

E, considerando q(α) um polinômio sobre $GF(p)$ e que $q(\alpha) \cdot 0 = 0$, segue que

$$\alpha^{p^m-1} - 1 = 0.$$

Somando o elemento unitário em ambos os lados temos que,

$$\alpha^{p^m-1} = 1. \tag{2.1}$$

Portanto, a sequência de potências se repete para $\alpha^{k^{(p^m-1)}}$, onde $k = 1, 2, \ldots$, excluindo o elemento zero. O conjunto de tais potências forma um corpo com p^m elementos distintos, denotado por

$$GF(p^m) = F = \{0, 1, \alpha, \ldots, \alpha^{p^m-2}\}.$$

Temos que $GF(p^m)$ contem $GF(p)$ como um subcorpo. Logo, se $GF(p^m)$ for construído a partir de $GF(p)$ e de um polinômio primitivo mônico sobre $GF(p)$, então $GF(p^m)$ é chamado de **extensão de corpo** de $GF(p)$. Ressalta-se que, qualquer polinômio primitivo de grau *m* resulta em uma extensão de corpo isomorfo a $GF(p^m)$.

Exemplo 2.2.7 *Seja $GF(2) = \{0, 1\}$ um corpo finito. Vamos construir a extensão de Galois $GF(2^5)$ de $GF(2)$. Sabendo que $m = 5$, considere $p(X) = 1 + X^2 + X^5$ um polinômio primitivo de grau 5. Se α é uma raiz de $p(X)$, temos que*

$$p(\alpha) = 1 + \alpha^2 + \alpha^5 = 0 \implies \alpha^5 = 1 + \alpha^2. \tag{2.2}$$

Agora, devemos encontrar qual a potência de α cuja sequência começará a se repetir. Pela Equação (2.1) segue que

$$\alpha^{2^5-1} = \alpha^{31} = 1. \tag{2.3}$$

Utilizando (2.2) e (2.3), podemos construir os demais elementos da extensão $GF(2^5)$. Como exemplos,

$$\begin{aligned} \alpha^6 &= \alpha \cdot \alpha^5 = \alpha \cdot (1 + \alpha^2) = \alpha + \alpha^3, \\ \alpha^7 &= \alpha^2 \cdot \alpha^5 = \alpha^2 \cdot (1 + \alpha^2) = \alpha^2 + \alpha^4 \end{aligned}.$$

A Tabela 8 apresenta todos os elementos de $GF(2^5)$ em três diferentes representações: potência de α, representação polinomial e representação vetor.

Potência	Polinômio	Vetor
0	0	(00000)
1	1	(10000)
α	α	(01000)
α^2	α^2	(00100)
α^3	α^3	(00010)
α^4	α^4	(00001)
α^5	$1+\alpha^2$	(10100)
α^6	$\alpha+\alpha^3$	(01010)
α^7	$\alpha^2+\alpha^4$	(00101)
α^8	$1+\alpha^2+\alpha^3$	(10110)
α^9	$\alpha+\alpha^3+\alpha^4$	(01011)
α^{10}	$1+\alpha^4$	(10001)
α^{11}	$1+\alpha+\alpha^2$	(11100)
α^{12}	$\alpha+\alpha^2+\alpha^3$	(01110)
α^{13}	$\alpha^2+\alpha^3+\alpha^4$	(00111)
α^{14}	$1+\alpha^2+\alpha^3+\alpha^4$	(10111)
α^{15}	$1+\alpha+\alpha^2+\alpha^3+\alpha^4$	(11111)
α^{16}	$1+\alpha+\alpha^3+\alpha^4$	(11011)
α^{17}	$1+\alpha+\alpha^4$	(11001)
α^{18}	$1+\alpha$	(11000)
α^{19}	$\alpha+\alpha^2$	(01100)
α^{20}	$\alpha^2+\alpha^3$	(00110)
α^{21}	$\alpha^3+\alpha^4$	(00011)
α^{22}	$1+\alpha^2+\alpha^4$	(10101)
α^{23}	$1+\alpha+\alpha^2+\alpha^3$	(11110)
α^{24}	$\alpha+\alpha^2+\alpha^3+\alpha^4$	(01111)
α^{25}	$1+\alpha^3+\alpha^4$	(10011)
α^{26}	$1+\alpha+\alpha^2+\alpha^4$	(11101)
α^{27}	$1+\alpha+\alpha^3$	(11010)
α^{28}	$\alpha+\alpha^2+\alpha^4$	(01101)
α^{29}	$1+\alpha^3$	(10010)
α^{30}	$\alpha+\alpha^4$	(01001)
α^{31}	1	(10000)

Tabela 8 – Elementos de GF(2^5)

Além de α, $p(X)$ possui outras 4 raízes. Tais raízes podem ser obtidas, substituindo X em $p(X) = 1+X^2+X^5$ pelos elemento de $GF(2^5)$. Por exemplo, para α_8 temos utilizando a representação polinomial dada na Tabela 8 que

$$\begin{aligned} p(\alpha_8) &= 1+\alpha^{16}+\alpha^{40} = 1+\alpha^{16}+\alpha^9 \\ &= 1+(1+\alpha+\alpha^3+\alpha^4)+(\alpha+\alpha^3+\alpha^4) \\ &= (1+1)+(1+1)\alpha+(1+1)\alpha^3+(1+1)\alpha^4 \\ &= 0. \end{aligned}$$

Da mesma forma, α^2, α^4 e α^{16} são raízes de $p(X)$.

Agora, seja $\beta = \alpha^j$, para algum $\alpha_j \in GF(q^m)$. Vamos obter um polinômio importante que será utilizado na contrução de códigos BCH, chamado de polinômio minimal.

Teorema 2.2.4 *Sejam $f(X) = f_0 + f_1X + \cdots + f_kX^k$, um polinômio sobre $GF(q)$ e β um elemento da extensão de corpos $GF(q^m)$ de $GF(q)$. Se β é uma raiz de $f(X)$, então β^{q^t} é também uma raiz de $f(X)$, onde t é um inteiro não negativo.*

Os elementos $\beta^{q^t} = \{\beta, \beta^q, \beta^{q^2}, \cdots\}$ são chamados de **conjugados** de β.

Exemplo 2.2.8 *Considere o polinômio $f(X) = 1 + X^3 + X^5$ sobre $GF(2)$. Este polinômio tem α^{15} como raiz. De fato, substituindo α^{15} em $f(X)$ e utilizando a Tabela 8 segue que*

$$\begin{aligned} f(\alpha^{15}) &= 1 + \alpha^{45} + \alpha^{75} = 1 + \alpha^{14} + \alpha^{13} \\ &= 1 + (1 + \alpha^2 + \alpha^3 + \alpha^4) + (\alpha^2 + \alpha^3 + \alpha^4) \\ &= (1+1) + (1+1)\alpha^2 + (1+1)\alpha^3 + (1+1)\alpha^4 \\ &= 0. \end{aligned}$$

Pelo Teorema 2.2.4, os conjugados de α^{15} são

$$(\alpha^{15})^2 = \alpha^{30},\ \ (\alpha^{15})^{2^2} = \alpha^{60} = \alpha^{29},\ \ (\alpha^{15})^{2^3} = \alpha^{120} = \alpha^{27} \text{ e } (\alpha^{15})^{2^4} = \alpha^{240} = \alpha^{23}.$$

Note que $(\alpha^{15})^{2^5} = \alpha^{480} = \alpha^{15}$. Logo, $\alpha^{30}, \alpha^{29}, \alpha^{27}$ e α^{23} são as demais raízes de $f(X)$.

Definição 2.2.13 *Seja $GF(q^m)$ uma extensão de $GF(q)$ e $\beta \in GF(q^m)$. Então, se $\phi(X)$ é o polinômio mônico de menor grau sobre $GF(q)$ que tem β como raiz, será denominado **polinômio minimal** de β.*

Teorema 2.2.5 *O polinômio minimal $\phi(X)$ de um elemento β de um corpo é único e irredutível.*

Corolário 2.2.1 *Seja $p(X)$ um polinômio mônico irredutível sobre $GF(q)$ e β um elemento de $GF(q^m)$ com polinômio minimal $\phi(X)$. Se β for raiz de $p(X)$, então $p(X) = \phi(X)$.*

Teorema 2.2.6 *Seja $\phi(X)$ o polinômio minimal de um elemento $\beta \in GF(q^m)$. Se e for o menor inteiro positivo tal que $\beta^{q^e} = \beta$, então*

$$\phi(X) = \prod_{i=0}^{e-1} (X - \beta^{q^t}). \tag{2.4}$$

Se $e \leq m$, o grau do polinômio minimal de qualquer elemento β de $GF(q^m)$ é m ou menor.

Exemplo 2.2.9 *Considere a extensão de corpos $GF(2^5)$ sobre $GF(2)$. Seja $\beta = \alpha^3$. Pelo Teorema 2.2.4, os conjugados de β são*

$$\beta^2 = (\alpha^3)^2 = \alpha^6,\ \ \beta^{2^2} = (\alpha^3)^{2^2} = \alpha^{12},\ \ \beta^{2^3} = (\alpha^3)^{2^3} = \alpha^{24} \text{ e } \beta^{2^4} = (\alpha^3)^{2^4} = \alpha^{48} = \alpha^{17}.$$

Note que $\beta^{2^5} = (\alpha^3)^{2^5} = \alpha^{96} = \alpha^3$. *Assim,* $e = 5$ *e, pelo Teorema 2.2.6, o polinômio minimal de* $\beta = \alpha^3$ *é*

$$\phi(X) = (X - \alpha^3)(X - \alpha^6)(X - \alpha^{12})(X - \alpha^{17})(X - \alpha^{24}).$$

Multiplicando e utilizando a Tabela 8 obtemos

$$\phi(X) = 1 + X^2 + X^3 + X^4 + X^5.$$

Vimos que $GF(p) = \mathbb{Z}_p$, p primo, é um corpo finito e $GF(p^m)$ é uma extensão de Galois sobre $GF(p)$. Se ao invés de um corpo estivermos trabalhando com um anel $GF(q) = \mathbb{Z}_q$, onde $q = p^k$ com $k > 1$, podemos de modo análogo obter a partir de um polinômio primitivo sobre $GF(p)$, e consequentemente sobre $\mathbb{Z}_q$, uma extensão de anel, chamada de **anel de Galois** e denotada por $GR(p^k, m)$.

Exemplo 2.2.10 *Considere o anel* $\mathbb{Z}_4 = \{0, 1, 2, 3\}$ *e* $p(X) = X^6 + X + 1$ *um polinômio primitivo sobre* $GF(2)$ *e* $\mathbb{Z}_4$. *Então,* $GR(4, 6)$ *é uma extensão de anel de Galois sobre* $\mathbb{Z}_4$. *Na Seção 2.3.2.1, iremos mostrar a construção de um código BCH de comprimento 63 utilizando este anel e esta extensão, para então no Capítulo 3 associar este código com sequências de DNA de mesmo comprimento.*

2.3 CÓDIGOS CORRETORES DE ERROS

Os código corretores de erros são divididos em duas classes principais: **códigos de blocos** e **códigos convolucionais**. A classe de códigos que será utilizada neste trabalho será a classe dos códigos de bloco, lineares e cíclicos, que está fortemente fundamentada em estruturas algébricas e vetoriais. Estas estruturas são fundamentais, pois sistematizam os processos de codificação, decodificação e análise de desempenho dos códigos.

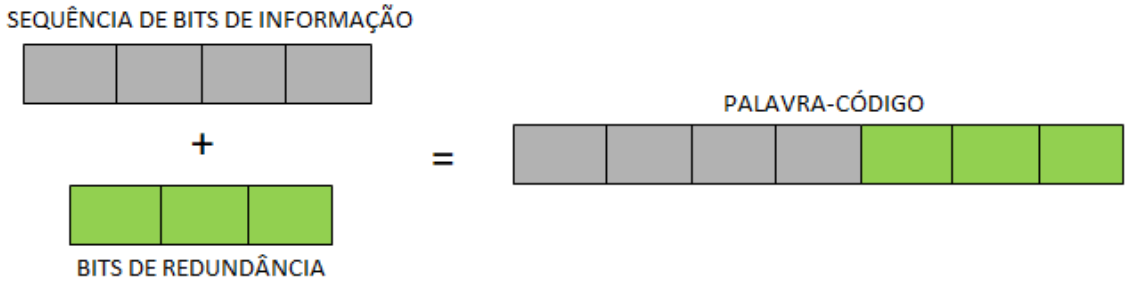
Na Subseção 2.3.1 apresentamos os códigos de bloco lineares e, na Subseção 2.3.2, os códigos de bloco cíclicos. Por fim, na subseção 2.3.2.1, abordaremos sobre os códigos BCH que são de interesse principal deste trabalho. Os conceitos apresentados nesta seção podem ser encontrados em [McWILLIANS e SLOANE 1977, PETERSON e Jr. 1972, INTERLANDO 1994, VITERBI e OMURA 1979, LIN e Jr. 1983, BARBOSA 2000, SHANKAR 1979].

2.3.1 Códigos de Blocos Lineares

Primeiramente, devemos relembrar como é esquematizado um sistema de comunicações digital, ilustrado na Figura 1. Os códigos de blocos serão utilizados no codificador do sistema e são ditos sem memória. Quando a sequência de *bits* de informações chega ao codificador, ela é segmentada em blocos de comprimento fixo, k e então, *bits* de redundância são adicionados, formando blocos de comprimento n, denominados palavras-código, como esboçado na Figura 10.

Definição 2.3.1 *Seja* $GF(q)$ *um corpo finito de q elementos. Dizemos que* C *é um* ***código de bloco linear*** *de comprimento* n *e dimensão* k, *se* C *for um subespaço vetorial de dimensão* k *de* $GF(q)^n$ *de* $GF(q)$. *A notação do código de bloco linear é* $C(n, k)$.

Figura 10 – Redundância.



Fonte: Produção do próprio autor.

Observação 2.3.1 *Se $q = 2$, é denominado código binário, se $q = 3$, código terciário, e generalizando, para um valor q qualquer o código é chamado q-ário.*

Observação 2.3.2 *Todas as palavras-código $\boldsymbol{cv} \in C$, possuem comprimento n. A quantidade de palavras-código é q^k. A razão*

$$R = \frac{k}{n},$$

é chamada de taxa do código.

Definição 2.3.2 *Seja $\boldsymbol{u} = (u_1, \ldots, u_n)$, $\boldsymbol{v} = (v_1, \ldots, v_n) \in GF(q)^n$. Definimos a **distância de Hamming**, $d_h(\boldsymbol{u}, \boldsymbol{v})$, como o número de coordenadas que este vetores diferem entre si. Isto é,*

$$d_h(\boldsymbol{u}, \boldsymbol{v}) = \#\{i \| u_i \neq v_i, 1 \leq i \leq n\}.$$

Exemplo 2.3.1 *Em $GF(2)^3$:*

$$d_h(001, 111) = 2,$$
$$d_h(000, 111) = 3,$$
$$d_h(100, 110) = 1.$$

Propriedade 2.3.1 *Sejam $\boldsymbol{u}, \boldsymbol{v} \in GF(q)^n$ então,*

1. $d_h(\boldsymbol{u}, \boldsymbol{v}) \geq 0$;
2. $d_h(\boldsymbol{u}, \boldsymbol{v}) = d_h(\boldsymbol{v}, \boldsymbol{u})$;
3. $d_h(\boldsymbol{u}, \boldsymbol{v}) \leq d_h(\boldsymbol{u}, \boldsymbol{w}) + d_h(\boldsymbol{w}, \boldsymbol{v})$.

Definição 2.3.3 *Seja C um código linear. A **distância mínima de Hamming** deste código é definida como*

$$d_h(C) = min\{d_h(\boldsymbol{u}, \boldsymbol{v})\},$$

para todo $u, v \in C$ e $u \neq v$.

Exemplo 2.3.2 *Seja o código* $C = \{0000, 0110, 1001, 1111\} \subset GF(2)^4$. *Temos que:*

$$\begin{aligned} d(0000, 0110) &= 2 \quad d(0110, 1111) = 2; \\ d(0110, 1001) &= 4 \quad d(0000, 1111) = 4; \\ d(0000, 1001) &= 2 \quad d(1001, 1111) = 2. \end{aligned}$$

Portanto, a distância mínima de Hamming deste código é $d_{min} = 2$.

Observação 2.3.3 *Um código de comprimento n, dimensão k e distância mínima* d_{min} *é chamado de um código linear* $C(n, k, d_{min})$.

Teorema 2.3.1 *Um código de bloco linear* $C(n, k, d_{min})$ *pode detectar* $(d_{min} - 1)$ *erros e corrigir até* $\lfloor \frac{(d_{min}-1)}{2} \rfloor$ *erros.*

Definição 2.3.4 *O peso de Hamming* $w_h(\boldsymbol{u})$ *para um* $\boldsymbol{u} \in GF(q)^n$ *é definido como o número de coordenadas diferentes de zero de* $\boldsymbol{u}$. *O peso mínimo de Hamming de* C, $w_h(C)$, *é o menor peso das palavras-código de* C.

Exemplo 2.3.3 *Considerando o mesmo código do Exemplo 2.3.2, encontraremos o peso mínimo de* C. *Temos que:*

$$\begin{aligned} w_h(0110) &= 2, \\ w_h(1001) &= 2, \\ w_h(1111) &= 4. \end{aligned}$$

Portanto, o peso mínimo de C *é igual a* $w_h(C) = 2$.

Observe que nos Exemplos 2.3.2 e 2.3.3 que

$$d_h(C) = w_h(C) = 2.$$

Isto acontece para todo código C como segue no resultado a seguir.

Teorema 2.3.2 *Seja* C *um código linear e* $\boldsymbol{u}, \boldsymbol{v} \in C$. *Então,*

1. $d_h(\boldsymbol{u}, \boldsymbol{v}) = w_h(\boldsymbol{u} - \boldsymbol{v})$;
2. $d_h(C) = w_h(C)$.

A seguir iremos definir a matriz geradora de um código de bloco linear C.

Definição 2.3.5 *Seja* $GF(q)$ *um corpo finito e* $C \subset GF(q)^n$ *um código de bloco linear com parâmetro* (n, k, d) *e* q^k *palavras-códigos. Seja* $B_C = \{G_0, G_1, \ldots, G_{k-1}\}$ *uma base de C e considere G a matriz cujas linhas são os vetores* $G_i = \{g_{i,0}, ..., g_{i,n-1}\}$, *i=1,...,k, ou seja,*

$$G = \begin{pmatrix} G_0 \\ G_1 \\ \vdots \\ G_{k-1} \end{pmatrix} = \begin{pmatrix} g_{0,0} & \cdots & g_{0,n-1} \\ \vdots & \ddots & \vdots \\ g_{k-1,0} & \cdots & g_{k-1,n-1}. \end{pmatrix}.$$

*Temos que G é chamada de **matriz geradora** do códico C.*

Exemplo 2.3.4 *A matriz*

$$G = \begin{pmatrix} 0110 \\ 1001 \end{pmatrix},$$

é a matriz geradora do código $C = \{0000, 0110, 1001\}$.

Observação 2.3.4 *Em geral, há mais de uma base de C e, consequentemente, a matriz geradora não é única. Portanto, qualquer escolha de base fornece uma matriz G.*

Seja $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1}) \in GF(q)^k$ a mensagem a ser codificada. A palavra-código $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}) \in C \subset GF(q)^n$ para essa mensagem é dada pela combinação linear de $G_0, G_1, \ldots, G_{k-1}$, com os k *bits* de **u** como os coeficientes. Obtemos **c** da seguinte maneira,

$$\mathbf{c} = u_0G_0 + u_1G_1 + \cdots + u_{k-1}G_{k-1}.$$

Assim, **c** pode ser expresso como:

$$\mathbf{v} = \mathbf{u} \cdot G.$$

onde, G é a matriz geradora de C, **u** é o código de fonte e **c** é o código de canal.

Exemplo 2.3.5 *Seja C um código $(5, 3)$ em $GF(2)^5$. Demonstraremos a codificação. Para construirmos G, é preciso definir uma base para C. Baseado nos valores de k e n, nossa base B_c possuíra 3 vetores com 5 bits. Dentre as 2^5 palavras possíveis, escolhemos 3 vetores que sejam linearmente independentes, portanto,*

$$B_c = \{10011, 01001, 00111\}.$$

Assim,

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

Temos que

$$\{000, 100, 010, 001, 110, 011, 101, 111\} \in F_2^3.$$

A codificação é dada por:

$$\boldsymbol{c} = \boldsymbol{u} \cdot G.$$

E então,

$$\begin{array}{ll}(000)\cdot G=(00000) & (010)\cdot G=(01001);\\ (110)\cdot G=(11010) & (101)\cdot G=(10100);\\ (100)\cdot G=(10011) & (001)\cdot G=(00111);\\ (011)\cdot G=(01110) & (111)\cdot G=(11101).\end{array}$$

Logo, o código será

$$C(5,3)=\{00000,11010,10011,01110,01001,10100,00111,11101\}.$$

Exemplo 2.3.6 *Suponhamos que* $\boldsymbol{v}=(10101)$ *seja a mensagem recebida e queremos identificar a mensagem enviada,* $\boldsymbol{u}$*. É preciso mostrar que existe um* $\boldsymbol{u}\in\mathbb{Z}_2^3$*, tal que,*

$$\boldsymbol{u}\cdot G=(10101).$$

Consideremos a matriz geradora,

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

Portanto,

$$(u_1u_2u_3)\cdot\begin{pmatrix}1&0&1&0&1\\1&1&0&1&0\\1&1&1&1&1\end{pmatrix}=(10101).$$

Realizando a multiplicação, obtemos um sistema linear com 3 *incógnitas e* 3 *equações:*

$$\begin{cases}u_1+u_2+u_3=1\\ u_2+u_3=0\\ u_1+u_3=1\end{cases}.$$

Temos que

$$\begin{array}{l}u_2=-u_3\\ u_1=1-u_3\end{array}.$$

Portanto,

$$1-u_3-u_3+u_3=1\longrightarrow u_3=0,u_2=0\ \textit{e}\ u_1=1$$

e a mensagem enviada foi $\boldsymbol{u}=(100)$*.*

Definição 2.3.6 *Uma matriz geradora* G *de um código* C *está na forma padrão (ou sistemática) se* $G=(I_k|P)$*, em que* I_k *é a matriz identidade de ordem* k *e* P *é uma matriz de ordem* $k\times(n-k)$*.*

Teorema 2.3.3 *Dado um código linear* C*, existe um código equivalente* C_d *cuja matriz geradora está na forma padrão.*

Exemplo 2.3.7 *Considere o código* $C\subset GF(3)^6$ *com a seguinte matriz geradora na forma padrâo*

$$G = \left(\begin{array}{cccc|cc} 1 & 0 & 0 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{array}\right).$$

Considerando que a mensagem enviada seja $\boldsymbol{u} = (2210)$*, a mensagem recebida será*

$$\boldsymbol{c} = \boldsymbol{u} \cdot G = (2210) \cdot \left(\begin{array}{cccc|cc} 1 & 0 & 0 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{array}\right) = (221002).$$

Observe que a mensagem $\boldsymbol{u} = (2210)$ *aparece no início da palavra codificada* $\boldsymbol{c} = (221002)$

Definição 2.3.7 *Seja* C *um código* (n, k)*. Se* $G = (I_k|P)$ *for uma matriz geradora de* C *na forma padrão, então* $H = (-P_t|I_{n-k})$ *é chamada de matriz controle de paridade de* C*, em que* P_t *é a transposta de* P *de ordem* $(n-k) \times k$ *e,* I_{n-k} *a matriz identidade de ordem* $n-k$*.*

A matriz H é utilizada na decodificação e na identificação de uma palavra **v** como palavra-código, pois como

$$G \cdot H^t = (I_k|P) \cdot (-P_t|I_{n-k}) = -P + P = 0,$$

se $\mathbf{v} \in C$, então,

$$\mathbf{v} = \mathbf{u} \cdot G \longrightarrow \mathbf{c} \cdot H_t = \mathbf{u} \cdot (G \cdot H^t) = 0,$$

ou seja, **v** é uma palavra código.

Exemplo 2.3.8 *Seja* C *um código binário* $(6, 3)$ *com a matriz geradora*

$$G = \left(\begin{array}{cccccc} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{array}\right).$$

Como matriz geradora se encontra na forma padrão, pela Definição 2.3.7, obtemos:

$$H = \left(\begin{array}{cccccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right).$$

Agora, vamos verificar se $\boldsymbol{v} = (100111)$ *é uma palavra código de* C*. Para isso, precisamos multiplicá-la pela* H^t*:*

$$\boldsymbol{v} \cdot H^t = (100111) \cdot \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (000).$$

Como a multiplicação resultou em um vetor nulo, a palavra recebida é uma palavra código, ou seja $\boldsymbol{v} \in C$.

Até esta etapa, entendemos como codificar uma mensagem para ser transmitida por um canal ruidoso. Apresentaremos como recuperar essa informação inicial ao ser entregue ao destinatário. Este processo é denominado decodificação, presente no decodificador no receptor, como mostrado na Figura 1.

Quando uma mensagem é transmitida por um canal ruidoso, erros podem ser acrescidos a ela. Essas alterações de um ou mais *bits* podem modificar a informação original, por isso a necessidade de codificá-la, tornando-a mais robustas a esses erros. O decodificador precisa conhecer o método utilizado no codificador para aplicá-lo novamente e recuperar a mensagem. Algumas das técnicas para detectar e corrigir esses erros serão apresentadas a seguir.

Um dos métodos de decodificação é por **máxima verossimilhança**, em que para decidir qual palavra código foi mais provavelmente transmitida, dado um vetor **r** recebido, é preciso escolher **c** como o vetor com a menor distância em relação a **r**. Analogamente, imaginemos um ponto central referente a palavra-código original, $\mathbf{c}_i$. Há uma região circunscrita a ele, onde se um ponto estiver contido nela se sabe identificar essa palavra código. Caso localiza-se fora, não.

Definição 2.3.8 *Seja* C *um código de bloco linear* (n, k) *com matriz controle de paridade* H. *Suponhamos que uma palavra-código* $\boldsymbol{c} \in C$ *é transmitida por um canal ruidoso e* $\boldsymbol{r}$ *seja o vetor recebido. Então, o vetor erro é definido por:*

$$\boldsymbol{e} = \boldsymbol{r} - \boldsymbol{c}.$$

Exemplo 2.3.9 *Em um dado código binário, se transmitirmos* $\boldsymbol{c} = (010011)$ *e recebermos* $\boldsymbol{r} = (101011)$. *Então, o erro introduzido foi:*

$$\boldsymbol{e} = (010011) - (101011) = (111000).$$

Ao receber a mensagem **r**, o decodificador precisa determinar onde ocorreram os erros no vetor, de acordo com seu parâmetro de distância do código, d_{min}. Considerando $d_{min} \geq 3$, assim como, na codificação quando verificamos se o vetor é uma palavra-código, também multiplicaremos **r** por H^t. Essa multiplicação é denominada **síndrome**,

$$\mathbf{s} = \mathbf{r} \cdot H^t.$$

Caso $\mathbf{s} = 0$, então $\mathbf{r}$ é uma palavra-código de C. Caso contrário, $\mathbf{r}$ não é uma palavra-código e é detectada a presença de erros na mensagem. Porém, há a possibilidade de $\mathbf{r}$ ser uma palavra-código diferente de $\mathbf{c}$. Quando isso acontece, o vetor $\mathbf{e}$ é idêntico a essa outra palavra-código não nula de C e dizemos que ocorreu um padrão de erro não detectável. Existem $2^k - 1$ desses erros.

Observação 2.3.5 *Consideraremos que, se $\boldsymbol{e} \cdot H^t = 0$ e $\boldsymbol{r} \cdot H^t = 0$, então, $\boldsymbol{r} \in C$ e $\boldsymbol{c} = \boldsymbol{r}$.*

Consideremos C, um código de bloco linear com distância mínima de $d_{min} \geq 3$, capaz de corrigir t erros, tal que $t = \lfloor \frac{d_{min}-1}{2} \rfloor$. Seja $\mathbf{e}$, o vetor erro introduzido na palavra-código $\mathbf{c}$ e $\mathbf{r}$, o vetor recebido. Caso $\mathbf{s} = 0$, então, $\mathbf{c} = \mathbf{r}$. Caso contrário, se houve um erro então, $\mathbf{e} = (0, 0, \ldots, a, 0, \ldots, 0)$, para $a \neq 0$ na i-ésima posição. Assim,

$$\mathbf{e} \cdot H^t = a \cdot h_i,$$

onde, h_i é a i-ésima coluna da matriz controle de paridade, H, do código C. Logo, no processo inverso, fazemos:

$$\mathbf{r} \cdot H^t = a \cdot h_i.$$

Então, consideramos o vetor erro como o vetor com todas as componentes nulas menos na i-ésima posição que teremos a. A seguir será apresentado as etapas para utilizar essa técnica.

- **Algoritmo para correção de 1 erro:**

  1. Calcule $s = \mathbf{r} \cdot H^t$;
  2. Se $\mathbf{s} = 0$, então $\mathbf{r} = \mathbf{c}$;
  3. Se $\mathbf{s} \neq 0$, compare $\mathbf{s}$ com as colunas de H.
  4. Se existirem i e a, tais que $\mathbf{s} = a \cdot h_i$, então $\mathbf{e} = (0, 0, \ldots, a, \ldots, 0)$ com a na i-ésima posição e 0 nas demais;
  5. Corrija r fazendo $\mathbf{c} = \mathbf{r} - \mathbf{e}$;
  6. Se a Etapa 4 não ocorrer, então há mais de um erro e esse algoritmo não poderá ser utilizado.

O exemplo a seguir, mostra a utilização do algoritmo para correção de um erro.

Exemplo 2.3.10 *Seja C um código com matriz verificação de paridade:*

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Se $\boldsymbol{r} = (10100)$ é o vetor recebido, demosntraremos como encontrar a palavra-código utilizando as etapas descritas acima.

- ***Etapa 1**: calcular a síndrome:*

$$\boldsymbol{s} = \boldsymbol{r} \cdot H^t = (10100) \cdot \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} = (010).$$

Como $s \neq 0$, passaremos para a Etapa 3:

- ***Etapa 3**: comparar **s** com as colunas de H.*

  A síndrome é igual a quarta coluna da matriz H, h_4.

- ***Etapa 4**: Temos que,*

$$\boldsymbol{s} = h_4.$$

  Então, $\boldsymbol{e} = (00010)$.

- ***Etapa 5**: corrigir **r** :*

$$\boldsymbol{c} = \boldsymbol{r} - \boldsymbol{e} = (10100) - (00010) = (10110).$$

  Portanto, a palavra-código correta é: (10110). Para conferirmos, basta multiplicarmos essa palavra por H^t e verificar se resulta em um vetor nulo.

$$\boldsymbol{c} \cdot H^t = (10110) \cdot \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (000).$$

  *Verificamos que **c** é uma palavra-código.*

2.3.1.1 Arranjo Padrão

O arranjo padrão é uma técnica que descreve a decodificação por máxima verossimilhança através da montagem de uma tabela utilizando as palavras-código, os vetores líderes e suas respectivas classes laterais.

Definição 2.3.9 *Seja C um código (n, k, d) sobre $GF(q)$. Para todo $\boldsymbol{v} \in GF(q)^n$ e $\boldsymbol{c} \in C$, o conjunto $\boldsymbol{v} + \boldsymbol{c}$ é chamado de **classe lateral** de **v**.*

Definição 2.3.10 *Todos os vetores de uma mesma classe lateral possui a mesma síndrome.*

Exemplo 2.3.11 *Seja $C = \{00000, 10110, 10101, 01011\}$ e $\boldsymbol{v} = (00010) \in F_2^5$. A classe lateral de **v** é dada por*

$$\boldsymbol{v} + \boldsymbol{c} = \{00010, 10100, 11111, 01001\}.$$

*Verificaremos que todos os vetores da classe lateral de **v** possuem a mesma síndrome. Seja*

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

Primeiramente, devemos construir a matriz H para obtermos assim, H^t. A matriz G esta na forma padrão $(I_2|P_{2\times 3})$. Logo,

$$H = (-P^t_{3\times 2}|I_3) = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

E então,

$$H^t = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Calculando as síndromes:

$$(00010) \cdot H^t = (010) \quad (10100) \cdot H^t = (010);$$
$$(11111) \cdot H^t = (010) \quad (01001) \cdot H^t = (010).$$

Logo, comprovamos que todas os vetores da classe $\boldsymbol{v} + C$ possuem a mesma síndrome.

Definição 2.3.11 *Seja $\boldsymbol{v} \in \boldsymbol{v} + \boldsymbol{c}$. $\boldsymbol{v}$ é chamado de **líder** da classe lateral.*

O arranjo padrão é uma forma simples de tabelar todas as classes laterais envolvidas em uma codificação. Seja C um código (n, k, d_{min}) sobre $GF(q)$ capaz de corrigir $\lfloor \frac{d_{min}-1}{2} \rfloor$ erros e sejam $(0, \mathbf{c}_2, \ldots, \mathbf{c}_{q^k})$ as palavras-códigos de C.

Na primeira linha da tabela colocamos todas as palavras-código. Na segunda, escolhemos $\mathbf{v}_1 \in GF(q)^n$ tal que, $d_h(\mathbf{v}_1, 0) = 1$ e o somamos com todas as palavras-código, formando sua classe lateral $(\mathbf{v}_1 + \mathbf{c})$. Esse processo é repetido com os demais vetores $\mathbf{v}_j \in GF(q)^n$, tais que, $d_h(\mathbf{v}_j, 0) = 1$. Finalizada essa etapa, passaremos aos vetores que distam 2 da palavra toda nula e assim por diante, até que todos os vetores possíveis apareçam uma única vez na tabela, como mostrado na Tabela 9.

0	$\mathbf{c}_2$	$\mathbf{c}_3$	$\ldots$	$\mathbf{c}_{q^k}$
$\mathbf{v}_1$	$\mathbf{v}_1 + \mathbf{c}_2$	$\mathbf{v}_1 + \mathbf{c}_3$	$\ldots$	$\mathbf{v}_1 + \mathbf{c}_{q^k}$
$\vdots$	$\vdots$	$\vdots$	$\ldots$	$\vdots$
$\mathbf{v}_j$	$\mathbf{v}_j + \mathbf{c}_2$	$\mathbf{v}_j + \mathbf{c}_3$	$\ldots$	$\mathbf{v}_j + \mathbf{c}_{q^k}$

Tabela 9 – Arranjo Padrão.

Quando a palavra é recebida, localize-a no arranjo padrão e decodifique-a como sendo a palavra-código correspondente à palavra-código que se encontra no topo daquela coluna.

Observação 2.3.6 *A quantidade de linhas da tabela será q^{n-k} e todos os vetores de uma mesma coluna estão mais próximos da palavra código do topo dessa coluna.*

Exemplo 2.3.12 *Considere o código binário $C(5, 2, 3)$ com matriz geradora, G. Construíremos a tabela de arranjo padrão.*

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

Para encontrarmos as palavras-código, precisamos fazer a combinação linear das linhas de G. Portanto,

$$C = \{00000, 10111, 01101, 11010\}.$$

A Tabela 10 terá $2^{5-2} = 2^3 = 8$ linhas.

00000	10111	01101	11010
00001	10110	01100	11011
00010	10101	01111	11000
00100	10011	01001	11110
01000	11111	00101	10010
10000	00111	11101	01010
00110	10001	01011	11100
00011	10100	01110	11001

Tabela 10 – Arranjo Padrão de $C(5, 2, 3)$.

Suponhamos que $\boldsymbol{r} = (11101)$ seja o vetor recebido. Temos que

$$\boldsymbol{s} = \boldsymbol{r} \cdot H^t = (11101) \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} = (111) \neq 0.$$

Como $\boldsymbol{s} \neq 0$, segue que $\boldsymbol{r} \notin C$. Buscando na Tabela 10, vemos que $\boldsymbol{r}$ se encontra na tabela na sexta linha e terceira coluna. Logo, a palavra-código correspondente será a palavra-código que se encontra no topo daquela coluna, ou seja, $\boldsymbol{c} = (01101)$.

2.3.2 Códigos Cíclicos

Os códigos cíclicos formam uma classe especial dos código de bloco linear. Dessa forma, muitos dos conceitos vistos para códigos de bloco lineares continuam sendo válidos para os códigos cíclicos. A seguir, apresentamos alguns conceitos sobre estes códigos e veremos um tipo especial destes códigos, que são os códigos BCH.

Definição 2.3.12 *Um código de bloco linear $C(n, k)$ será chamado de código cíclico se o deslocamento cíclico de cada palavra código pertencente a C resultar em outra palavra código também pertencente a C.*

Primeiramente, para analisar as propriedades dos códigos cíclicos, utilizamos a representação polinomial das palavras códigos. O polinômio

$$c(X) = c_0 + c_1X + \cdots + c_{n-1}X^{n-1},$$

é denominado **polinômio código** e possui grau menor ou igual a $n-1$. Existem 2^k polinômios código, tais que 2^k-1 polinômios são não nulos.

Em um código cíclico $C(n,k)$ existe um polinômio

$$g(X) = 1 + g_1X + g_2X^2 + \cdots + g_{n-k-1}X^{n-k-1} + X^{n-k},$$

não nulo, mônico e com grau $n-k$, denominado **polinômio gerador**. Todos os polinômios código $c(X) \in C$ são divisíveis por $g(X)$, ou seja, são múltiplos de $g(X)$.

Assim,

$$c(X) = m(X)g(X),$$

onde

$$m(X) = m_0 + m_1X + \cdots + m_{k-1}X^{k-1}.$$

representa o **polinômio mensagem** com grau menor ou igual a $k-1$ associado a mensagem a ser codificada.

Suponhamos que $m = (m_0, m_1, \ldots, m_{k-1})$ seja a mensagem a ser codificada. Multiplicando o polinômio referente a esta mensagem por X^{n-k}, obtemos:

$$X^{n-k}m(X) = m_0X^{n-k} + m_1X^{n-k+1} + \cdots + m_{k-1}X^{n-1}.$$

Dividindo essa multiplicação pelo polinômio gerador $g(X)$,

$$X^{n-k}m(X) = a(X)g(X) + b(X),$$

onde, $a(X)$ e $b(X)$ são os polinômios quociente e resto da divisão, respectivamente. Rearranjando a expressão acima, obtemos que $b(X) + X^{n-k}m(X)$ é divisível por $g(X)$. Portanto,

$$b(X) + X^{n-k}m(X) = a(X)g(X).$$

Baseado nisso, concluímos que,

$$c(X) = X^{n-k}m(X) + b(X). \tag{2.5}$$

O polinômio gerador também pode ser representado na forma matricial, $G_{k\times n}$, chamada de **matriz geradora** de C, assim como visto nos códigos de bloco lineares. A primeira linha da matriz são os coeficientes do polinômio, completada com zeros. As próximas $k-1$ linhas são obtidas pelo deslocamento cíclico a direita da primeira linha. Contudo, essa matriz não estará em sua forma sistemática, sendo necessárias operações entre as linhas, sem permutações de coluna.

$$G = \begin{pmatrix} g(X) \\ Xg(X) \\ X^2g(X) \\ \vdots \\ x^{k-1}g(X) \end{pmatrix} = \begin{pmatrix} g_0 & g_1 & g_2 & \dots & g_{n-k} & 0 & 0 & \dots & 0 \\ 0 & g_0 & g_1 & \dots & g_{n-k-1} & g_{n-k} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 0 & g_0 & g_1 & \dots & g_{n-k} \end{pmatrix}.$$

Exemplo 2.3.13 *Considere $C(7,4,3)$ um código cíclico com polinômio gerador $g(x) = 1 + x + x^3$. Codificaremos a mensagem $m = (1100)$. Primeiramente, identificamos os parâmetros do código: $n = 7$, $k = 4$ e $d = 3$. Em seguida, como estamos trabalhando com polinõmios, passaremos a mensagem para seu formato polinomial $m(x)$. Portanto,*

$$m(x) = 1 + x.$$

Substituindo esses valores na Equação 2.5, temos que o polinômio codificador é da seguinte maneira:

$$c(x) = x^{7-4}(1 + x) + b(x) = (x^3 + x^4) + b(x).$$

onde, $b(x)$ é o polinômio referente ao resto da divisão de $x^{n-k}m(x)$ por $g(x)$. Dessa forma, $b(x) = x^2 + 1$ e obtemos

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

Retornando a forma vetorial, a mensagem codificada é $c = (1011100)$.

Definição 2.3.13 *Se C é um código cíclico de comprimento n com polinômio gerador $g(X)$ então, o polinômio verificação de paridade de C será dado por:*

$$h(X) = \frac{x^n + 1}{g(X)} = h_0 + h_1X + \cdots + h_kx^k.$$

Assim como, para o polinômio gerador, $h(X)$ também pode ser representado na forma matricial, $H_{(n-k)\times n}$, chamada de **matriz verificação de paridade**.

$$H = \begin{pmatrix} h_0 & h_1 & h_2 & \dots & h_k & 0 & 0 & \dots & 0 \\ 0 & h_0 & h_1 & \dots & h_{k-1} & h_k & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 0 & h_0 & h_1 & \dots & h_k \end{pmatrix}.$$

Como um código cíclico também é um código de bloco linear, a decodificação pode ser realizada através do cálculo do vetor síndrome. Uma vez encontrado, deve-se associá-lo a um erro.

Suponhamos que a palavra código $C = (C_0, C_1, ..., C_{n-1})$ foi transmitida por um canal ruidoso e $r = (r_0, r_1, ..., r_{n-1})$ seja a palavra recebida. Na sua forma polinomial temos $r(x) = r_0 + r_1x + ... +$

$r_{n-1}x^{n-1}$. A síndrome de $r(x)$ será o resto da divisão de $r(x)$ por $g(x)$, ou seja,

$$r(x) = q(x)r(x) + S(x), \tag{2.6}$$

onde $S(x)$ é o polinômio síndrome. O polinômio síndrome $S(x)$ tem grau $n-k-1$ ou menor. Se $S(x) = 0$ dizemos que a palavra-recebida foi a palavra enviada, se $S(x) \neq 0$ ocorreu erro na transmissão. A síndrome da palavra-recebida é igual a síndrome do erro. Se,

$$r(x) = C(x) + e(x), \tag{2.7}$$

onde $e(x)$ é o polinômio erro, então o polinômio síndrome de $r(x)$ e $e(x)$ são iguais. Assim, basta conhecer a tabela da síndrome dos erros para corrigir a palavra recebida.

Exemplo 2.3.14 *Considere o mesmo código cíclico $C(7,4,3)$ com polinômio gerador $g(x) = 1 + x + x^3$ do exemplo anterior. Seja $r = (0110001)$ a palavra recebida, mostraremos sua decodificação. Em primeiro lugar, precisamos construir a tabela de síndromes. Sabemos que o comprimento da palavra codificada será 7, então, os vetores líderes da tabela também terão comprimento 7. Além disso, como a distância do código é igual a 3, o código é capaz de corrigir apenas um erro. Dessa forma, os líderes distarão 1 da palavra toda nula. A Tabela 11 apresenta os vetores e polinômios líderes, assim como, suas respectivas síndromes, obtidas como o resto da divisão de cada polinômio líder pelo polinômio gerador.*

Líder	**Polinômio líder**	**Polinômio síndrome**	**Síndrome**
0000000	0	0	000
1000000	1	1	100
0100000	x	x	010
0010000	x^2	x^2	001
0001000	x^3	$1+x$	110
0000100	x^4	$x+x^2$	011
0000010	x^5	$1+x+x^2$	111
0000001	x^6	$1+x^2$	101

Tabela 11 – Tabela das síndromes.

Calculando a síndrome da palavra recebida $r(x) = x + x^2 + x^6$, temos que

$$s(x) = \frac{r(x)}{g(x)} = 1 + x.$$

Como a síndrome não resultou igual a zero, sabemos que ocorreu um erro. Para identificarmos a posição que isso ocorreu, procuramos na Tabela 11 o erro referente a esta síndrome. Dessa maneira, obtemos

$$c(x) = r(x) + e(x) = (x + x^2 + x^6) + x^3 = x + x^2 + x^3 + x^6.$$

Logo, a palavra recebida corrigida é $c = (0111001)$.

2.3.2.1 Códigos BCH sobre anéis

Os códigos BCH formam uma importante classe de códigos cíclicos devido, principalmente, à simplicidade de seus processos de codificação e decodificação. Esse código também permite a múltipla correção de t erros. Dessa maneira, formam a classe de um dos códigos construtivos para canais onde os erros afetam os símbolos de forma independente. Contudo, quando o comprimento das palavras-código não é grande, existem bons códigos BCH, caso contrário, o desempenho destes é prejudicado devido às baixas taxas de transmissão. A seguir, fazemos algumas considerações sobre os códigos BCH e posteriormente, passamos à construção de tais códigos.

Os códigos de bloco lineares e cíclicos foram apresentados sobre corpos finitos $GF(q)$. Porém, podemos utilizar anéis ao invés de corpos para obter alguns códigos, como os códigos BCH. A principal diferença da construção de códigos sobre anéis para a construção de códigos sobre corpos, está no fato das raízes do polinômio gerador dos códigos cíclicos sobre anéis encontrarem-se na extensão do anel $\mathbb{Z}_q$, ao invés de serem encontradas na extensão do corpo $GF(q) = GF(p^k)$.

Definição 2.3.14 *Um código cíclico sobre $\mathbb{Z}_q$ com comprimento $n = q^m - 1$, onde $q = p^k$ e m é o grau da extensão de Galois, ou seja, o grau do polinômio primitivo, é denominado* ***código cíclico primitivo****.*

Vamos assumir que p e n são relativamente primos, isto é, o máximo divisor comum é um, denotado por $mdc(p, n) = 1$, pois assim garantimos que $(x^n - 1)$ não apresenta fatores quadráticos. Como queremos construir códigos cíclicos sobre anéis, o primeiro passo está relacionado a fatoração de $(x^n - 1)$ no grupo multiplicativo $GR^*(p^k, m)$ para obter o polinômio gerador do código $g(x)$.

Em geral, temos a seguinte definição para códigos BCH sobre corpos finitos $GF(p)$.

Definição 2.3.15 *Um código cíclico de comprimento n sobre $GF(p)$ é denominado um* ***código BCH*** *com distância de projeto d se o seu gerador $g(x)$ for o mínimo múltiplo comum dos polinômios minimais de*

$$\beta^l, \beta^{l+1}, \beta^{l+2}, \ldots, \beta^{l+d-2},$$

para l inteiro não negativo, onde β é um elemento primitivo de $(x^n - 1)$, em uma extensão $GF(p^r)$ de $GF(p)$.

Normalmente, consideramos $l = 1$, o que nos fornece o chamado código BCH no sentido estrito. Observamos ainda que, para obter os polinômios minimais de

$$\beta, \beta^2, \beta^3, \ldots, \beta^{2t},$$

utilizamos a Teorema 2.2.6. Agora, analogamente à Definição 2.3.14, podemos extender a definição de códigos BCH para anéis.

Definição 2.3.16 *Se $n = p^m - 1$, ou seja, se β for um elemento primitivo em $\mathbb{Z}_q$, então o código BCH é chamado* ***código BCH primitivo****.*

Podemos especificar a matriz verificação de paridade, H, para um código BCH da seguinte forma

$$H = \begin{pmatrix} 1 & \beta & \beta^2 & \dots & \beta^{n-1} \\ 1 & \beta^2 & (\beta^2)^2 & \dots & (\beta^2)^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \beta^{2t} & (\beta^{2t})^2 & \dots & (\beta^{2t})^{n-1} \end{pmatrix},$$

onde, os elementos β^i, $1 \leq i \leq 2t$ de H pertencem a $GR^*(p^k, m)$, e portanto, os coeficientes de β são tomados módulo n. Substituindo os elementos β^i pelos vetores linha de comprimento r correspondentes, formamos a matriz H sobre $\mathbb{Z}_q$.

Seja β um elemento primitivo. Se $\beta^{e_1}, \beta^{e_2}, \dots, \beta^{e_j}$ são raízes do polinômio gerador, $g(x)$, podemos gerar um código BCH com elementos de $\mathbb{Z}_q$. Esse polinômio será dado pelo mínimo múltiplo comum entre os polinômios minimais, $\phi_{e_i}(x)$, de β^{e_i},

$$g(x) = mmc(\phi_{e_1}(x), \phi_{e_2}(x), \dots, \phi_{e_j}).$$

Observação 2.3.7 *A construção de códigos BCH cíclicos sobre o anel $\mathbb{Z}_q$ reduz-se à escolha de elementos do $GR^*(p^k, m)$ para serem raízes do polinômio gerador $g(x)$.*

Teorema 2.3.4 *A distância de Hamming mínima de um código BCH satisfaz a relação:*

$$d_{min} \geq 2t + 1.$$

Mostraremos no exemplo a seguir como calcular o polinômio gerador $g(x)$ de um código BCH de comprimento $n = 2^m - 1$ e $d_{min} \geq 2t + 1$ sobre o anel $\mathbb{Z}_q$.

Exemplo 2.3.15 *Considere o polinômio primitivo $p(x) = x^6 + 1x^4 + 1x^3 + 1x^1 + 1$ sobre $\mathbb{Z}_4$ e a extensão do anel de Galois $GR(4, 6)$. Seja α uma raíz de $p(x)$. Temos que*

$$\alpha^6 + \alpha^4 + \alpha^3 + \alpha^1 + 1 = 0 \rightarrow \alpha^6 = -\alpha^4 - \alpha^3 - \alpha^1 - 1.$$

Porém, como estamos em $\mathbb{Z}_4$

$$\alpha^6 = 3\alpha^4 + 3\alpha^3 + 3\alpha^1 + 3.$$

Associando $f = (010000) = \alpha$, definimos as potências f^i's para $i > 6$, através da potenciação de f. Por exemplo,

$$\begin{aligned} f^7 &= \alpha^7 = \alpha \cdot \alpha^6 = 3\alpha^5 + 3\alpha^4 + 3\alpha^2 + 3\alpha; \\ f^8 &= \alpha^8 = \alpha \cdot \alpha^7 = 3\alpha^6 + 3\alpha^5 + 3\alpha^3 + 3\alpha^2 \\ &= 3(3\alpha^4 + 3\alpha^3 + 3\alpha^1 + 3) + 3\alpha^5 + 3\alpha^3 + 3\alpha^2 \\ &= 3\alpha^5 + 9\alpha^4 + 12\alpha^3 + 3\alpha^2 + 9\alpha^1 + 9 \ (mod 4) \\ &= 3\alpha^5 + \alpha^4 + 3\alpha^2 + \alpha^1 + 1. \end{aligned}$$

Assumindo a forma vetorial $(\alpha^0\alpha^1\alpha^2\alpha^3\alpha^4\alpha^5)$*, temos que:* $f^7 = \alpha^7 = (033033)$ *e* $f^8 = \alpha^8 = (113013)$. *A Tabela 12 apresenta os elementos desse grupo cíclico:*

1	(100000)	$f^{10} = \alpha^{10}$	(302031)
$f = \alpha^1$	(010000)	$f^{11} = \alpha^{11}$	(320133)
$f^2 = \alpha^2$	(001000)	$f^{12} = \alpha^{12}$	(102123)
$f^3 = \alpha^3$	(000100)	$\vdots$	$\vdots$
$f^4 = \alpha^4$	(000010)	$f^{121} = \alpha^{121}$	(201122)
$f^5 = \alpha^5$	(000001)	$f^{122} = \alpha^{122}$	(200332)
$f^6 = \alpha^6$	(330330)	$f^{123} = \alpha^{123}$	(200213)
$f^7 = \alpha^7$	(033033)	$f^{124} = \alpha^{124}$	(130131)
$f^8 = \alpha^8$	(113013)	$f^{125} = \alpha^{125}$	(303303)
$f^9 = \alpha^9$	(121011)	$f^{126} = \alpha^{126}$	(100000)

Tabela 12 – Elementos do grupo cíclico $GR^*(4, 6)$ em notação de $6 - uplas$.

A partir desses elementos, iremos construir um subgrupo cíclico do grupo $GR^*(4, 6)$ *com 63 elementos, conforme o valor da distância de projeto* d *do código BCH especificada por*

$$n \cdot d = 126 \rightarrow 63 \cdot d = 126 \rightarrow d = 2.$$

Como $d = 2$ *tomamos* $f^2 = (001000) = \alpha^2$ *para gerar um subgrupo cíclico de ordem 63 em* $GR^*(4, 6)$. *Assumindo,* $\beta = \alpha^2$ *como o elemento primitivo, encontramos os elementos constituintes desse subgrupo, apresentados na Tabela 13.*

$\beta^1 = \alpha^2$	(001000)	$\beta^2 = \alpha^4$	(000010)	$\beta^3 = \alpha^6$	(330330)
$\beta^4 = \alpha^8$	(113013)	$\beta^5 = \alpha^{10}$	(302031)	$\beta^6 = \alpha^{12}$	(102123)
$\beta^7 = \alpha^{14}$	(232212)	$\beta^8 = \alpha^{16}$	(310230)	$\beta^9 = \alpha^{18}$	(113212)
$\beta^{10} = \alpha^{20}$	(313000)	$\beta^{11} = \alpha^{22}$	(003130)	$\beta^{12} = \alpha^{24}$	(110101)
$\beta^{13} = \alpha^{26}$	(030130)	$\beta^{14} = \alpha^{28}$	(110011)	$\beta^{15} = \alpha^{30}$	(320023)
$\beta^{16} = \alpha^{32}$	(230031)	$\beta^{17} = \alpha^{34}$	(101003)	$\beta^{18} = \alpha^{36}$	(012021)
$\beta^{19} = \alpha^{38}$	(213333)	$\beta^{20} = \alpha^{40}$	(123210)	$\beta^{21} = \alpha^{42}$	(331122)
$\beta^{22} = \alpha^{44}$	(201113)	$\beta^{23} = \alpha^{46}$	(303312)	$\beta^{24} = \alpha^{48}$	(311301)
$\beta^{25} = \alpha^{50}$	(032102)	$\beta^{26} = \alpha^{52}$	(022303)	$\beta^{27} = \alpha^{54}$	(011230)
$\beta^{28} = \alpha^{56}$	(110222)	$\beta^{29} = \alpha^{58}$	(203300)	$\beta^{30} = \alpha^{60}$	(002033)
$\beta^{31} = \alpha^{62}$	(121101)	$\beta^{32} = \alpha^{64}$	(030200)	$\beta^{33} = \alpha^{66}$	(000302)
$\beta^{34} = \alpha^{68}$	(022021)	$\beta^{35} = \alpha^{70}$	(213033)	$\beta^{36} = \alpha^{72}$	(123211)
$\beta^{37} = \alpha^{74}$	(320111)	$\beta^{38} = \alpha^{76}$	(322120)	$\beta^{39} = \alpha^{78}$	(223001)
$\beta^{40} = \alpha^{80}$	(031223)	$\beta^{41} = \alpha^{82}$	(231103)	$\beta^{42} = \alpha^{84}$	(013322)
$\beta^{43} = \alpha^{86}$	(202331)	$\beta^{44} = \alpha^{88}$	(101122)	$\beta^{45} = \alpha^{90}$	(203213)
$\beta^{46} = \alpha^{92}$	(303333)	$\beta^{47} = \alpha^{94}$	(120110)	$\beta^{48} = \alpha^{96}$	(331131)
$\beta^{49} = \alpha^{98}$	(102010)	$\beta^{50} = \alpha^{100}$	(331310)	$\beta^{51} = \alpha^{102}$	(333203)
$\beta^{52} = \alpha^{104}$	(010303)	$\beta^{53} = \alpha^{106}$	(011110)	$\beta^{54} = \alpha^{108}$	(330001)
$\beta^{55} = \alpha^{110}$	(032333)	$\beta^{56} = \alpha^{112}$	(121000)	$\beta^{57} = \alpha^{114}$	(001210)
$\beta^{58} = \alpha^{116}$	(330302)	$\beta^{59} = \alpha^{118}$	(021321)	$\beta^{60} = \alpha^{120}$	(213022)
$\beta^{61} = \alpha^{122}$	(200332)	$\beta^{62} = \alpha^{124}$	(130131)	$\beta^{63} = \alpha^{126}$	(100000)

Tabela 13 – Elementos de G_{63}.

Determinados estes elementos, podemos calcular as raízes do polinômio gerador:

$$\{(\beta^i), (\beta^i)^2, (\beta^i)^{2^2}, (\beta^i)^{2^3}, (\beta^i)^{2^4}, (\beta^i)^{2^{5(mod\ 63)}}\}.$$

Dessa forma, obtemos $g(x)$ da seguinte maneira:

$$g(x) = mmc(\phi_1(x), \phi_2(x)), \ldots, \phi_{62}(x)), \tag{2.8}$$

onde, ϕ_i's são os polinômios minimais associados aos β^i's, para $i = 1, 2, \ldots 62$. Para construir um código BCH com $d_{min} = 3$, consideramos apenas os polinômios $\phi_1(x)$ e $\phi_2(x)$ tal que,

$$\phi_1(x) = \{(\beta^1), (\beta^1)^2, \ldots, (\beta^1)^{32(mod\ 63)}\} = \{\beta, \beta^2, \beta^4, \beta^8, \beta^{16}, \beta^{32}\};$$

$$\phi_2(x) = \{(\beta^2), (\beta^2)^2, \ldots, (\beta^2)^{32(mod\ 63)}\} = \{\beta^2, \beta^4, \beta^8, \beta^{16}, \beta^{32}, \beta\}.$$

Verificando que ambos os polinômios minimais possuem as mesmas raízes, o polinômio gerador será:

$$g(x) = (x - \beta^1)(x - \beta^2)(x - \beta^4)(x - \beta^8)(x - \beta^{16})(x - \beta^{32}). \tag{2.9}$$

Resolvendo a Equação 2.9, obtemos:

$$\begin{aligned} g(x) \;=\; & 1x^6 + 3\beta^{32}x^5 + 3\beta^{16}x^5 + 1\beta^{48}x^4 + 3\beta^8x^5 + 1\beta^{40}x^4 + 1\beta^{24}x^4 + 3\beta^{56}x^3 + 3\beta^4x^5 + 1\beta^{36}x^4 + \\ & 1\beta^{20}x^4 + 3\beta^{52}x^3 + 1\beta^{12}x^4 + 3\beta^{44}x^3 + 3\beta^{28}x^3 + 1\beta^{60}x^2 + 3\beta^2x^5 + 1\beta^{34}x^4 + 1\beta^{18}x^4 + \\ & 3\beta^{50}x^3 + 1\beta^{10}x^4 + 3\beta^{42}x^3 + 3\beta^{26}x^3 + 1\beta^{58}x^2 + 1\beta^6x^4 + 3\beta^{38}x^3 + 3\beta^{22}x^3 + 1\beta^{54}x^2 + \\ & 3\beta^{14}x^3 + 1\beta^{46}x^2 + 1\beta^{30}x^2 + 3\beta^{62}x^1 + 3\beta^1x^5 + 1\beta^{33}x4 + 1\beta^{17}x^4 + 3\beta^{49}x^3 + 1\beta^9x^4 + \\ & 3\beta^{41}x^3 + 3\beta^{25}x^3 + 1\beta^{57}x^2 + 1\beta^5x^4 + 3\beta^{37}x^3 + 3\beta^{21}x^3 + 1\beta^{53}x^2 + 3\beta^{13}x^3 + 1\beta^{45}x^2 + \\ & 1\beta^{29}x^2 + 3\beta^{61}x^1 + 1\beta^3x^4 + 3\beta^{35}x^3 + 3\beta^{19}x^3 + 1\beta^{51}x^2 + 3\beta^{11}x^3 + 1\beta^{43}x^2 + 1\beta^{27}x^2 + \\ & 3\beta^{59}x^1 + 3\beta^7x^3 + 1\beta^{39}x^2 + 1\beta^{23}x^2 + 3\beta^{55}x^1 + 1\beta^{15}x^2 + 3\beta^{47}x^1 + 3\beta^{31}x^1 + 1\beta^63. \end{aligned}$$

Colocando em evidência as potências de x:

$$\begin{aligned} g(x) \;=\; & x^6 + (3\beta^{32} + 3\beta^{16} + 3\beta^8 + 3\beta^4 + 3\beta^2 + 3\beta^1)x^5 + (1\beta^{48} + 1\beta^{40} + 1\beta^{24} + 1\beta^{36} + 1\beta^{20} + \\ & 1\beta^{12} + 1\beta^{34} + 1\beta^{18} + 1\beta^{10} + 1\beta^6 + 1\beta^{33} + 1\beta^{17} + 1\beta^9 + 1\beta^5 + 1\beta^3)x^4 + (3\beta^{56} + 3\beta^{52} + \\ & 3\beta^{44} + 3\beta^{28} + 3\beta^{50} + 3\beta^{42} + 3\beta^{26} + 3\beta^{38} + 3\beta^{22} + 3\beta^{14} + 3\beta^{49} + 3\beta^{41} + 3\beta^{25} + 3\beta^{37} + \\ & 3\beta^{21} + 3\beta^{13} + 3\beta^{35} + 3\beta^{19} + 3\beta^{11} + 3\beta^7)x^3 + (1\beta^{60} + 1\beta^{58} + 1\beta^{54} + 1\beta^{46} + 1\beta^{30} + 1\beta^{57} + \\ & 1\beta^{53} + 1\beta^{45} + 1\beta^{29} + 1\beta^{51} + 1\beta^{43} + 1\beta^{27} + 1\beta^{39} + 1\beta^{23} + 1\beta^{15})x^2 + (3\beta^{62} + 3\beta^{61} + 3\beta^{59} + \\ & 3\beta^{55} + 3\beta^{47} + 3\beta^{31})x^1 + 1\beta^{63}. \end{aligned}$$

Nesta etapa, é necessário somar os valores de β's, com o auxílio da Tabela 13. Exemplificamos para a

variável x^1:

$$x^1 : 3\beta^{62} + 3\beta^{61} + 3\beta^{59} + 3\beta^{55} + 3\beta^{47} + 3\beta^{31}.$$

Então, somando os valores de β's e aplicando módulo 4, temos que:

$3\beta^{31}$	(390393)
$+3\beta^{47}$	(600996)
$+3\beta^{55}$	(063963)
$+3\beta^{59}$	(096999)
$+3\beta^{61}$	(360330)
$+3\beta^{62}$	(363303)
$(mod4) =$	(300000)

O coeficiente de x^1 é 3. Realizando esse processo para os coeficientes das outras variáveis, encontramos que o polinômio gerador $g(x)$ é:

$$g(x) = 1x^6 + 2x^5 + 1x^4 + 1x^3 + 3x^1 + 1. \tag{2.10}$$

Portanto, $g(x)$ é um polinômio gerador de um código BCH com parâmetros (63,57,3).

Exemplo 2.3.16 *Considerando os procedimentos do exemplo anterior, podemos determinar outros polinômios geradores que podem ser utilizados para construir códigos BCH com parâmetros (63,57,3). No Capítulo 3, iremos utilizar tais polinômios para tentar identificar e reproduzir sequências de DNA de comprimento 63. Os seis polinômios primitivos que serão utilizados e seus respectivos polinômios geradores são:*

- $p_{01}(x) = 1x^6 + 1x^5 + 1x^2 + 1x^1 + 1$;
  - $g_{01}(x) = 1x^6 + 3x^5 + 2x^4 + 1x^2 + 1x^1 + 1$;
- $p_{02}(x) = 1x^6 + 1x^5 + 1x^4 + 1x^1 + 1$;
  - $g_{02}(x) = 1x^6 + 1x^5 + 1x^4 + 2x^2 + 3x^1 + 1$;
- $p_{03}(x) = 1x^6 + 1x^5 + 1x^3 + 1x^2 + 1$;
  - $g_{03}(x) = 1x^6 + 3x^5 + 1x^3 + 1x^2 + 2x^1 + 1$;
- $p_{04}(x) = 1x^6 + 1x^5 + 1$;
  - $g_{04}(x) = 1x^6 + 3x^5 + 2x^3 + 1$;
- $p_{05}(x) = 1x^6 + 1x^1 + 1$;
  - $g_{05}(x) = 1x^6 + 2x^3 + 3x^1 + 1$;
- $p_{06}(x) = 1x^6 + 1x^4 + 1x^3 + 1x^1 + 1$;
  - $g_{06}(x) = 1x^6 + 2x^5 + 1x^4 + 1x^3 + 3x^1 + 1$.

3 IDENTIFICAÇÃO DE SEQUÊNCIAS DE DNA VIA CÓDIGOS BCH

Neste capítulo iremos descrever um algoritmo para identificação de sequências de DNA. Na Seção 3.1, descreveremos as estapas e as funcionalidades do algoritmo. E, na Seção 3.2, mostraremos exemplos do uso do programa bem como a explicação dos seus passos para sequências de DNA de comprimento 63.

3.1 DESCRIÇÃO DO ALGORITMO

O algoritmo apresentado a seguir foi baseado nos algoritmos propostos em [FARIA 2011,PEREIRA 2014] e tem como objetivo identificar e reproduzir sequências de DNA. Tal algoritmo foi implementado utilizando o *software MatLab* e é capaz de reproduzir sequências de DNA com comprimentos ímpares na forma $n = 2^m - 1$, em que *m* é o grau de extensão de Galois, para que a fatoração de $x^n - 1$ na extensão de anel seja única. Para isso, será utilizado um código BCH sobre anéis com parâmetros (n, k, d_{min}) sobre a extensão de anel $GR(4, m)$ para gerar as palavras-código, associadas biologicamente, as sequências de RNA mensageiro.

Como vimos, as sequências de DNA são compostas por nucleotídeos podendo ter quatro tipos de bases: A (adenina), C (citosina), G (guanina) ou T (timina). Contudo, como na sequência de RNA é utilizada a U (uracila) no lugar da timina, o alfabeto genético será: N = {A, C, G, T/U}. Por esse motivo, utilizaremos o alfabeto 4-ário para o código BCH sobre o anel $\mathbb{Z}_4 = \{0, 1, 2, 3\}$, obedecendo as operações de adição e multiplicação módulo 4, apresentadas no Exemplo 2.2.5.

3.1.1 Etapas do algoritmo

Inicialmente, é preciso selecionar no NBCI, a sequência de DNA a ser analisada. Em [PEREIRA 2014] foi proposto um algoritmo para criação de um banco de dados que auxilia na busca por sequências de interesse no NBCI. Neste trabalho, analisaremos sequências de comprimentos 63 e 127 nucleotídeos.

O programa, a partir da sequência de entrada, irá rotular esses nucleotídeos em 24 permutações distintas. Em seguida, utilizando os polinômios primitivos e seus respectivos polinômios geradores, será calculado o polinômio verificação de paridade *h(x)*, para a construção da matriz verificação de paridade *H*. Com isso, calcularemos as síndromes das sequências rotuladas, armazenando as palavras-código encontradas. Caso não seja encontrada nenhuma, simularemos mutações pontuais em cada posição da sequência, respeitando a distância do código $d_{min} = 3$, pois neste caso o código BCH associado é capaz de corrigir 1 erro. Por fim, retornaremos essas palavras-código ao alfabeto genético *N* e iremos compará-las com a sequência original. Se houver alteração de algum nucleotídeo, será feita a análise comparativa dos aminoácidos, para verificar se esta sequência poderá ser utilizada como referência na análise mutacional. Esta análise mutacional será exemplificada no Capítulo 4.

Os passos do algoritmo são apresentados a seguir:

- ***Passo 1****: Inicio*;

- ***Passo 2**: Receber a sequência de DNA de entrada.*

  Inserimos a sequência de nucleotídeos obtida no NCBI com as bases A, C, G e T.

- ***Passo 3**: Determinar os parâmetros do código BCH $(n, k, 3)$ sobre a extensão de anel $GR(4, m)$ que será utilizado.*

  - n: comprimento do código que é determinado pelo número de nucleotídeos da sequência do Passo 1;
  - m: grau da extensão do anel de Galois, onde $n = 2^m - 1$;
  - k: dimensão do código, onde $k = n - m$;
  - $g(x)$: polinômio gerador de grau m obtido através de um polinômio primitivo $p(x)$ de mesmo grau como mostrado no Exemplo 2.3.15, lembrando que para cada polinômio primitivo, há um *g(x)* e portanto, um código novo.

- ***Passo 4**: Construir os 24 rotulamentos;*

  Nesta etapa, faremos a relação entre os quatro elementos de *N* com o alfabeto do código BCH sobre $\mathbb{Z}_4$. Como o mapeamento entre $N \to \mathbb{Z}_4$ não é conhecido, então devemos considerar todas as possibilidades de permutação ($4! = 24$ possibilidades). Através da associação de complementaridade dos nucleotídeos A - T/U e C - G, [ROCHA 2010, FARIA 2011] consideraram que os 24 rotulamentos obtidos possuem características geométricas que permitem classificá-los em três grupos distintos, os quais foram chamados de Rotulamentos A, B e C.

  Dessa forma, o programa irá compor uma matriz P de dimensão $24 \times n$ cujas linhas seguirão as permutações e os tipos de rotulamentos conforme apresentado na Tabela 14.

- ***Passo 5**: Para cada polinômio gerador, fazer.*

  - ***Passo 5.1**: Cálculo do polinômio, h(x);*

    O polinômio *h(x)* é obtido através da relação:

    $$h(x) = \frac{x^n - 1}{g(x)},$$

    em que *g(x)* é o polinômio gerador do código determinado no Passo 3.

  - ***Passo 5.2**: Construção da matriz verificação de paridade, H;*

    O código BCH é um código cíclico, por isso as linhas da matriz *H* serão construídas a partir do deslocamento do polinômio *h(x)*. A primeira linha da matriz será constituída pelos coeficientes do referido polinômio. Para as linhas subsequentes devemos fazer um deslocamento da direita para a esquerda de cada elemento da primeira linha. Essa matriz terá dimensão $(n - k) \times n$.

  - ***Passo 5.3**: Cálculo das síndromes **s**;*

LINHA	ROTULAMENTO (A,C,G,T)	TIPO (A,B,C)
L_1	(0,1,2,3)	B
L_2	(0,1,3,2)	A
L_3	(0,2,1,3)	C
L_4	(0,2,3,1)	C
L_5	(0,3,2,1)	B
L_6	(0,3,1,2)	A
L_7	(1,0,2,3)	A
L_8	(1,0,3,2)	B
L_9	(1,2,0,3)	A
L_{10}	(1,2,3,0)	B
L_{11}	(1,3,0,2)	C
L_{12}	(1,3,2,0)	C
L_{13}	(2,0,1,3)	C
L_{14}	(2,0,3,1)	C
L_{15}	(2,1,0,3)	B
L_{16}	(2,1,3,0)	A
L_{17}	(2,3,0,1)	B
L_{18}	(2,3,1,0)	A
L_{19}	(3,0,1,2)	B
L_{20}	(3,0,2,1)	A
L_{21}	(3,1,0,2)	C
L_{22}	(3,1,2,0)	C
L_{23}	(3,2,0,1)	A
L_{24}	(3,2,1,0)	B

Tabela 14 – 24 possibilidades de permutação.

Considerando $\mathbf{v}_i$, $i = 1, \cdots, 24$ cada linha da matriz P, a síndrome de $\mathbf{v}_i$ é obtida através da relação

$$\mathbf{s} = \mathbf{v}_i \cdot H^t,$$

em que H^t é a transposta da matriz verificação de paridade H. Para cada $\mathbf{v}_i$:

- Se $\mathbf{s} = 0$, consideramos $\mathbf{v}_i$ como uma palavra código e passamos para o Passo 5.5.
- Se $\mathbf{s} \neq 0$, passamos para o Passo 5.4.

– ***Passo 5.4****: Para cada linha de P simular as mutações pontuais para $d_{min} = 3$. Voltar ao Passo 5.3;*

Para cada posição da palavra rotulada, alteramos a base atual para uma das outras três possibilidades e, calculamos novamente sua síndrome, no Passo 5.3.

Por exemplo, se em uma posição tivermos a base A, ao alterá-la para C, a síndrome será recalculada. Caso resulte em diferente de zero novamente, a alteramos para G. Se ainda não obtivermos uma palavra código, por fim, alteramos para T. Calculando de novo ***s*** e se o resultado não for zero, voltamos à base original da posição e começamos alterar a base do próximo elemento até a palavra código ser encontrada.

Ressaltamos que há polinômios geradores e rotulamentos que não resultam em nenhuma palavra código, mesmo alterando uma posição de sua sequência. Além disso, mais de uma

posição não é modificada simultaneamente, porque estamos trabalhando com códigos BCH de distância mínima igual a 3, sendo assim, possível corrigir apenas 1 erro.

- ***Passo 5.5****: Armazenar as palavras-código e seus respectivos rotulamentos. Voltar ao Passo 5.4;*

  Neste passo, armazenamos a palavra-código obtida nos Passos 5.3 ou 5.4 em uma matriz $C_{l\times n}$, onde, l é a quantidades de palavras-código encontradas. E, seus respectivos rotulamentos em uma matriz $R_{l\times 1}$, que representa as linhas da matriz P onde foram escontradas palavras-código. Este procedimento é realizado até o Passo 5.4 simular as mutações pontuais em todas as linhas necessárias de P.

- ***Passo 6****: Comparação dos resultados;*

  Nesta etapa, retornamos ao alfabeto genético ($\mathbb{Z}_4 \rightarrow N$), respeitando o rotulamento de cada palavra código. Após isso, geramos a sequência de mRNA, onde o complemento da adenina é a uracila. Por fim, ao compararmos a sequência gerada com a de entrada, se detectarmos alguma troca de base, realizamos a comparação dos aminoácidos resultantes da leitura dos códons, de acordo com a Tabela 2 da Subseção 2.1.2.

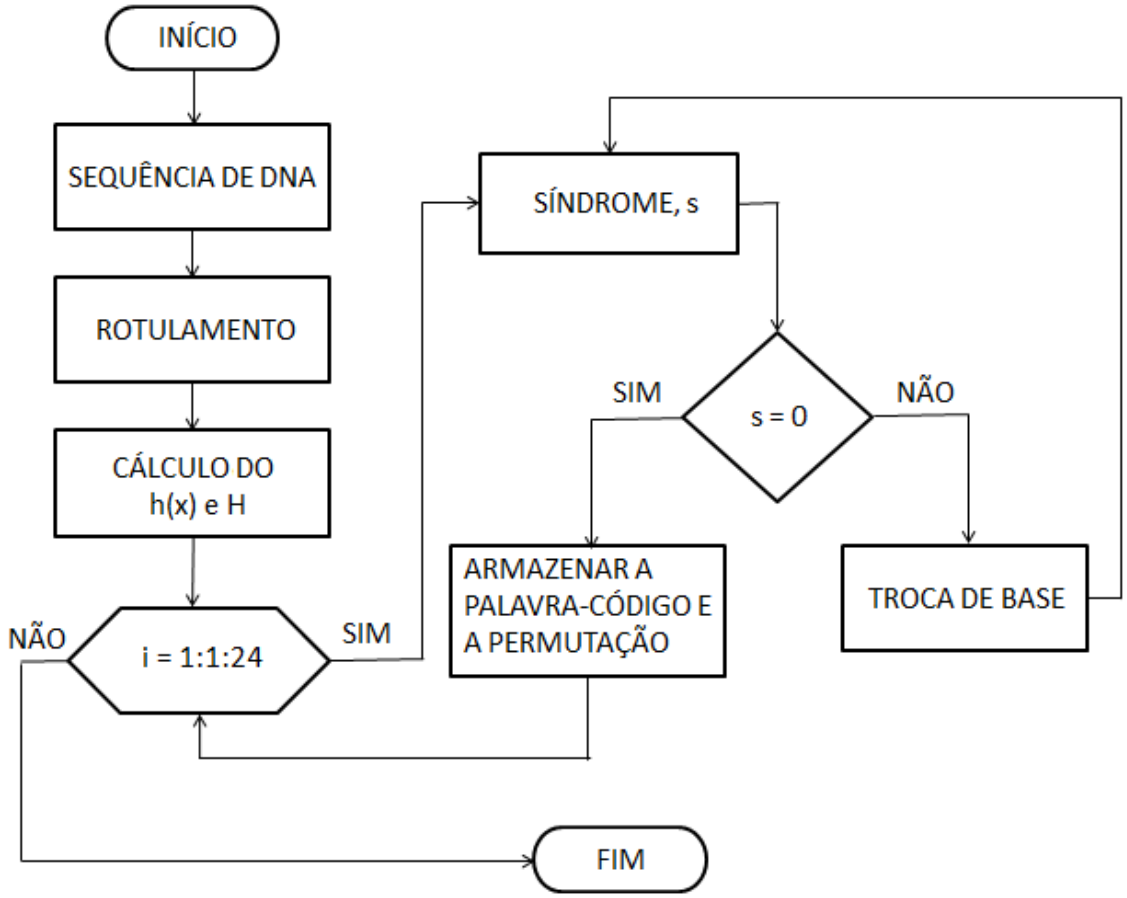
  Na comparação dos resultados consideraremos:

  - **Oaa**: sequência de aminoácidos original;
  - **OmR**: sequência de RNA mensageiro original;
  - **Ont**: sequência de nucleotídeos original;
  - **Olb**: rotulamento original;
  - **Glb**: rotulamento gerado;
  - **Gnt**: sequência de nucleotídeos gerada;
  - **GmR**: sequência de RNA mensageiro gerada;
  - **Gaa**: sequência de aminoácidos gerada.

- ***Passo 7****: Fim.*

O fluxograma referente a este algoritmo é ilustrado na Figura 11:

Figura 11 – Fluxograma.



Fonte: Produção do próprio autor.

Apresentamos a seguir, dois exemplos detalhados da execução do programa.

3.2 EXEMPLO - IDENTIFICAÇÃO DA SEQUÊNCIA DO ÉXON 23 DO GENE BRCA1.

Neste exemplo, identificaremos e reproduziremos a sequência de DNA referente ao éxon 23 do gene BRCA1, denominado: *BRCA1 exon 23, internal fragment [human, serous papillary ovarian adenocarcinoma, patient sample 61, Genomic Mutant, 68 nt]*. O BRCA1 é um gene cuja função é impedir o surgimento de tumores através da reparação de moléculas de DNA danificadas. Mutações neste gene podem levar ao desenvolvimento de doenças como o câncer de ovário, que é a segunda neoplasia ginecológica mais comum. Em 2018, foram estimados 6.150 novos casos segundo o Instituto Nacional de Câncer, [INCA 2018].

***Passo 1**: Início;*

***Passo 2**: Receber a sequência de DNA de entrada*;

- **sequência de nucleotídeos**: CCAGATCCTGGACAGAGGACAATGGCTTCCATGCAATTGGG CAGATGTGTGAGGCACCTGTGG.

***Passo 3**: Determinar os parâmetros do código BCH $(n, k, 3)$ sobre a extensão de anel $GR(4, m)$ que será utilizado. Os polinômios geradores utilizados foram obtidos no Exemplo 2.3.16*

- **número de nucleotídeos, *n***: 63;

- ***m*** = 6;
  - 63 nucleotídeos: $2^6 - 1$;
- ***k*** = 57;
  - $k = n - r = 63 - 6 = 57$;
- $g_{01}(x) = 1x^6 + 3x^5 + 2x^4 + 1x^2 + 1x^1 + 1$;
- $g_{02}(x) = 1x^6 + 1x^5 + 1x^4 + 2x^2 + 3x^1 + 1$;
- $g_{03}(x) = 1x^6 + 3x^5 + 1x^3 + 1x^2 + 2x^1 + 1$;
- $g_{04}(x) = 1x^6 + 3x^5 + 2x^3 + 1$;
- $g_{05}(x) = 1x^6 + 2x^3 + 3x^1 + 1$;
- $g_{06}(x) = 1x^6 + 2x^5 + 1x^4 + 1x^3 + 3x^1 + 1$.

A análise se dará por meio do código BCH (63, 57, 3) sobre a extensão de anel GR(4, 6).

Passo 4*: Construir os 24 rotulamentos*;

Obtemos a matriz $P_{24\times 63}$ como disposta a seguir:

$$P = \begin{bmatrix}
110203113220102022010032213311032100332221020323232022101132322\\
110302112330103033010023312211023100223331030232323033101123233\\
220103223110201011020031123322031200331112010313131011202231311\\
220301221330203033020013321122013200113332030131313033202213133\\
330201331220302022030012231133012300112223020121212022303312122\\
330102332110301011030021132233021300221113010212121011303321211\\
001213003221012122101132203300132011332220121323232122010032322\\
001312002331013133101123302200123011223330131232323133010023233\\
221013223001210100121130023322130211330002101303030100212230300\\
221310220331213133121103320022103211003332131030303133212203033\\
331012332001310100131120032233120311220003101202020100313320200\\
331210330221312122131102230033102311002223121020202122313302022\\
002123003112021211202231103300231022331110212313131211020031311\\
002321001332023233202213301100213022113330232131313233020013133\\
112023113002120200212230013311230122330001202303030200121130300\\
112320110332123233212203310011203122003331232030303233121103033\\
332021331002320200232210031133210322110003202101010200323310100\\
332120330112321211232201130033201322001113212010101211323301011\\
003132002113031311303321102200321033221110313212121311030021211\\
003231001223032322303312201100312033112220323121212322030012122\\
113032112003130300313320012211320133220001303202020300131120200\\
113230110223132322313302210011302133002221323020202322131102022\\
223031221003230300323310021122310233110002303101010300232210100\\
223130220113231311323301120022301233001112313010101311232201011
\end{bmatrix}.$$

Passo 5*: Para cada polinômio gerador, fazer.*

Passo 5.1*: Cálculo do polinômio, h(x)*;

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

$h(x) = 1x^{57} + 1x^{56} + 3x^{55} + 1x^{54} + 2x^{53} + 2x^{52} + 1x^{51} + 3x^{48} + 2x^{47} + 1x^{46} + 3x^{44} + 2x^{43} + 2x^{42} + 3x^{41} + 3x^{40} + 3x^{38} + 2x^{36} + 2x^{34} + 3x^{33} + 2x^{32} + 2x^{31} + 2x^{30} + 1x^{29} + 2x^{28} + 1x^{27} + 3x^{26} + 1x^{24} + 1x^{23} + 1x^{22} + 3x^{21} + 1x^{20} + 1x^{19} + 1x^{17} + 1x^{15} + 3x^{14} + 3x^{13} + 1x^{9} + 3x^{8} + 2x^{7} + 3x^{5} + 3x^{4} + 3x^{3} + 1x^{1} + 3.$

Passo 5.2*: Construção da matriz verificação de paridade, H*;

A matriz verificação de paridade H, associada ao polinômio $h(x)$ resultante do passo anterior, possui dimensão 6x63.

$$H_{01} = \begin{bmatrix} 000001131221003210322330302023222121301113110101330001320333013 \\ 00001131221003210322330302023222121301113110101330001320333013 0 \\ 0001131221003210322330302023222121301113110101330001320333013 00 \\ 001131221003210322330302023222121301113110101330001320333013000 \\ 011312210032103223303020232221213011311010133000132033301300 00 \\ 1131221003210322330302023222121301113110101330001320333013000 00 \end{bmatrix}.$$

Passo 5.3*: Cálculo das síndromes, s*;

A matriz S apresenta as síndromes calculadas a partir da multiplicação de cada uma das 24 palavras, apresentadas no Passo 4, pela matriz transposta verificação de paridade H^t de H construída no Passo 5.2:

$$S = \begin{bmatrix} 110101 \\ 120002 \\ 212303 \\ 232101 \\ 330303 \\ 320002 \\ 320002 \\ 330303 \\ 120002 \\ 110101 \\ 232101 \\ 212303 \\ 232101 \\ 212303 \\ 330303 \\ 320002 \\ 110101 \\ 120002 \\ 110101 \\ 120002 \\ 212303 \\ 232101 \\ 320002 \\ 330303 \end{bmatrix}.$$

Passo 5.4*: Para cada linha de P simular as mutações pontuais para $d_{min} = 3$. Voltar ao passo 5.3*;

A partir da matriz síndrome calculada no passo anterior, verificamos que não há nenhuma palavra código dentre as 24 palavras rotuladas, visto que nenhuma síndrome resultou no vetor nulo. Por esse motivo, modificaremos sucessivamente a base de cada posição das linhas de P, substituindo-a pelas outras três possibilidades existentes no alfabeto N e sempre retornando ao Passo 5.3 para recalcular a síndrome. Inicialmente, a primeira linha de P com rotulamento $L_1 = \{A = 0, C = 1, G = 2, T = 3\}$ apresentou a seguinte sequência rotulada:

$$1102031132201020220100322133110321003322210203232320221011323 22;$$

Verificamos que na posição 43, considerando a leitura da esquerda para direita, a base é uma adenina (A). Ao trocarmos essa base pela timina (T) resultamos na seguinte sequência:

$$1102031132201020220100322133110321003322213203232320221011323 22;$$

Ao recalcularmos a síndrome, encontramos:

$$\mathbf{s} = [000000] : \text{portanto, é palavra-código;}$$

Contudo, se não tivessemos encontrado uma palavra-código, retornaríamos para a base adenina e continuaríamos o processo para as outras posições até que seja encontrada a palavra-código. Neste ponto, cessamos as alterações de base da linha selecionada e passamos ao Passo 5.5.

Passo 5.5*: Armazenar as palavras-código e seus respectivos rotulamentos. Voltar ao Passo 5.4.*;

Depois de realizadas as trocas de bases nas 63 posições da primeira linha (matriz P) e encontrada uma palavra código, esta é armazenada na matriz C com seu respectivo rotulamento na matriz R. Então, voltamos ao Passo 5.4 para simular as mutações para a próxima linha de P. Ao final de todas as simulações para $g_{01}(x)$ encontramos 8 palavras-código relacionadas ao Rotulamento B, apresentadas a seguir:

$$C_{8\times 63} = \begin{bmatrix} 110203113220102022010032213311032100332221320323232022101132322 \\ 330201331220302022030012231133012300112223120121212022303312122 \\ 001312002331013133101123302200123011223330231232323133010023233 \\ 221310220331213133121103320022103211003332031030303133212203033 \\ 112023113002120200212230013311230122330001302303030200121130300 \\ 332021331002320200232210031133210322110003102101010200323310100 \\ 003132002113031311303321102200321033221110213212121311030021211 \\ 223130220113231311323301120022301233001112013010101311232201011 \end{bmatrix}.$$

Os rotulamentos referentes a cada linha da matriz C são:

$$R_{8\times1} = \begin{bmatrix} 1 \\ 5 \\ 8 \\ 10 \\ 15 \\ 17 \\ 19 \\ 24 \end{bmatrix}.$$

***Passo 6**: Comparação dos resultados*;

Nesta etapa, comparando as palavras-código apresentadas no passo anterior com a sequência original, verificamos que estas se diferenciam em uma posição, conforme:

Linha 1 – Rotulamento B: {A = 0, C = 1, G = 2, T = 3}.

Olb: 110 203 113 220 102 022 010 032 213 311 032 100 332 221 **0**20 323 232 022 101 132 322

Glb: 110 203 113 220 102 022 010 032 213 311 032 100 332 221 **3**20 323 232 022 101 132 322

Portanto, retornamos ao alfabeto N ($\mathbb{Z}_4 \to N$) e comparamos as sequências de nucleotídeos para identificar qual base, códon e aminoácido foram modificados. Mostraremos a comparação para a primeira linha de C, apresentando os resultados na Tabela 15.

Biologicamente:

Ont: $CCA\ GAT\ CCT\ GGA\ CAG\ AGG\ ACA\ ATG\ GCT\ TCC\ ATG\ CAA\ TTG\ GGC\ \mathbf{A}GA\ TGT\ GTG\ AGG\ CAC\ CTG\ TGG$

OmR: $GGU\ CUA\ GGA\ CCU\ GUC\ UCC\ UGU\ UAC\ CGA\ AGG\ UAC\ GUU\ AAC\ CCG\ \mathbf{U}CU\ ACA\ CAC\ UCC\ GUG\ GAC\ ACC$

Oaa: $P\ D\ P\ G\ Q\ R\ T\ M\ A\ S\ M\ Q\ L\ G\ \mathbf{R}\ C\ V\ R\ H\ L\ W$

Gnt: $CCA\ GAT\ CCT\ GGA\ CAG\ AGG\ ACA\ ATG\ GCT\ TCC\ ATG\ CAA\ TTG\ GGC\ \mathbf{T}GA\ TGT\ GTG\ AGG\ CAC\ CTG\ TGG$

GmR: $GGU\ CUA\ GGA\ CCU\ GUC\ UCC\ UGU\ UAC\ CGA\ AGG\ UAC\ GUU\ AAC\ CCG\ \mathbf{A}CU\ ACA\ CAC\ UCC\ GUG\ GAC\ ACC$

Gaa: $P\ D\ P\ G\ Q\ R\ T\ M\ A\ S\ M\ Q\ L\ G\ \mathbf{STOP}\ C\ V\ R\ H\ L\ W$

Linha	Troca de base	Troca de códon	Troca de aminoácido
1	$A \to T$	$AGA \to TGA$	$Arginina \to$ ***STOP***

Tabela 15 – Resultado das comparações.

A sequência gerada pelo código apresentou a troca de adenina para timina ocasionando a mudança do códon que traduz o aminoácido Arginina para um códon de parada (*stop*). Verificamos na Tabela 16 que para todas as palavras-código geradas com diferentes rotulamentos, a troca de base da adenina

para a timina ocorreu na mesma posição (destacada pela cor vermelha). Essa alteração é conhecida como mutação pontual *nonsense*.

Linha	Palavras-código geradas
1	1102031132201020220100322133110321003322213203232322022101132322
5	3302013312203020220300122311330123001122231201212120223033121 22
8	0013120023310131331011233022001230112233302312323233010023233
10	2213102203312131331211033200221032110033320310303033212203033
15	1120231130021202002122300133112301223300013023030302001211303 00
17	3320213310023202002322100311332103221100031021010102003233101 00
19	0031320021130313113033211022003210332211102132121213110300212 11
24	2231302201132313113233011200223012330011120130101013112322010 11

Tabela 16 – Comparação de todas as linhas de C.

A partir dos resultados das comparações, observamos que apesar do código BCH conseguir identificar e reproduzir essa sequência de DNA (éxon 23 do gene BRCA1) com uma diferença de nucleotídeo, o códon de parada poderá modificar o comprimento da palavra, não sendo possível utilizá-la como referência em análises mutacionais. Isso ocorre, porque uma das limitações do código BCH é de que o comprimento da sequência de entrada deve se manter constante durante toda a sua execução.

Observamos também que dentre os seis polinômios geradores apresentados no Passo 2, o único que gerou palavras-código foi o o $g_{01}(x) = 1x^6 + 3x^5 + 2x^4 + 1x^2 + 1x^1 + 1$. Por esse motivo, escolhemos este polinômio para exemplificar o algoritmo.

Passo 7: fim.

3.3 EXEMPLO - IDENTIFICAÇÃO DA SEQUÊNCIA DE DNA ASSOCIADA A UM RNA MENSAGEIRO PARA A CADEIA BETA DO RECEPTOR DA CÉLULA T EM CAMUNDONGO.

As células T ou linfócitos T são um grupo de glóbulos brancos reponsáveis pela defesa do organismo contra os antígenos. Para o reconhecimento desse agente agressor, as células T apresentam receptores na sua membrana, [ABBAS A. K.; LICHTMAN e PILLAI 2012]. Neste exemplo, identificaremos a sequência de DNA associada a um mRNA para a cadeia beta V(beta)14-J(beta)2.2 do receptor da célula T em camundongo, denominada *Mouse mRNA for T-cell receptor beta-chain V(beta)14-J(beta)2.2*. Esta sequência também possui comprimento 63 e para seu código BCH serão utilizados os mesmos seis polinômios geradores do exemplo anterior.

***Passo 1**: Início;*
***Passo 2**: Receber a sequência de DNA de entrada*;

- **sequência de nucleotídeos**: TGTGCCTGGAGTCTAGCGGGGGAGCAGCTCTACTTTGGTGAAGGCTCAAAGCTGACAGTGCTG.

***Passo 3**: Determinar os parâmetros do código BCH $(n, k, 3)$ sobre a extensão de anel $GR(4, m)$ que será utilizado. Os polinômios geradores utilizados foram obtidos no Exemplo 2.3.16*

- **número de nucleotídeos, *n***: 63;
- ***m*** = 6;
  - 63 nucleotídeos: $2^6 - 1$;
- ***k*** = 57;
  - $k = n - r = 63 - 6 = 57$;
- $g_{01}(x) = 1x^6 + 3x^5 + 2x^4 + 1x^2 + 1x^1 + 1$;
- $g_{02}(x) = 1x^6 + 1x^5 + 1x^4 + 2x^2 + 3x^1 + 1$;
- $g_{03}(x) = 1x^6 + 3x^5 + 1x^3 + 1x^2 + 2x^1 + 1$;
- $g_{04}(x) = 1x^6 + 3x^5 + 2x^3 + 1$;
- $g_{05}(x) = 1x^6 + 2x^3 + 3x^1 + 1$;
- $g_{06}(x) = 1x^6 + 2x^5 + 1x^4 + 1x^3 + 3x^1 + 1$.

A análise se dará por meio do código BCH (63, 57, 3) sobre a extensão de anel GR(4, 6).
***Passo 4**: Construir os 24 rotulamentos*;
Obtemos a matriz P como disposta a seguir:

$$P = \begin{bmatrix}
323211322023130212222202102131301333223200221310002132010232132\\
232311233032120313333303103121201222332300331210003123010323123\\
313122311013230121111101201232302333113100112320001231020131231\\
131322133031210323333303203212102111331300332120003213020313213\\
121233122021310232222202302313103111221200223130002312030212312\\
212133211012320131111101301323203222112100113230001321030121321\\
323200322123031202222212012030310333223211220301112032101232032\\
232300233132021303333313013020210222332311330201113023101323023\\
303022300103231020000010210232312333003011002321110230121030230\\
030322033130201323333313213202012000330311332021113203121303203\\
202033200102321030000010310323213222002011003231110320131020320\\
020233022120301232222212312303013000220211223031112302131202302\\
313100311213032101111121021030320333113122110302221031202131031\\
131300133231012303333323023010120111331322330102223013202313013\\
303011300203132010000020120131321333003022001312220130212030130\\
030311033230102313333323123101021000330322331012223103212303103\\
101033100201312030000020320313123111001022003132220310232010310\\
010133011210302131111121321303023000110122113032221301232101301\\
212100211312023101111131031020230222112133110203331021303121021\\
121200122321013202222232032010130111221233220103332012303212012\\
202011200302123010000030130121231222002033001213330120313020120\\
020211022320103212222232132101031000220233221013332102313202102\\
101022100301213020000030230212132111001033002123330210323010210\\
010122011310203121111131231202032000110133112023331201323101201
\end{bmatrix}.$$

***Passo 5**: Para cada polinômio gerador, fazer.*

Dentre os seis polinômios geradores, apenas dois geraram palavras-código, o $g_{01}(x) = 1x^6 + 3x^5 + 2x^4 + 1x^2 + 1x^1 + 1$ e o $g_{02}(x) = 1x^6 + 1x^5 + 1x^4 + 2x^2 + 3x^1 + 1$.

Importante lembrar que, para o polinômio gerador $g_{01}(x)$ já obtivemos $h_{01}(x)$ e H_{01} no exemplo anterior. Mostraremos os Passos 5.1 e 5.2 apenas para $g_{02}(x)$.

***Passo 5.1**: Cálculo do polinômio, h(x)*;

$$h_{02}(x) = \frac{x^{63} - 1}{1x^6 + 1x^5 + 1x^4 + 2x^2 + 3x^1 + 1};$$

$h_{02}(x) = 1x^{57} + 3x^{56} + 1x^{54} + 1x^{53} + 1x^{52} + 2x^{50} + 1x^{49} + 3x^{48} + 1x^{44} + 1x^{43} + 3x^{42} + 3x^{40} + 3x^{38} + 3x^{37} + 1x^{36} + 3x^{35} + 3x^{34} + 3x^{33} + 1x^{31} + 3x^{30} + 2x^{29} + 3x^{28} + 2x^{27} + 2x^{26} + 2x^{25} + 1x^{24} + 2x^{23} + 2x^{21} + 1x^{19} + 1x^{17} + 1x^{16} + 2x^{15} + 2x^{14} + 1x^{13} + 3x^{11} + 2x^{10} + 1x^9 + 3x^6 + 2x^5 + 2x^4 + 3x^3 + 1x^2 + 3x^1 + 3.$

***Passo 5.2**: Construção da matriz verificação de paridade, H*;

A matriz verificação de paridade H possui dimensão 6x63.

$$H_{02} = \begin{bmatrix} 000001301110213000113030331333013232221202010112210321003223133 \\ 000013011102130001130303313330132322212020101122103210032231330 \\ 000130111021300011303033133301323222120201011221032100322313300 \\ 001301110213000113030331333013232221202010112210321003223133000 \\ 013011102130001130303313330132322212020101122103210032231330000 \\ 130111021300011303033133301323222120201011221032100322313300000 \end{bmatrix}.$$

***Passo 5.3**: Cálculo das síndromes, s*;

Aa matrizes S_{01} e S_{02} apresentam as síndromes calculadas a partir da multiplicação de cada uma das 24 palavras, apresentadas no Passo 4, pelas matrizes transpostas verificação de paridade H^t de H_{01} e H_{02}:

$$S_{01} = \begin{bmatrix} 013203 \\ 020203 \\ 013202 \\ 031202 \\ 031201 \\ 020201 \\ \hline 020201 \\ 031201 \\ 020203 \\ 013203 \\ 031202 \\ 013202 \\ 031202 \\ \hline 013202 \\ 031201 \\ 020201 \\ 013203 \\ 020203 \\ 013203 \\ 020203 \\ \hline 013202 \\ 031202 \\ 020201 \\ 031201 \end{bmatrix}.$$

$$S_{02} = \begin{bmatrix} 122103 \\ 220002 \\ 102101 \\ 302303 \\ 322301 \\ 220002 \\ 220002 \\ 322301 \\ 220002 \\ \hline 122103 \\ 302303 \\ 102101 \\ 302303 \\ 102101 \\ 322301 \\ 220002 \\ \hline 122103 \\ 220002 \\ 122103 \\ 220002 \\ 102101 \\ 302303 \\ 220002 \\ 322301 \end{bmatrix}.$$

Passo 5.4*: Para cada linha de P simular as mutações pontuais para $d_{min} = 3$. Voltar ao passo 5.3*;

Para cada matriz síndrome calculada no passo anterior, verificamos que não há nenhuma palavra código dentre as 24 palavras rotuladas, visto que nenhuma síndrome resultou no vetor nulo. Por esse motivo, modificaremos sucessivamente a base de cada posição das linhas de P, substituindo-a pelas outras três possibilidades existentes no alfabeto N e sempre retornando ao Passo 5.3 para recalcular a síndrome, associando-a a sua respectiva matriz verificação de paridade H.

Passo 5.5*: Armazenar as palavras-código e seus respectivos rotulamentos. Voltar ao Passo 5.4.*;

Depois de realizadas as trocas de bases nas 63 posições da primeira linha (matriz P) e encontrada uma palavra código, esta é armazenada na matriz C com seu respectivo rotulamento na matriz R. Então, voltamos ao Passo 5.4 para simular as mutações para a próxima linha de P. Ao final de todas as simulações obtemos para $g_{01}(x)$ 8 palavras-código relacionadas ao Rotulamento B e, para $g_{02}(x)$, 16 palavras relacionadas aos Rotulamentos A e B, apresentadas a seguir.

$$C_{01} = \begin{bmatrix}
323211322023130212222202102131301333223200221310002122010232132 \\
121233122021310232222023023131031112212002231300023220302123 12 \\
232300233132021303333313013020210222332311330201113033101323023 \\
030322033130201323333313213202012000330311332021113233121303203 \\
303011300203132010000020120131321333003022001312220100212030130 \\
101033100201312030000020320313123111001022003132220300232010310 \\
212100211312023101111131031020230222112133110203331011303121021 \\
010122011310203121111131231202032000110133112023331211323101201
\end{bmatrix};$$

$$C_{02} = \begin{bmatrix}
323211322023130212222202102131301333223200221300002132010232132 \\
232311233032100313333303103121201222332300331210003123010323123 \\
121233122021310232222023023131031112212002231000023120302123 12 \\
212133211012300131111101301323203222112100113230001321030121321 \\
323200322123011202222120120303103333223211220301112032101232032 \\
232300233132021303333313013020210222332311330211113023101323023 \\
303022300103211020000010210232312333003011002321110230121030230 \\
030322033130201323333313213202012000330311332011113203121303203 \\
303011300203132010000020120131321333003022001322220130212030130 \\
030311033230122313333323123101021000330322331012223103212303103 \\
101033100201312030000020320313123111001022003122220310232010310 \\
010133011210322131111121321303023000110122113032221301232101301 \\
212100211312023101111131031020230222112133110233331021303121021 \\
121200122321033202222232032010130111221233220103332012303212012 \\
101022100301233020000030230212132111001033002123330210323010210 \\
010122011310203121111131231202032000110133112033331201323101201
\end{bmatrix}.$$

Os rotulamentos referentes a cada linha das matrizes C_{01} e C_{02} são, respectivamente:

$$R_{01} = \begin{bmatrix} 1 \\ 5 \\ 8 \\ 10 \\ 15 \\ 17 \\ 19 \\ 24 \end{bmatrix} \quad ; \quad R_{02} = \begin{bmatrix} 1 \\ 2 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 15 \\ 16 \\ 17 \\ 18 \\ 19 \\ 20 \\ 23 \\ 24 \end{bmatrix}.$$

***Passo 6**: Comparação dos resultados*;

Nesta etapa, comparando as palavras-código apresentadas no passo anterior com a sequência original, verificamos que estas se diferenciam em uma posição, conforme demonstramos a seguir:

- $g_{01}(x)$:

  Linha 1 – Rotulamento B: {A = 0, C = 1, G = 2, T = 3}.

  Olb: 323 211 322 023 130 212 222 202 102 131 301 333 223 200 221 310 002 1**3**2 010 232 132

  Glb: 323 211 322 023 130 212 222 202 102 131 301 333 223 200 221 310 002 1**2**2 010 232 132

- $g_{02}(x)$:

  Linha 1 – Rotulamento B: {A = 0, C = 1, G = 2, T = 3}.

  Olb: 323 211 322 023 130 212 222 202 102 131 301 333 223 200 221 3**1**0 002 132 010 232 132

  Glb: 323 211 322 023 130 212 222 202 102 131 301 333 223 200 221 3**0**0 002 132 010 232 132

  Linha 2 – Rotulamento A: {A = 0, C = 1, G = 3, T = 2}.

  Olb: 232 311 233 032 1**2**0 313 333 303 103 121 201 222 332 300 331 210 003 123 010 323 123

  Glb: 232 311 233 032 1**0**0 313 333 303 103 121 201 222 332 300 331 210 003 123 010 323 123

Portanto, retornamos ao alfabeto N ($\mathbb{Z}_4 \to N$) e comparamos as sequências de nucleotídeos para identificar quais bases, códons e aminoácidos foram modificados. Mostraremos as comparações para a primeira linha de C_{01} e para a primeira e segunda linha de C_{02}, apresentando os resultados na Tabela 17.

Biologicamente:

- $g_{01}(x)$:

  Ont: $TGT\ GCC\ TGG\ AGT\ CTA\ GCG\ GGG\ GAG\ CAG\ CTC\ TAC\ TTT\ GGT\ GAA\ GGC\ TCA\ AAG\ C\mathbf{T}G\ ACA\ GTG\ CTG$

  OmR: $ACA\ CGG\ ACC\ UCA\ GAU\ CGC\ CCC\ CUC\ GUC\ GAG\ AUG\ AAA\ CCA\ CUU\ CCG\ AGU\ UUC\ G\mathbf{A}C\ UGU\ CAC\ GAC$

  Oaa: $C\ \ A\ \ W\ \ S\ \ L\ \ A\ \ G\ \ E\ \ Q\ \ L\ \ Y\ \ F\ \ G\ \ E\ \ G\ \ S\ \ K\ \ \mathbf{L}\ \ T\ \ V\ \ L$

  Gnt: $TGT\ GCC\ TGG\ AGT\ CTA\ GCG\ GGG\ GAG\ CAG\ CTC\ TAC\ TTT\ GGT\ GAA\ GGC\ TCA\ AAG\ C\mathbf{G}G\ ACA\ GTG\ CTG$

  GmR: $ACA\ CGG\ ACC\ UCA\ GAU\ CGC\ CCC\ CUC\ GUC\ GAG\ AUG\ AAA\ CCA\ CUU\ CCG\ AGU\ UUC\ G\mathbf{C}C\ UGU\ CAC\ GAC$

  Gaa: $C\ \ A\ \ W\ \ S\ \ L\ \ A\ \ G\ \ E\ \ Q\ \ L\ \ Y\ \ F\ \ G\ \ E\ \ G\ \ S\ \ K\ \ \mathbf{R}\ \ T\ \ V\ \ L$

- $g_{02}(x)$:

  Linha 1 – Rotulamento B:

  Ont: $TGT\ GCC\ TGG\ AGT\ CTA\ GCG\ GGG\ GAG\ CAG\ CTC\ TAC\ TTT\ GGT\ GAA\ GGC\ T\mathbf{C}A\ AAG\ CTG\ ACA\ GTG\ CTG$

  OmR: $ACA\ CGG\ ACC\ UCA\ GAU\ CGC\ CCC\ CUC\ GUC\ GAG\ AUG\ AAA\ CCA\ CUU\ CCG\ A\mathbf{G}U\ UUC\ GAC\ UGU\ CAC\ GAC$

  Oaa: $C\ \ A\ \ W\ \ S\ \ L\ \ A\ \ G\ \ E\ \ Q\ \ L\ \ Y\ \ F\ \ G\ \ E\ \ G\ \ \mathbf{S}\ \ K\ \ L\ \ T\ \ V\ \ L$

  Gnt: $TGT\ GCC\ TGG\ AGT\ CTA\ GCG\ GGG\ GAG\ CAG\ CTC\ TAC\ TTT\ GGT\ GAA\ GGC\ T\mathbf{A}A\ AAG\ CTG\ ACA\ GTG\ CTG$

  GmR: $ACA\ CGG\ ACC\ UCA\ GAU\ CGC\ CCC\ CUC\ GUC\ GAG\ AUG\ AAA\ CCA\ CUU\ CCG\ A\mathbf{U}U\ UUC\ GAC\ UGU\ CAC\ GAC$

  Gaa: $C\ \ A\ \ W\ \ S\ \ L\ \ A\ \ G\ \ E\ \ Q\ \ L\ \ Y\ \ F\ \ G\ \ E\ \ G\ \ \mathbf{STOP}\ \ K\ \ L\ \ T\ \ V\ \ L$

  Linha 2 – Rotulamento A:

  Ont: $TGT\ GCC\ TGG\ AGT\ C\mathbf{T}A\ GCG\ GGG\ GAG\ CAG\ CTC\ TAC\ TTT\ GGT\ GAA\ GGC\ TCA\ AAG\ CTG\ ACA\ GTG\ CTG$

  OmR: $ACA\ CGG\ ACC\ UCA\ G\mathbf{A}U\ CGC\ CCC\ CUC\ GUC\ GAG\ AUG\ AAA\ CCA\ CUU\ CCG\ A\mathbf{G}U\ UUC\ GAC\ UGU\ CAC\ GAC$

  Oaa: $C\ \ A\ \ W\ \ S\ \ \mathbf{L}\ \ A\ \ G\ \ E\ \ Q\ \ L\ \ Y\ \ F\ \ G\ \ E\ \ G\ \ S\ \ K\ \ L\ \ T\ \ V\ \ L$

Gnt: $TGT\ GCC\ TGG\ AGT\ CAA\ GCG\ GGG\ GAG\ CAG\ CTC\ TAC\ TTT\ GGT\ GAA\ GGC\ TAA\ AAG\ CTG\ ACA\ GTG\ CTG$

GmR: $ACA\ CGG\ ACC\ UCA\ GUU\ CGC\ CCC\ CUC\ GUC\ GAG\ AUG\ AAA\ CCA\ CUU\ CCG\ AUU\ UUC\ GAC\ UGU\ CAC\ GAC$

Gaa: $C\ A\ W\ S\ Q\ A\ G\ E\ Q\ L\ Y\ F\ G\ E\ G\ S\ K\ L\ T\ V\ L$

Polinômio gerador	Linha	Troca de base	Troca de códon	Troca de aminoácido
$g_{01}(x)$	1	$T \rightarrow G$	$CTG \rightarrow CGG$	$Leucina \rightarrow Arginina$
$g_{02}(x)$	1	$C \rightarrow A$	$TCA \rightarrow TAA$	$Serina \rightarrow$ ***STOP***
	2	$T \rightarrow A$	$CTA \rightarrow CAA$	$Leucina \rightarrow Glutamina$

Tabela 17 – Resultado das comparações.

A partir das comparações apresentadas nas Tabelas 17 e 18 observamos que, assim como, no exemplo da Seção 3.2 todas as trocas de base ocorreram na mesma posição, resultando na mesma base, códon e aminoácido. A modificação da Leucina para a Arginina alterou a característica do aminoácido de não polar para básico. Biologicamente, não sabemos a consequência dessa mudança, porém, consideramos a priori que as palavras-código com essa modificação são válidas para serem utilizadas em uma análise mutacional.

Linha	**Palavras-código geradas**
1	3232113220231302122220210213130133223200221310002122010232132
5	1212331220213102322222023023131031112212002231300023220302123 12
8	2323002331320213033333130130202102223323113302011130331013230 23
10	0303220331302013233333132132020120003303113320211132331213032 03
15	3030113002031320100000201201313213330030220013122201002120301 30
17	1010331002013120300000203203131231110010220031322203002320103 10
19	2121002113120231011111310310202302221121331102033310113031210 21
24	0101220113102031211111312312020320001101331120233312113231012 01

Tabela 18 – Comparação de todas as linhas de C_{01}.

Já, nas comparações apresentadas nas Tabelas 17 e 19, observamos que para diferentes rotulamentos, a troca de base ocorreu em diferentes posições alterando para a mesma base adenina, mas ocasionando diferentes códons e aminoácidos. Analisando os aminoácidos resultantes, concluímos que para o Rotulamento B com a geração do códon de parada, as palavras-código não poderão ser utilizadas em análises mutacionais. Entretanto, para o Rotulamento A, a troca da Leucina para Glutamina, alterou a característica do aminoácido de não polar para polar. Da mesma forma, como explicado anteriormente, consideramos também que as palavras-código com essa modificação são válidas para serem utilizadas em uma análise mutacional.

Passo 7: fim.

A partir dos dois exemplos anteriormente apresentados, observamos que:

Linha	Palavras-código geradas
1	3232113220231302122222021021313013332232002213**0**0002132010232132
2	2323112330321**0**03133333031031212012223323003312100031230103231 23
5	1212331220213102322222023023131031112212002231**0**0002312030212312
6	2121332110123**0**013111110130132320322211210011323000132103012 1321
7	3232003221230**1**120222221201203031033322321122030111203210123 2032
8	2323002331320213033333130130202102223323113302**1**1113023101323023
9	3030223001032**1**10200000102102323123330030110023211102301210302 30
10	0303220331302013233333132132020120003303113320**1**1113203121303203
15	3030113002031320100000201201313213330030220013**2**2220130212030130
16	0303110332301**2**2313333323123101021000330322331012223103212303103
17	1010331002013120300000203203131231110010220031**2**2220310232010310
18	0101330112103**2**213111112132130302300011012211303222130123210 1301
19	2121002113120231011111310310202302221121331102**3**3331021303121021
20	1212001223210**3**320222223203201013011221233220103332012303212012
23	1010221003012**3**302000003023021213211100103300212333021032301021 0
24	0101220113102031211111312312020320001101331120**3**3331201323101201

Tabela 19 – Comparação de todas as linhas de C_{02}.

1. O polinõmio $g_{01}(x)$ identificou palavras-código com o mesmo Rotulamento B, para ambas as sequências. Contudo, a troca de base ocorreu em posições distintas, ocasionando diferentes bases, códons e aminoácidos;

2. Dentre os seis polinômios geradores utilizados, não encontramos um polinômio específico que obtivesse as mesmas trocas de base para diferentes sequências. Isto é, não identificamos um padrão para as trocas.

No próximo capítulo, realizaremos a análise mutacional da sequência do éxon 14 do gene BRCA1 que possui comprimento 127, com o objetivo de apresentar outra aplicação para esse algoritmo.

4 ANÁLISE MUTACIONAL DO ÉXON 14 DO GENE BRCA1

Neste capítulo abordaremos a análise da sequência de DNA do éxon 14 do gene BRCA1, com o objetivo de demonstrar como o código corretor de erro identifica mutações. As análises mutacionais serão divididas em duas etapas e realizadas considerando duas mutações *nonsense* e duas *missense*, descritas na Tabela 20, dentre as que foram apresentadas no Exemplo 2.1.2 da Seção 2.1.

Mutação	Posição	Códon	Troca de base	Troca de aminoácido	Designação	Importância clínica
Missense	4521	1468	A para C	Asn para His	N1468H	desconhecido
Missense	4569	1484	T para A	Ser para Thr	S1484T	desconhecido
Nonsense	4489	1457	C para G	Ser para Stop	S1457X	sim
Nonsense	4508	1463	C para A	Tyr para Stop	Y1463X	sim

Tabela 20 – Mutações.

A seguir descreveremos cada etapa e seus respectivos resultados.

- **Primeira etapa**: encontrar as palavras-código referentes à sequência de nucleotídeos do éxon 14 por meio do código BCH(127,120,3). Ressaltamos que para a palavra-código ser válida para ser utilizada como referência na próxima etapa, não poderá codificar um códon de parada, pois reduzirá o comprimento da sequência;

  - Sequência: CAGTATTAACTTCACAGAAAAGTAGTGAATACCCTATAAGCCAGAATCC AGAAGGCCTTTCTGCTGACAAGTTTGAGGTGTCTGCAGATAGTTCTACCAGTAAA AATAAAGAACCAGGAGTGGAAAG;
  - Número de nucleotídeos: 127;
  - m: 7;
  - $p(x) = x^7 + x^3 + 1$;
    - $g(x) = x^7 + 2x^5 + x^3 + 3$;
    - $h(x) = 1x^{120} + 2x^{118} + 3x^{116} + 1x^{113} + 1x^{112} + 2x^{110} + 2x^{109} + 3x^{108} + 1x^{106} + 3x^{105} + 3x^{104} + 3x^{102} + 3x^{100} + 1x^{99} + 2x^{98} + 1x^{97} + 1x^{96} + 2x^{93} + 2x^{91} + 1x^{90} + 3x^{89} + 2x^{88} + 1x^{86} + 1x^{85} + 2x^{84} + 3x^{83} + 2x^{82} + 3x^{81} + 2x^{80} + 3x^{78} + 3x^{77} + 3x^{76} + 2x^{74} + 3x^{73} + 1x^{72} + 1x^{71} + 3x^{70} + 2x^{69} + 1x^{68} + 1x^{67} + 2x^{66} + 1x^{65} + 1x^{60} + 2x^{59} + 3x^{58} + 1x^{56} + 2x^{55} + 3x^{54} + 1x^{53} + 3x^{52} + 3x^{51} + 3x^{50} + 2x^{49} + 1x^{48} + 1x^{45} + 2x^{44} + 1x^{43} + 2x^{42} + 2x^{41} + 2x^{40} + 3x^{39} + 3x^{38} + 2x^{37} + 1x^{36} + 3x^{35} + 1x^{34} + 2x^{33} + 1x^{30} + 3x^{29} + 1x^{28} + 3x^{27} + 3x^{26} + 3x^{25} + 1x^{24} + 2x^{22} + 2x^{21} + 2x^{20} + 3x^{19} + 1x^{18} + 1x^{17} + 1x^{15} + 1x^{14} + 3x^{13} + 1x^{12} + 2x^{11} + 2x^{10} + 1x^{9} + 1x^{7}$.

A partir desses parâmetros, fazendo o rotulamento da sequência acima através das 24 possibilidades apresentadas na Tabela 14, geramos uma matriz P de ordem 24×127. Após calcularmos as síndromes e simular as mutações pontuais, foram encontradas 4 palavras-código pertencentes ao Rotulamento A

$$C = \begin{pmatrix} 10320220012210103000032032300201112020031103002110300331122212312301003222\ldots \\ \ldots 30332321231030203221201103200000200030011133032330003 \\ 30120220032230301000012012100203332020013301002330100113322232132103001222\ldots \\ \ldots 10112123213010201223203301200000200010033311012110001 \\ 12302002210012123222230230322021110202231123220112322331100010310321223000\ldots \\ \ldots 32330001031232023001021123022222022232211233230332223 \\ 32102002230032321222210210122023330202213321220332122113300030130123221000\ldots \\ \ldots 12110003013212021003023321022222022212233211210112221 \end{pmatrix};$$

As linhas referentes aos rotulamentos encontrados são

$$R = \begin{bmatrix} 2 \\ 6 \\ 16 \\ 18 \end{bmatrix}.$$

Segue as comparações das palavras-código com a sequência de entrada. Em negrito está destacada a posição onde ocorreu a troca de base.

Linha 2: Rotulamento A: {A = 0, C = 1, G = 3, T = 2}

Olb : 103 202 200 122 101 030 000 320 323 002 011 120 200 311 030 021 103 003 311 222 123 123 010 032 223 033 232 123 103 020 322 120 110 320 000 020 003 001 1**0**3 303 233 000 3

Glb : 103 202 200 122 101 030 000 320 323 002 011 120 200 311 030 021 103 003 311 222 123 123 010 032 223 033 232 123 103 020 322 120 110 320 000 020 003 001 1**1**3 303 233 000 3

Linha 6: Rotulamento A: {A = 0, C = 3, G = 1, T = 2}

Olb : 301 202 200 322 303 010 000 120 121 002 033 320 200 133 010 023 301 001 133 222 321 321 030 012 221 011 212 321 301 020 122 320 330 120 000 020 001 003 3**0**1 101 211 000 1

Glb : 301 202 200 322 303 010 000 120 121 002 033 320 200 133 010 023 301 001 133 222 321 321 030 012 221 011 212 321 301 020 122 320 330 120 000 020 001 003 3**3**1 101 211 000 1

Linha 16: Rotulamento A: {A = 2, C = 1, G = 3, T = 0}.

Olb : 123 020 022 100 121 232 222 302 303 220 211 102 022 311 232 201 123 223 311 000 103 103 212 230 003 233 0**3**0 103 123 202 300 102 112 302 222 202 223 221 123 323 033 222 3

Glb : 123 020 022 100 121 232 222 302 303 220 211 102 022 311 232 201 123 223 311 000 103 103 212 230 003 233 0**0**0 103 123 202 300 102 112 302 222 202 223 221 123 323 033 222 3

Linha 18: Rotulamento A: {A = 2, C = 3, G = 1, T = 0}.

Olb : 321 020 022 300 323 212 222 102 101 220 233 302 022 133 212 203 321 221 133 000 301 301 232 210 001 211 0**1**0 301 321 202 100 302 332 102 222 202 221 223 321 121 011 222 1

Glb : 321 020 022 300 323 212 222 102 101 220 233 302 022 133 212 203 321 221 133 000 301 301 232 210 001 211 0**0**0 301 321 202 100 302 332 102 222 202 221 223 321 121 011 222 1

A Tabela 21 apresenta as trocas de bases que ocorreram em cada linha, assim como, seus correspondentes códons e aminoácidos.

Linha	Troca de base	Troca de códon	Troca de aminoácido
2	$A \rightarrow C$	$CAG \rightarrow CCG$	$Glutamina \rightarrow Prolina$
6	$A \rightarrow C$	$CAG \rightarrow CCG$	$Glutamina \rightarrow Prolina$
16	$G \rightarrow T$	$TGT \rightarrow TTT$	$Cisteina \rightarrow Fenilalanina$
18	$G \rightarrow T$	$TGT \rightarrow TTT$	$Cisteina \rightarrow Fenilalanina$

Tabela 21 – Resultado das comparações.

Observamos nesta primeira etapa, que mesmo todas as palavras pertencendo ao rotulamento A, nas linhas 2 e 6 ocorreu a troca da base adenina pela citosina, traduzindo o aminoácido Prolina ao invés da Glutamina e, nas linhas 16 e 18, a troca da base guanina pela timina ocasionou na tradução do aminoácido Fenilalanina ao invés da Cisteína. Essas mutações pontuais alteraram a característica do aminoácido original de polar para não polar. Biologicamente, ainda não se sabe a consequência dessa alteração, mas como não obtivemos o códon de parada, consideraremos essas palavras aptas para serem utilizadas como referência na próxima etapa. Entretanto, dadas as trocas de base diferentes entre as linhas, ao utilizar as linhas 2 ou 6 como referência, não poderemos analisar mutações nas linhas 16 e 18 e, vice-versa.

- **Segunda etapa**: selecionar, dentre as palavras-código resultantes da etapa anterior, uma como referência. Para cada tipo de mutação, alterar pontualmente as bases dessa sequência nas respectivas posições, conforme informadas na Tabela 20. O objetivo é identificar se o código corretor de erro consegue reproduzir a sequência de referência novamente. Para isso, será utilizado o mesmo código BCH da etapa anterior, sem alterar o polinômio primitivo, o gerador e o de verificação de paridade que codificaram a palavra referência.

Apresentamos os resultados para a mutação *nonsense* 4489, considerando as linhas 2 e 16 como referência (**Ref**). E em seguida, os resultados para a mutação *nonsense* 4508, considerando as linhas 6 e 18 como referência. As bases em negrito vermelho são referentes à mutação incluída manualmente; em negrito preto referentes à mutação ocorrida na primeira etapa e; em negrito azul referentes à base gerada na palavra-código resultante após a mutação.

- **Palavra-código referente à permutação 02 incluída a mutação 4489**: $CAGTATTAACT$ T**G**$ACAGAAAAGTAGTGAATACCCTATAAGCCAGAATCCAGAAGGCCTTT$

$CTGCTGACAAGTTTGAGGTGTCTGCAGATAGTTCTACCAGTAAAAATAA$
$AGAACC$**C**$GGAGTGGAAAG.$

Linha 2: {A = 0, C = 1, G = 3, T = 2}.

Ref : 103 202 200 122 **1**01 030 000 320 323 002 011 120 200 311 030 021 103 003 311 222 123 123 010 032 223 033 232 123 103 020 322 120 110 320 000 020 003 001 1**1**3 303 233 000 3

4489 : 103 202 200 122 **3**01 030 000 320 323 002 011 120 200 311 030 021 103 003 311 222 123 123 010 032 223 033 232 123 103 020 322 120 110 320 000 020 003 001 1**1**3 303 233 000 3

Glb : 103 202 200 122 **1**01 030 000 320 323 002 011 120 200 311 030 021 103 003 311 222 123 123 010 032 223 033 232 123 103 020 322 120 110 320 000 020 003 001 1**1**3 303 233 000 3

- **Palavra-código referente à permutação 16 incluída a mutação 4489**: $CAGTATTAACT$
T**G**$ACAGAAAAGTAGTGAATACCCTATAAGCCAGAATCCAGAAGGCCTTT$
$CTGCTGACAAGTTTGAGGT$**T**$TCTGCAGATAGTTCTACCAGTAAAAATAA$
$AGAACCAGGAGTGGAAAG.$

Linha 16: {A = 2, C = 1, G = 3, T = 0}.

Ref : 123 020 022 100 **1**21 232 222 302 303 220 211 102 022 311 232 201 123 223 311 000 103 103 212 230 003 233 0**0**0 103 123 202 300 102 112 302 222 202 223 221 123 323 033 222 3

4489 : 123 020 022 100 **3**21 232 222 302 303 220 211 102 022 311 232 201 123 223 311 000 103 103 212 230 003 233 0**0**0 103 123 202 300 102 112 302 222 202 223 221 123 323 033 222 3

Glb : 123 020 022 100 **1**21 232 222 302 303 220 211 102 022 311 232 201 123 223 311 000 103 103 212 230 003 233 0**0**0 103 123 202 300 102 112 302 222 202 223 221 123 323 033 222 3

- **Palavra-código referente à permutação 06 incluída a mutação 4508:** $CAGTATTAACT$
$TCACAGAAAAGTAGTGAATA$**A**$CCTATAAGCCAGAATCCAGAAGGCCTTT$
$CTGCTGACAAGTTTGAGGTGTCTGCAGATAGTTCTACCAGTAAAAATAA$
$AGAACC$**C**$GGAGTGGAAAG.$

Linha 6: Rotulamento A: {A = 0, C = 3, G = 1, T = 2}.

Ref : 301 202 200 322 303 010 000 120 121 002 0**3**3 320 200 133 010 023 301 001 133 222 321 321 030 012 221 011 212 321 301 020 122 320 330 120 000 020 001 003 3**3**1 101 211 000 1

4508 : 301 202 200 322 303 010 000 120 121 002 0**0**3 320 200 133 010 023 301 001 133 222 321 321 030 012 221 011 212 321 301 020 122 320 330 120 000 020 001 003 3**3**1 101 211 000 1

Glb : 301 202 200 322 303 010 000 120 121 002 0**3**3 320 200 133 010 023 301 001 133 222 321 321 030 012 221 011 212 321 301 020 122 320 330 120 000 020 001 003 3**3**1 101 211 000 1

- **Palavra-código referente à permutação 18 incluída a mutação 4508**: $CAGTATTAACT$ $TCACAGAAAAGTAGTGAATA$**A**$CCTATAAGCCAGAATCCAGAAGGCCTTT$ $CTGCTGACAAGTTTGAGGT$**T**$TCTGCAGATAGTTCTACCAGTAAAAATAA$ $AGAACCAGGAGTGGAAAG$.

  Linha 18: Rotulamento A: {A = 2, C = 3, G = 1, T = 0}.

  Ref : 321 020 022 300 323 212 222 102 101 220 2**3**3 302 022 133 212 203 321 221 133 000 301 301 232 210 001 211 0**0**0 301 321 202 100 302 332 102 222 202 221 223 321 121 011 222 1

  4508 : 321 020 022 300 323 212 222 102 101 220 2**2**3 302 022 133 212 203 321 221 133 000 301 301 232 210 001 211 0**0**0 301 321 202 100 302 332 102 222 202 221 223 321 121 011 222 1

  Glb : 321 020 022 300 323 212 222 102 101 220 2**3**3 302 022 133 212 203 321 221 133 000 301 301 232 210 001 211 0**0**0 301 321 202 100 302 332 102 222 202 221 223 321 121 011 222 1

Agora, apresentamos os resultados para a mutação *missense* 4521, considerando a linha 2 como referência e posteriormente, a linha 16. Em seguida, os resultados para a mutação *missense* 4569, considerando a linha 6 como referência e posteriormente, a linha 18.

- **Palavra-código referente à permutação 02 incluída a mutação 4521**: $CAGTATTAACT$ $TCACAGAAAAGTAGTGAATACCCTATAAGCCAG$**C**$ATCCAGAAGGCCTTT$ $CTGCTGACAAGTTTGAGGTGTCTGCAGATAGTTCTACCAGTAAAAATAA$ $AGAACC$**C**$GGAGTGGAAAG$.

  Linha 2: {A = 0, C = 1, G = 3, T = 2}

Ref : 103 202 200 122 101 030 000 320 323 002 011 120 200 311 03**0** 021 103 003 311 222 123 123 010 032 223 033 232 123 103 020 322 120 110 320 000 020 003 001 1**1**3 303 233 000 3

4521 : 103 202 200 122 101 030 000 320 323 002 011 120 200 311 03**1** 121 103 003 311 222 123 123 010 032 223 033 232 123 103 020 322 120 110 320 000 020 003 001 1**1**3 303 233 000 3

Glb : 103 202 200 122 101 030 000 320 323 002 011 120 200 311 03**0** 021 103 003 311 222 123 123 010 032 223 033 232 123 103 020 322 120 110 320 000 020 003 001 1**1**3 303 233 000 3

- **Palavra-código referente à permutação 16 incluída a mutação 4521**: $CAGTATTAACT$ $TCACAGAAAAGTAGTGAATACCCTATAAGCCAG$**C**$ATCCAGAAGGCCTTT$ $CTGCTGACAAGTTTGAGGT$**T**$TCTGCAGATAGTTCTACCAGTAAAAATAA$ $AGAACCAGGAGTGGAAAG$.

  Linha 16: {A = 2, C = 1, G = 3, T = 0}.

  Ref : 123 020 022 100 121 232 222 302 303 220 211 102 022 311 23**2** 201 123 223 311 000 103 103 212 230 003 233 0**0**0 103 123 202 300 102 112 302 222 202 223 221 123 323 033 222 3

  4521 : 123 020 022 100 121 232 222 302 303 220 211 102 022 311 23**1** 201 123 223 311 000 103 103 212 230 003 233 0**0**0 103 123 202 300 102 112 302 222 202 223 221 123 323 033 222 3

  Glb : 123 020 022 100 121 232 222 302 303 220 211 102 022 311 23**2** 201 123 223 311 000 103 103 212 230 003 233 0**0**0 103 123 202 300 102 112 302 222 202 223 221 123 323 033 222 3

- **Palavra-código referente à permutação 06 incluída a mutação 4569**: $CAGTATTAACT$ $TCACAGAAAAGTAGTGAATACCCTATAAGCCAGAATCCAGAAGGCCTTT$ $CTGCTGACAAGTTTGAGGTGTCTGCAGATAGT$**A**$CTACCAGTAAAAATAA$ $AGAACC$**C**$GGAGTGGAAAG$.

  Linha 6: {A = 0, C = 3, G = 1, T = 2}.

  Ref : 301 202 200 322 303 010 000 120 121 002 033 320 200 133 010 023 301 001 133 222 321 321 030 012 221 011 212 321 301 020 12**2** 320 330 120 000 020 001 003 3**3**1 101 211 000 1

  4569 : 301 202 200 322 303 010 000 120 121 002 033 320 200 133 010 023 301 001 133 222 321 321 030 012 221 011 212 321 301 020 12**0** 320 330 120 000 020 001 003 3**3**1 101 211 000 1

  Glb : 301 202 200 322 303 010 000 120 121 002 033 320 200 133 010 023 301 001 133 222 321 321 030 012 221 011 212 321 301 020 12**2** 320 330 120 000 020 001 003 3**3**1 101 211 000 1

- **Palavra-código referente à permutação 18 incluída a mutação 4569**: $CAGTATTAACT$ $TCACAGAAAAGTAGTGAATACCCTATAAGCCAGAATCCAGAAGGCCTTT$

$CTGCTGACAAGTTTGAGGT\mathbf{T}TCTGCAGATAGT\mathbf{A}CTACCAGTAAAAATAA$
$AGAACCAGGAGTGGAAAG.$

Linha 18: {A = 2, C = 3, G = 1, T = 0}.

Ref : 321 020 022 300 323 212 222 102 101 220 233 302 022 133 212 203 321 221 133 000 301 301 232 210 001 211 0**0**0 301 321 202 10**0** 302 332 102 222 202 221 223 321 121 011 222 1

4569 : 321 020 022 300 323 212 222 102 101 220 233 302 022 133 212 203 321 221 133 000 301 301 232 210 001 211 0**0**0 301 321 202 10**2** 302 332 102 222 202 221 223 321 121 011 222 1

Glb : 321 020 022 300 323 212 222 102 101 220 233 302 022 133 212 203 321 221 133 000 301 301 232 210 001 211 0**0**0 301 321 202 10**0** 302 332 102 222 202 221 223 321 121 011 222 1

Analisando os quatro casos de mutações, para diferentes rotulamentos, observamos que o código corretor de erro reproduziu a palavra-código escolhida como referência. Por esse motivo, essas análises podem contribuir para o desenvolvimento de metodologias que poderão ser aplicadas ou aprimoradas para auxiliar nos diagnósticos de doenças. Consideremos um exemplo de aplicação: suponhamos que um paciente esteja apresentando certos sintomas, mas não identificam qual a causa. Analisando sua sequência de DNA, podemos identificar se ocorreu alguma troca de base nitrogenada, sabendo qual a posição e qual a base resultante dessa troca. A partir desse resultado e baseado em dados existentes nas literaturas sobre as patologias e suas mutações pontuais, podemos apontar que a sequência de entrada apresentava uma troca de base nitrogenada em uma determinada posição similar a uma mutação pontual conhecida. Isto é, conseguiríamos alertar o médico responsável de que as análises apontaram uma possível compatibilidade com determinada(s) doença(s). Contudo, para um diagnóstico definitivo, seria preciso mais estudos, uma vez que para diferentes pacientes, há diferentes respostas referentes a mesma mutação.

5 CONCLUSÕES

As analogias entre o sistema biológico e o sistema de comunicação digital possibilitaram o desenvolvimento de um modelo de sistema para importação genética utilizando códigos BCH sobre a extensão de anel de Galois. Baseado nisso, no presente trabalho, implementamos um algoritmo que foi capaz de identificar e reproduzir sequências de DNA com comprimento 63 e funções biológicas distintas, resultando em palavras-código com um nucleotídeo de diferença da sequência original. Além disso, o algoritmo também foi capaz de analisar mutações em uma sequência de DNA com comprimento 127, onde partindo de uma palavra-código referência com mutação, recuperamos essa palavra-código referência original.

A partir dos resultados obtidos, concluímos que podemos apontar uma estrutura matemática associada aos códigos corretores de erros para a fita simples de DNA, contribuindo para o desenvolvimento de uma metodologia que poderá reduzir o tempo e custos laboratoriais, diagnótico de doenças, análises de mutações, produção de novos fármacos e melhoramento genético. Contudo, verificamos que para sequências extensas e consequentemente, para valores do grau da extensão de Galois altos, a complexidade computacional aumenta, sendo necessário utilizar equipamentos mais potentes. Por fim, como propostas de trabalhos futuros, estudos utilizando diferentes códigos corretores de erros como os convolucionais são interessantes, dado que os códigos BCH possuem um limitante relacionado aos comprimentos das sequências, que devem ser fixos durante toda a execução do algoritmo. Ademais, paralelamente, estudos biológicos também são necessários para analisar as consequências das trocas de bases nitrogenadas ocorridas nas sequências de DNA.

REFERÊNCIAS

ABBAS A. K.; LICHTMAN, A. H.; PILLAI, S. ***Cellular and Molecular Immunology***. 9. ed. [S.l.]: Elsevier/Saunders, 2012.

ALBERTS B.; JOHNSON, A. L. J. R. M. R. K.; WALTER, P. ***Molecular Biology of the Cell***. 4. ed. [S.l.]: Editora Artmed, 2005.

BARBOSA, P. R. **Construção de Códigos Z2k-pseudolineares através de Aplicações Isométricas e Extensões de Galois sobre Anéis Locais**. 2000.

BENEDITO, C. W. O. **Famílias de Reticulados Algébricos e Reticulados Ideais**. São José do Rio Preto, São Paulo - Brasil: [s.n.], 2010.

BIC *An Open Access On-Line Breast Cancer Mutation Data Base*. ***An Open Access On-Line Breast Cancer Mutation Data Base***. 2018. Acesso em: 22 de dezembro de 2018. Disponível em: <www.research.nhgri.nih.gov/bic/>.

BIOLOGIA, B. de. **Ácido Ribonucleico (ARN)**. 2019. Acesso em: 05 de janeiro de 2019. Disponível em: <www.blogdebiologia.com/acido-ribonucleico-arn.html>.

BOERI L.; CANZONIERI, C. C. C. O. F.; DANESINO, C. *Breast cancer and genetics*. ***Journal of Ultrasound Elsevier***, v. 14, p. 171–176, 2011.

BRODY, L. C.; BIESECKER, B. B. *Breast cancer susceptibility genes.BRCA1 and BRCA2*. ***Medicine***, v. 77, n. 3, p. 208–26, 1998.

FARIA, L. C. B. **Existências de Códigos Corretores de Erros e Protocolos de Comunicação em Sequências de DNA**. Tese (Doutorado) — Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, Campinas, São Paulo - Brasil, 2011.

FARIA, L. C. B. e. a. *Error-correcting codes identify mutations in BRCA1 exon14*. ***X-Meeting* 2012**, 2012.

FUTREAL P.A.; LIU, Q. S.-E. D. C. C. e. a. *BRCA1 mutations in primary breast and ovarian carcinomas*. ***Science***, v. 266, p. 120–2, 1994.

GARCIA, A.; LEQUAIN, Y. **Elementos de álgebra**. [S.l.]: Projeto Euclides, IMPA, Rio de Janeiro, 2003.

GONCALVES, A. **Introdução à álgebra**. [S.l.]: Projeto Euclides, IMPA, Rio de Janeiro, 1999.

HALL J. M.; LEE, M. K. N. B. M. J. A. L. H. B.; KING, M. C. *Linkage of early-on-set familial breast cancer to chromosome 17q21*. ***Science***, v. 250, n. 4988, p. 1684–9, 1990.

HERSTEIN, I. N. ***Topics in Algebra***. New York: John Wiley *and* Sons, 1975.

INCA, I. N. de C. **Câncer de ovário**. 2018. Acesso em: 18 de dezembro de 2018. Disponível em: <www.inca.gov.br/tipos-de-cancer/cancer-de-ovario>.

INCA, I. N. de C. **Câncer de mama**. 2019. Acesso em: 15 de janeiro de 2019. Disponível em: <www.inca.gov.br/tipos-de-cancer/cancer-de-mama>.

INTERLANDO, J. C. **Uma Contribuiçãao à Construção e Decodicação de Códigos Lineares sobre Grupos Abelianos via Concatenação de Códigos sobre Anéis de Inteiros Residuais**. Tese (Doutorado) — DT-FEEC-UNICAMP, 1994.

LARSON J. S.; TONKINSON, J. L.; LAI, M. T. *A BRCA1 mutant alters G2-M cell cycle control in human mammary epithelial cells*. ***Cancer Res***, v. 57, p. 3351–5, 1997.

LIN, S.; JR., D. J. C. ***Error control coding: fundamentals and applications***. [S.l.]: Prentice-Hall, 1983.

LODISH B.; MATSUDAIRA, K. K. S.; ZIPURSKY, D. ***Molecular Cell Biology***. 5. ed. [S.l.]: Editora Artmed, 2005.

MCWILLIANS, F. J.; SLOANE, N. J. A. ***The Theory of Error Correcting Code***. [S.l.]: *North Holland Publishing Company*, 1977.

MIKI Y.; SWENSEN, J. S.-E. D. F. P. A. e. a. *A strong candidate for the breast and ovarian cancer susceptibility gene BRCA1*. ***Science***, n. 5182, p. 66–71, 1994.

NCBI *National Center for Biotechnology Information*. ***National Center for Biotechnology Information***. 2018. Acesso em: 15 de outubro de 2018. Disponível em: <www.ncbi.nlm.nih.gov>.

PEREIRA, D. G. **Uma Abordagem Computacional para a Análise de Sequências de DNA por meio dos Códigos Corretores de Erros**. Campinas, São Paulo - Brasil: [s.n.], 2014.

PEREIRA, D. G. e. a. *Analysis of DNA sequences using Error Correcting Codes*. ***Workshop in Advanced Topics in Genomics and Cell biology*** **(ATGC)**, 2013.

PETERSON, W. W.; JR., E. J. W. ***Error-correcting Codes***. 2. ed. [S.l.]: *MIT Press*, 1972.

PETRIN, N. **Sıntese Proteíca**. 2018. Acesso em: 18 de dezembro de 2018. Disponível em: <www.todoestudo.com.br/biologia/sintese-proteica>.

ROCHA, A. S. L. **Modelo de Sistema de Comunicações Digital para o Mecanismo de Importação de Proteínas Mitocondriais através de Códigos Corretores de Erros**. Tese (Doutorado) — Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, Campinas, São Paulo - Brasil, 2010.

RYAN, W. E.; LIN, S. ***Channel Codes: Classical and Modern***. [S.l.]: *Cambridge University Press*, 2009.

SCULLY R.; CHEN, J. P.-A. X. Y. e. a. *Association of BRCA1 with Rad51 in mitotic and meiotic cells*. ***Cell***, v. 88, p. 265–75, 1997.

SHANKAR, P. *On BCH codes over arbitrary integer rings*. ***IEEE Trans. Inform. Theory***, IT-25, p. 480–483, 1979.

SHANNON, C. E. *A Mathematical Theory of Communication*. ***The Bell System Technical Journal***, v. 27, p. 397–423 *and* 623–656, 1948. *Reprinted in*: C.E.Shannon and W.Weaver, eds., *A Mathematical Theory of Communication*, (Univ. of Illinois Press, Urbana, Illinois, (1963)).

SOMASUNDARAM K.; ZHANG, H. Z.-Y.-X. H. Y. e. a. *Arrest of the cell cycle by the tumour-suppressor BRCA1 requires the CDK-inhibitor p21WAF1/CiP1*. ***Nature***, v. 389, p. 187–90, 1997.

SZABO, C. I.; KING, M. C. *Population genetics of BRCA1 and BRCA2*. ***Am J Hum Genet***, p. 1013–20, 1997.

VITERBI, A.; OMURA, J. K. ***Principles of digital communication and coding***. [S.l.]: McGraw-Hill, 1979.