UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Engenharia Elétrica e de Computação

Mario Enrique Duarte Gonzalez

Modelagem da Síntese de Proteínas e sua Estrutura Organizacional através de Códigos Corretores de Erros

Campinas

2017

Mario Enrique Duarte Gonzalez

Modelagem da Síntese de Proteínas e sua Estrutura Organizacional através de Códigos Corretores de Erros

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica, na Área de Telecomunicações e Telemática.

Orientador: Prof. Dr. Reginaldo Palazzo Jr.

Este exemplar corresponde à versão final da tese defendida pelo aluno Mario Enrique Duarte Gonzalez, e orientada pelo Prof. Dr. Reginaldo Palazzo Jr.

Campinas

2017

Agência(s) de fomento e nº(s) de processo(s): CAPES
ORCID: http://orcid.org/http://orcid.org/00

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

D85m

Duarte Gonzalez, Mario Enrique, 1986-
Modelagem da síntese de proteínas e sua estrutura organizacional através de códigos corretores de erros / Mario Enrique Duarte Gonzalez. – Campinas, SP : [s.n.], 2017.

Orientador: Reginaldo Palazzo Junior.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Teoria da codificação. 2. Genética - Modelos matemáticos. 3. Anéis (Álgebra) - Codificação. 4. Proteinas - Modelos matemáticos. 5. Códigos corretores de erros (Teoria da informação). I. Palazzo Junior, Reginaldo,1951-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Modeling of protein synthesis and its organizational structure by the use of error correcting codes
Palavras-chave em inglês:
Coding theory
Genetic - Mathematical models
Rings (Algebra) - Coding
Proteins - Mathematical models
Error correcting codes (Information theory)
Área de concentração: Telecomunicações e Telemática
Titulação: Doutor em Engenharia Elétrica
Banca examinadora:
Reginaldo Palazzo Junior [Orientador]
Carlos Eduardo Câmara
Antonio Aparecido de Andrade
Marcelo Mendes Brandão
Mário Henrique Bengtson
Data de defesa: 23-02-2017
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Mario Enrique Duarte Gonzalez
RA: 121502

Data da Defesa: 23 de fevereiro de 2017

Título da Tese: "Modelagem da Síntese de Proteínas e sua Estrutura Organizacional através de Códigos Corretores de Erros"

Prof. Dr. Reginaldo Palazzo Jr.(Presidente, FEEC/UNICAMP)
Prof. Dr. Carlos Eduardo Câmara (UNIANCHIETA)
Prof. Dr. Antonio Aparecido de Andrade (IBILCE/UNESP)
Prof. Dr. Marcelo Mendes Brandão (CBMEG/UNICAMP)
Prof. Dr. Mario Henrique Bengtson (IB/UNICAMP)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no processo de vida acadêmica do aluno.

À todas as pessoas que contribuíram na minha formação como pessoa e como pesquisador;

À minha filha, Luciana, e à minha esposa, Ángela, pela paciência, apoio, estímulo e amor;

Aos meus pais, Maria del Pilar e Germán, pelo apoio continuo e incondicional em todos os momentos da minha vida;

Aos meus irmãos, Luis Felipe e Germán Darío, pelo companheirismo, compreensão e amor que nos une;

À TODA minha grande família.

Dedico

Agradecimentos

Ao meu orientador Prof. Reginaldo Palazzo Jr. por ter me acolhido no programa de doutorado, pelo aconselhamento, pelo conhecimento compartilhado, pelo apoio incondicional para o meu desenvolvimento pessoal e acadêmico e pela confiança depositada em mim para a conclusão deste trabalho.

Ao Professor e grande amigo Fernando Torres por ter me mostrado a linda área da matemática pela qual me apaixonei. Sua paixão e conhecimento compartilhado foram fundamentais para a conclusão deste trabalho. Saudades das nossas reuniões, regadas de discussões matemáticas e futebol.

Aos Professores Evandro Mazina e Milton Romero por me orientar e por me mostrar o caminho da pesquisa, sempre lembrarei das suas experiências de vida e conselhos.

Aos professores membros da banca examinadora pela disponibilidade e atenção dispensada ao trabalho, bem como por suas valiosas contribuições.

Aos professores do Doutorado em Engenharia Elétrica que contribuíram na minha formação, especialmente ao Prof. Anésio dos Santos por compartilhar suas experiências de vida, foi um grande prazer ser PED sob sua orientação.

Aos funcionários da pós-graduação por ser muito atenciosos e por sempre realizar um trabalho de altíssima qualidade.

À minha querida companheira Angela María pela sua constante dedicação e compreensão. Sou-lhe eternamente grato pela sua paciência, amizade, amor e por ter me dado o presente mais lindo que a vida pode dar: Luciana!

Aos meus pais Maria del Pilar e Luis Germán por ter me mostrado a importância da formação acadêmica, pelo apoio e incentivo incondicional, necessários para a conclusão desta etapa na minha vida, e por me ensinar a nunca desistir dos meus sonhos.

Aos meus irmãos Luis Felipe e Germán Darío por confiar e acreditar em mim. Como irmão mais velho, a razão para eu sempre fazer as coisas direito é ser o motivo de

inspiração para vocês dois.

À minha família TODA, em especial as minhas avôs Maria Otilia Duarte e Beatriz Rengifo e a aquelas pessoas que ficaram na memória de todos nós, por me ensinar e acompanhar no caminho da vida e por formar a pessoa que sou. A todos vocês peço perdão pelas horas de ausência.

Não poderia deixar de agradecer às pessoas que fizeram agradável a minha permanência no Brasil: Leandro, Cintya, Anderson, Luzinete, Luiz, Diogo, Maicon, Gustavo, Renato, Rodrigo, Marcos, André, Márcia, Nelson, Akemi, Carlos, Cibele, Felipe, Lina, Sylvia, Carol e muitos mais.... Vocês foram, são e serão amigos incondicionais. Obrigado!

Por último, agradeço à CAPES pelo financiamento do meu aperfeiçoamento através da concessão da bolsa de estudos de doutorado.

"O homem erudito é um descobridor
de fatos que já existem - mas o homem
sábio é um criador de valores que
não existem e que ele faz existir".

(Albert Einstein, 1879 - 1955)

"O gênio, esse poder que deslumbra os
olhos humanos, não é outra coisa senão
a perseverança disfarçada".

(Johann Wolfgang von Goethe, 1749 - 1832)

Resumo

O uso da teoria da informação e codificação nos sistemas biológicos emergiu como uma alternativa para a explicação de como o sistema biológico é capaz de manipular a informação genética nos processos de replicação de DNA e síntese de proteínas. Alguns modelos para o sistema biológico, decorrentes dessas pesquisas, têm sido estabelecidos no contexto da teoria da informação. Nos últimos trabalhos deste grupo de pesquisa, introduziu-se um modelo inovador para a importação de proteínas organelares, com o qual, foi possível mostrar que algumas sequências moleculares (DNA, RNA e mRNA) têm uma estrutura matemática no contexto da teoria de codificação. Utilizando o recente modelo, a proposta deste trabalho consiste, primeiro, em demonstrar alguns fatos matemáticos que ficaram em aberto em trabalhos recentes e, segundo, em validar as hipóteses que o Ribossomo age como o modulador do sistema biológico e que as proteínas possuem uma estrutura organizacional similar à estrutura de um código corretor de erros. Para realizar a validação das hipóteses, definiram-se duas possíveis abordagens. A primeira consistiu em mostrar que as sequências mRNA podem ser codificadas por um código BCH sobre $\mathbb{Z}_4$ ou $\mathbb{F}_4$ e que o ribossomo age simplesmente como o mapa sobrejetor de códons para aminoácidos de acordo com o "Código Genético". Assim a estrutura organizacional das proteínas é herdada deste código corretor de erros presente nas sequências mRNA. A segunda abordagem, inspirou-se no trabalho de Zyablov, que introduz o conceito de *Codificador Genético Concatenado*, o qual unifica o codificador com o modulador, e verifica que as proteínas podem ser codificadas por um código cíclico sobre $\mathbb{Z}_{20}$ ou $\mathbb{F}_4 \times \mathbb{Z}_5$, construído através de códigos BCH. Os resultados obtidos mostraram a existência de uma estrutura matemática nas proteínas, no contexto de teoria de comunicação, a qual, permitiu agrupar algumas proteínas *Cythocrome b6-f complex subunit 6-OS* de diferentes organismos em um único grupo monofilético com um controle de replicação de 99%, o qual não é usual inclusive para proteínas altamente conservadas evolutivamente. Finalmente, este trabalho contribui no desenvolvimento de uma metodologia que poderá ser aplicada em produção de novos fármacos, melhoramento genético, entre outros.

Palavras-chaves: Codificação genética, Teorema chinês do resto, Códigos sobre anéis, Códigos BCH sobre corpos e anéis, Identificação de proteínas, Ribossomo, Codificador genético concatenado.

Abstract

The use of information and coding theory in biological systems emerged as an alternative to explain how the biological system is able to manipulate genetic information in the DNA replication and protein synthesis processes. Some models for the biological system, derived from those researches, have been established on the information theory framework. In recent works from this research group, a novel model for the import of organelles protein has been introduced; this model allowed to show that some molecular sequences (DNA, RNA and mRNA) have a mathematical structure in the coding theory framework. Using the recent model, this work proposal was, first, to demonstrate some mathematical facts that remain open from recent studies and, second, to validate the hypotheses that the Ribosome acts as the modulator of the biological system and that the proteins have an structural organization similar to the structure of error-correcting codes. In order to validate these hypothesis, two possible approaches were proposed. The first one consists in showing that mRNA sequences can be encoded by a BCH code over $\mathbb{Z}_4$ or $\mathbb{F}_4$ and that the ribosome acts simply as a surjective map from codons to amino acids according to the "Genetic Code", thus, the structural organization of proteins is inherited from the error-correcting code present in the mRNA sequences. The second approach was inspired by Zyablov's work, and it introduces the concept of the *Concatenated Genetic Encoder*, which unifies the encoder and the modulator, and verifies that the proteins can be encoded by a cyclic code over $\mathbb{Z}_{20}$ or $\mathbb{F}_4 \times \mathbb{Z}_5$, constructed by BCH codes. The obtained results demonstrate the existence of a mathematical structure within proteins in the communication theory framework, which allowed to group some *Cythocrome b6-f complex subunit 6-OS* proteins of different organisms in a single monophyletic group with a replication control of 99%, not usual even for highly evolutionarily conserved proteins. Finally, this work contributes to the development of a methodology that can be applied in the production of new drugs, genetic improvement, among others.

Keywords: Genetic coding, Chinese Remainder Theorem, Codes over rings, BCH codes over fields and rings, Protein identification, Ribosome, Concatenated genetic encoder.

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos e Notação

DNA	*Deoxyribonucleic acid.*
RNA	*Ribonucleic Acid.*
mRNA	*messenger Ribonucleic Acid.*
tRNA	*transfer Ribonucleic Acid.*
miRNA	*micro Ribonucleic Acid.*
rRNA	*ribosomal Ribonucleic Acid.*
SNP	*Single Nucleotide Polymorphism.*
Oaa	Sequência original de aminoácidos.
Ont	Sequência original de nucleotídeos.
Olb	Sequência original rotulada.
Gaa	Sequência gerada de aminoácidos.
Gnt	Sequência gerada de nucleotídeos.
Glb	Sequência gerada e rotulada.
CG	Código Genético.
PF	Proteína Funcional.
PsS_i	Proteínas semelhantes.
PCR	*Polymerase Chain Reaction.*
NCBI	*National Center for Biotechnology Information.*
MC	*Mapeamento Casado.*
CCE	*Código Corretor de Erros.*
$\mathbb{F}_q$	Corpo finito com q elementos.
$GF(q)$	Corpo finito com q elementos.
$\mathbb{Z}_q$	Anel dos inteiros módulo q.

BCH	*Bose, R. C.; Ray-Chaudhuri, D. K.*
nsBCH	*narrow-sense BCH*. BCH no sentido estrito.
revBCH	BCH reversível.
CRT	Código baseado no Teorema Chinês do Resto.
ECRT	Código Estendido baseado no Teorema Chinês do Resto.
MPs	Polinômios Minimais.
$(\mathbb{A}, *)$	Grupo. Conjunto $\mathbb{A}$ munido com a operação $*$.
$(\mathbb{A}, +, \cdot)$	Anel. Conjunto $\mathbb{A}$ munido com as operações $+$ e $\cdot$.
$d_H(x, y)$	Distância de Hamming entre as sequências x e y.
$d(\mathcal{C})$	Distância Mínima de Hamming do código $\mathcal{C}$.
$GR(q, s)$	Extensão do anel com q elementos, usando um polinômio de grau s.
$GF(q, s)$	Extensão do corpo com q elementos, usando um polinômio de grau s.
$\frac{\mathbb{A}[x]}{\langle f(x) \rangle}$	Anel residual dos polinômios $\mathbb{A}[x]$ com o polinômio $f(x)$.

Sumário

1 Introdução

A informação genética, a qual permite a variabilidade e o correto funcionamento dos organismos vivos, está codificada como sequências de nucleotídeos no DNA (*Deoxyribonucleic acid*). A organização da informação genética e sua habilidade para ser preservada e traduzida em proteínas com taxas baixas de erro é o interesse tanto de biologistas como de matemáticos e engenheiros. O ponto em comum entre estas três áreas é a teoria da informação e codificação, a qual é utilizada para explicar a confiabilidade na transmissão da informação genética (BARBIERI, 2008). A aplicabilidade da teoria sugere que o sistema biológico, de maneira similar como o sistema de transmissão digital, deve ser capaz de detetar e corrigir erros tanto no processo de replicação de DNA e no processo de síntese (tradução) de proteínas, posto que as proteínas agem como mensageiras inter e intra celular para garantir o correto funcionamento do organismo vivo.

Os sistemas de comunicação digital tem sido amplamente utilizados e desenvolvidos para a transmissão confiável de informação (mensagens) no espaço de um lugar a outro. De maneira análoga o DNA é transmitido no tempo de geração em geração (BATTAIL, 2007a) e as proteínas no espaço inter e intra celular. Este trabalho visa estudar o processo de síntese de proteínas e apresenta um modelo para tal processo no contexto de teoria de codificação. Ainda assim, pelo fato das sequências mRNA (*messenger Ribonucleic acid*) e DNA serem sequências de nucleotídeos, muitos resultados desenvolvidos neste trabalho também se aplicam no processo de replicação do DNA.

Este capítulo está organizado da seguinte maneira. Na Seção 1.1, apresenta-se um breve histórico dos avanços na teoria de comunicação e biologia molecular. Na Seção 1.2, explicam-se os modelos existentes na literatura para a replicação do DNA e a síntese de proteínas no contexto da teoria da informação e codificação. Na Seção 1.3, detalha-se a proposta do presente trabalho com a finalidade de mostrar que o ribossomo age como um modulador restrito ao *Código Genético* e que o processo de síntese de proteínas pode ser modelado como um único codificador; finalmente, introduz-se e justifica-se o conceito de ***codificador genético concatenado***. Por último, na Seção 1.4, indica-se a descrição do trabalho.

Antes de continuar e para facilitar a compreensão dos conceitos introduzidos neste capítulo, definem-se e explicam-se alguns fatos da biologia molecular e a teoria de codificação.

Os ribossomos são organelas presentes nas células eucarióticas e procarióticas cuja principal função é a síntese de proteínas usadas pela célula. Estas organelas leem as sequências mRNA, as quais são formadas por nucleotídeos $\mathbb{N} = \{A, C, G, U\}$ e estão definidas pe-

las sequências de DNA, e constroem as sequências de aminoácidos ($\{E, D, N, Q, K, R, H, W, Y, F, L, M, I, V, C, T, S, A, G, P\}$) (polipeptídeos) ou proteínas. A tradução de mRNA para proteínas ocorre segundo o mapeamento estipulado pelo *Código Genético*. Note-se que, apesar do nome, o "Código Genético" será estudado como um mapa sobrejetivo de códons (tripletas de nucleotídeos) para aminoácidos, i.e.

$$CG : (\mathbb{N} \times \mathbb{N} \times \mathbb{N}) \to \mathbb{A}$$

Neste trabalho, assume-se a denominação de **proteínas funcionais** da seguinte maneira. No conjunto de todas as possíveis proteínas (ou sequências de aminoácidos) encontra-se o subconjunto das proteínas funcionais (PF). As proteínas funcionais são aquelas que têm uma funcionalidade biológica e, portanto, participam direta ou indiretamente em processos dentro da célula e são responsáveis pela catalização e regularização de reações bioquímicas, transporte de moléculas, e pela formação de estruturas básicas como: pele, cabelo, entre outras. Portanto, entender e modelar o funcionamento das PF é um grande desafio para a comunidade científica (LODISH, 2008b), em especial para as áreas de: farmacêutica (projeto de medicamentos) e biotecnologia (projeto de novas enzimas).

1.1 Os Avanços Históricos

A metade do século passado foi uma época inovadora tanto para a engenharia nos sistemas de comunicação como para a biologia com o advento da estrutura dupla hélice do DNA. Em 1953, a estrutura do DNA foi descoberta por James Watson, Francis Crick, Maurice Wilkins e Rosalind Franklin, mostrando que toda a informação genética está armazenada na estrutura de duas fitas complementares e formadas como duas sequências sobre um alfabeto com unicamente quatro moléculas (bases nucleicas). Esta descoberta mudou a maneira como a pesquisa na genética estava sendo desenvolvida, pois, antes disso, a genética estava sendo modelada segundo as regras propostas pelo austríaco Gregor Mendel. A elucidação da estrutura do DNA revolucionou a ciência da vida, viabilizou o desenvolvimento de tecnologias baseadas no DNA e fundamentou o lançamento de industrias biotecnológicas com aplicações médicas, como por exemplo, a identificação dos mecanismos genéticos que interferem em doenças graves como diabetes, hipertensão e câncer.

Em 1965, foi demonstrado que células normais dividiam-se apenas em um número limitado de vezes (o limite de Hayflick), isto é, as células envelhecem e morrem. Porém, também encontrou-se que as células tronco são uma exceção a esta regra. Em 1961, Nirenberg e Matthaei definiram o experimento: *The Nirenberg and Matthaei experiment*, o qual decifrou o código genético ao utilizar homopolímeros (ácidos nucleicos) na tradução

de aminoácidos específicos. Em 1983, introduz-se a técnica PCR (*polymerase chain reaction*), a qual é utilizada em laboratórios de pesquisas médicas e biológicas para diversas tarefas, como o sequenciamento de genes e diagnóstico de doenças hereditárias, identificação de "impressão digital" genético (usado em testes de paternidade e na medicina forense), detecção de diagnóstico de doenças infecciosas e criação de organismos. Projetos nesta direção permitiram mapear o material genético completo e, assim, identificar cada um dos 100 mil genes do corpo humano.

Em 1948, Claude Elwood Shannon publicou o artigo intitulado "*A Mathematical Theory of Communication*", no qual estabeleceu a teoria fundamental de um sistema de comunicação digital, introduzindo o conceito de informação baseado somente na característica estatística da fonte de informação, definindo a informação de maneira abstrata independente da semântica que não diferencia texto, vídeo ou áudio. Esta definição de informação tem contribuído nas áreas de compressão, codificação de canal e criptografia.

A matemática surge nos sistemas de comunicação pois brinda a precisão e a eficácia do fluxo informacional, porém, a maior característica do modelo proposto por Shannon é a abrangência. Este modelo não se restringe apenas à engenharia, mas, serve de referência a qualquer âmbito da comunicação e é adaptável a qualquer processo de comunicação, independentemente das características semânticas dos seus componentes.

A primeiro grande resultado é o primeiro teorema de Shannon (teorema de codificação de fonte), o qual prova que a mensagem gerada por uma fonte de informação pode ser compactada até o limite da entropia da fonte. O segundo grande resultado é o segundo teorema de Shanon (teorema da codificação de canal), o qual mostra a existência de um sistema de codificação que permite a transmissão de informação livre de erros com uma taxa máxima que o canal permite. Note-se que os teoremas de Shannon mostram a existência de tais sistemas de comunicação, porém, não indicam a sua estrutura. Desde então, a engenharia de comunicações tem criado algoritmos e estratégias que visam atingir os limitantes destes dois teoremas.

Sem muito sucesso, o uso da teoria da informação para a análises de dados genéticos iniciou-se na década de 1970 e atualmente um número muito reduzido de pesquisas têm sido desenvolvidas nesta direção. As pesquisas buscam analogias entre o fluxo de informação biológica e o sistema de comunicação e que permitam uma melhor compreensão dos paradigmas biológicos. Na outra direção, note-se que os desafios e fatos encontrados na modelagem dos sistema biológicos podem aportar para as teorias da informação, comunicação e codificação. Entre as principais contribuições e desafios desta área de pesquisa encontram-se: compreender como as interferências afetam os sistemas biológicos; descobrir como as propriedades da teoria da informação e codificação podem ser adaptadas para sua aplicação em sistemas moleculares; compreender como a informação codificada no DNA é gerada e usada em proteínas; entre outras.

1.2 Modelos Propostos na Literatura

Em (BATTAIL, 2007a), afirma-se que a teoria da informação e codificação devem estar presentes nos sistemas biológicos dado que o DNA deve ser transmitido no tempo de geração em geração (BATTAIL, 2007a) e as proteínas no espaço inter e intra celular. A diferença entre os sistemas tradicionais e os sistemas biológicos é que o primeiro é realizado pelo homem enquanto que a genética e a síntese de proteínas são realizadas por um processo natural. Embora a teoria da informação e codificação tenham desenvolvido um poderoso ferramental conceitual que visa a transmissão confiável e eficiente de mensagens de um lugar para outro, a pergunta que surge é: a estrutura teórica desenvolvida pelo homem pode ajudar na elucidação dos processos naturais que envolvem a informação biológica? Segundo Battail (BATTAIL, 2007a), a resposta é positiva e pode renovar a visão atual do "mundo vivo".

Pelo crescente número de dados genéticos e proteicos, nos últimos anos tem sido evidenciado um grande número de novas propostas que usam a teoria da informação e codificação como fundamento matemático. Schneider (SCHNEIDER *et al.*, 1986; SCHNEIDER *et al.*, 1990; SCHNEIDER; SPOUGE, 1997) apresenta um procedimento sistemático, baseado na teoria da informação, para identificar regiões codantes e não codantes nas sequências de DNA. Yockey (YOCKEY, 1992) apresentou um modelo para a expressão gênica associado com o sistema tradicional de comunicação digital. Forsdyke (FORSDYKE, 1981; FORSDYKE, 1995) considerou os exons como mensagens de informação e os introns como os dígitos de verificação de paridade. Rzeszowska-Wolny (RZESZOWSKA-WOLNY, 1983), afirma que a operacionalidade do DNA está baseada em um arranjo apropriado do DNA em nucleossomos. Liebovitch (LIEBOVITCH *et al.*, 1996) introduz uma metodologia para determinar se um código corretor de erros está presente na estrutura do DNA. Rosen (ROSEN, 2006) apresentou um método para a detecção de códigos de bloco lineares que permite a possibilidade de inserções e deleções nas sequências de DNA, além disso sugere o uso da teoria algébrica para elucidar a relação entre os códigos e a biologia. Battail (BATTAIL, 2007b) propõe a existência de códigos aninhados no DNA, uma vez que a evolução tem ocorrido acumulativamente e passo a passo. May (MAY *et al.*, 2004) introduz um modelo para o sistema genético e sugere a possibilidade de projetar códigos convolucionais para diferenciar sequências gênicas que codificam e não codificam proteínas. Mac Donnaill (DÓNAILL, 2003) propôs um código de verificação de paridade relacionado à composição dos nucleotídeos. Andrea e Luzinete (ROCHA *et al.*, 2010; FARIA *et al.*, 2010) mostram a existência de estruturas matemáticas em sequências de DNA relacionadas com códigos BCH sobre $\mathbb{Z}_4$ e $\mathbb{F}_4$. Na mesma direção, Debata (DEBATA *et al.*, 2012) indica a existência de algum código corretor de erros ao codificar sequências de DNA através de códigos de Hamming. Em (FARIA *et al.*, 2012) mostrou-se que o genoma completo do plasmídeo *Lactococcus lactis* pode ser identificado como palavra-código de

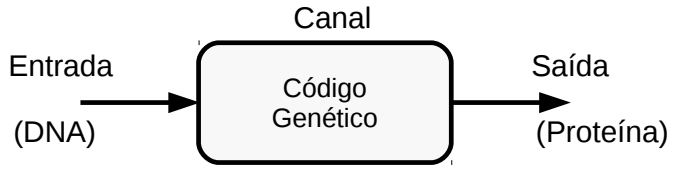


Figura 1 – Uso da teoria da informação de acordo com o modelo de (ROMÁN-ROLDÁN *et al.*, 1996)

um código cíclico de Hamming. Finalmente, em (FARIA *et al.*, 2014) estabelece-se os elementos fundamentais e a caracterização matemática de uma proposta para o sistema de transmissão intra-celular de informação genética.

Apesar dos trabalhos mencionados anteriormente buscarem um ponto em comum entre a teoria da informação e codificação com a biologia molecular, nem todos se adéquam ao mesmo modelo. A seguir, apresentam-se os três principais modelos encontrados na literatura, nos quais se evidenciará que o processo de **síntese de proteínas** recebe uma interpretação diferente para cada um deles.

A Figura 1, mostra o uso da teoria da informação para a síntese de proteínas de acordo com o modelo introduzido em (ROMÁN-ROLDÁN *et al.*, 1996). No qual a síntese de proteínas pode ser considerada como um sistema de processamento da informação permitindo que as sequências de nucleotídeos possam ser analisadas como mensagens para o processamento da informação cuja saída é uma proteína, isto é, a transferência da informação biológica pode ser modelada por um sistema de comunicação que considera a sequência de DNA como a entrada do canal e a sequência de aminoácidos (proteína) como a saída do canal.

O modelo definido pela May (MAY, 1998) é mostrado na Figura 2. Neste modelo, observa-se que as sequências de DNA codificam uma informação genética, a qual indica a maneira de agir da célula e, portanto, do organismo, que o RNA mensageiro (mRNA) é a saída do canal de comunicação, bloco no qual ocorrem os erros, e que o decodificador traduz o mRNA em proteína. Note-se que os modelos anteriores consideram a proteína como a forma final da informação. Logo, estes modelos não conseguem explicar a redundância existente na estrutura das proteínas; por exemplo, quando consideradas proteínas de tamanho 21, há $20^{21} = 2.097152 \times 10^{27}$ possíveis sequências de aminoácidos, porém, somente algumas dessas sequências são proteínas biologicamente funcionais.

O terceiro modelo foi introduzido em (ROCHA, 2010; FARIA *et al.*, 2014), é mostrado na Figura 3 e corresponde a um modelo para a importação de proteínas organelares. Neste modelo, incorpora-se um codificador e um modulador. A palavra-código na saída do codificador está associada à sequência de nucleotídeos (mRNA), enquanto que a saída do modulador está associada à sequência de aminoácidos. O mapeamento entre o códon e anti-códon é realizado pelo RNA transportador (tRNA) e é caracterizado, no contexto de

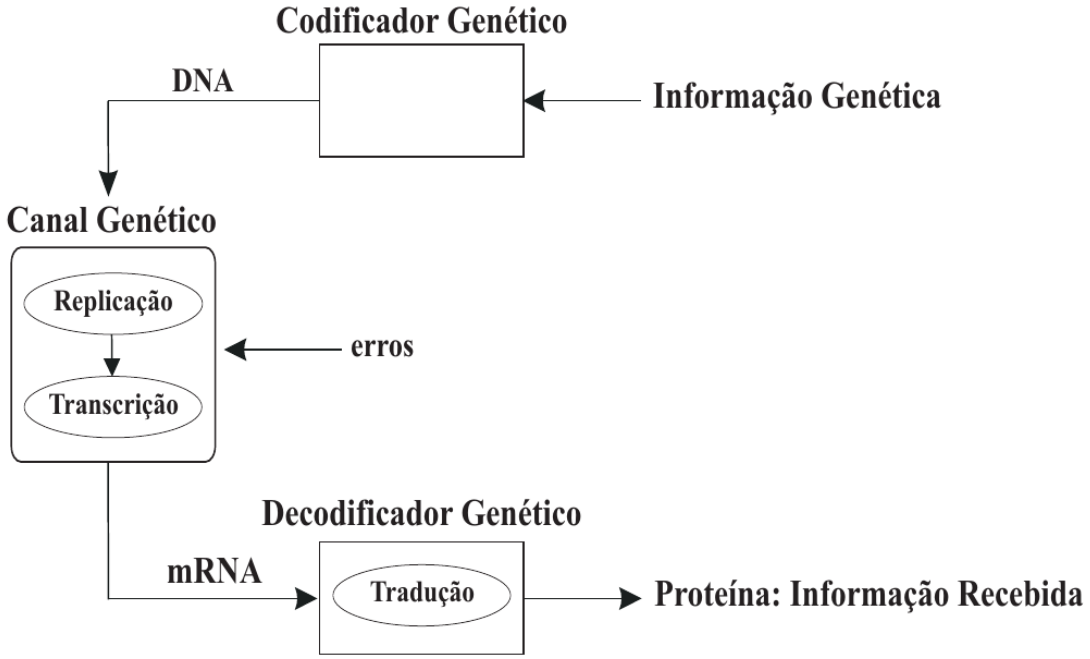


Figura 2 – Uso da teoria da informação de acordo com o modelo de (MAY, 1998)

um sistema de comunicação digital, como um mapeamento definido pelo código genético.

No modelo, o RNA transportador se comporta de maneira equivalente ao *mapeamento casado* (MC) atuando num sistema de comunicação. Este MC permite que a estrutura algébrica do codificador seja a mesma, a menos de um isomorfismo, que a da constelação de sinais, garantindo assim, a menor complexidade possível do sistema. Apesar de usar um codificador BCH no modelo, este codificador pode ser não BCH. Assumiram-se tais códigos pela complexidade do problema e porque a estrutura desses códigos é bem conhecida e são fáceis de projetar.

Assim, o modulador consiste do código genético, do RNA transportador e do Ribossomo. O código genético pode ser visto como uma constelação de sinais, onde cada códon é considerado como um sinal da constelação, o RNA transportador realiza o mapeamento casado, enquanto que o Ribossomo se comporta como um processador digital de sinais.

1.3 Proposta de Pesquisa

Para o presente trabalho, utiliza-se o modelo da Figura 3 e verifica-se a hipótese que o Ribossomo age como um processador de sinais e como o modulador do sistema biológico e que as proteínas são as mensagens codificadas que são enviadas através do canal.

A hipótese de que as proteínas são as mensagens codificadas no DNA e que permitem a comunicação inter e intra celular ganha força pelo fato da existência de redun-

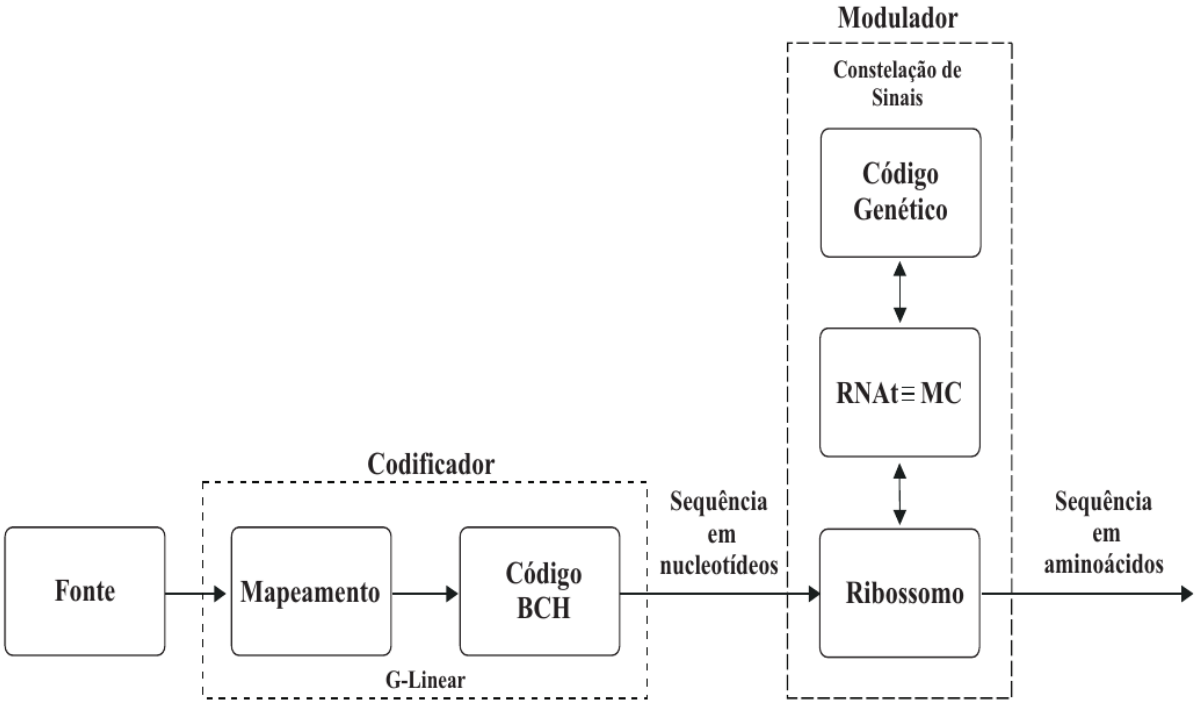


Figura 3 – Uso da teoria da informação de acordo com o modelo de (ROCHA, 2010; FARIA *et al.*, 2014)

dância nas proteínas; por exemplo, quando consideradas proteínas de tamanho 21, há $20^{21} \approx 2.097152 \times 10^{27}$ possíveis sequências de aminoácidos, porém, somente algumas dessas sequências são proteínas biologicamente funcionais. Além disso, pequenas mutações nas proteínas funcionais são toleradas pelo sistema biológico e preservando ainda a atividade biológica, isto é, a nova proteína com a mutação se encontra dentro da capacidade de correção do código das proteínas. Além disso, a célula possui complexos biológicos, denominados *chaperonas*, para a verificação de erros, que eliminam proteínas sintetizadas ou dobradas incorretamente. As proteínas dobradas incorretamente geralmente não possuem atividade biológica e, em alguns casos, podem estar associadas a doenças (BERG *et al.*, 2003). Portanto, a existência das chaperonas também justifica o uso de códigos corretores de erro (CCE) para representar as proteínas.

Lembre-se que a sequência linear de aminoácidos é o elo entre a mensagem genética no DNA e a estrutura tridimensional que executa a função biológica de uma proteína (BERG *et al.*, 2003; ROKDE; KSHIRSAGAR, 2013). O princípio central da biologia molecular estabelece que *a sequência linear de aminoácidos determina a conformação tridimensional da proteína* (LODISH, 2008b; BERG *et al.*, 2003) e a **conformação define o funcionamento da proteína**.

Os sistemas de comunicação enviam mensagens através de um canal de informação susceptível a ocorrência de erros. As mensagens são codificadas como sequências de elementos que pertencem a um alfabeto $\mathbb{A}$. Assim, um *código* $\mathcal{C}$ é um subconjunto formado por todas as sequências que codificam uma única mensagem e são definidas como

palavras-código.

Um *Código Corretor de Erros (CCE)* é um código que, no processo de decodificação, permite corrigir palavras-código com pequenas variações (erros), as quais foram adicionadas no envio através do canal de comunicação, e assim, permite reconhecer a mensagem que gerou a sequência no codificador.

A capacidade de correção de erros está associada ao parâmetro "*distância mínima*" de acordo com a função de medida que indica a "*distância*" entre sequências. Logo, a *distância mínima* determina a quantidade máxima de erros adicionados na palavra-código, tal que, o código corretor de erros é capaz de detectar ou corrigir.

Segundo as ideias expostas, evidência-se uma correspondência entre os CCEs e o sistema de codificação de proteínas; a qual é mostrada na Tabela 1.

Sistema biológico de codificação		**Sistema de codificação de canal**
Proteínas		Sequências de elementos de A
Proteínas Funcionais (PF)		Palavras-código (c_i)
Conjunto de todas as PF		Código corretor de erro (C)
Proteínas semelhantes (PsS_i)	$\Leftrightarrow$	Sequências corregíveis (Cc_i)
Número de variações que mantêm a semelhança biológica da proteína		Distância mínima

Tabela 1 – Correspondência entre a codificação de proteínas e a codificação de canal.

A proposta deste trabalho consiste em validar a hipótese do Ribossomo agir como o modulador do sistema biológico e que as proteínas possuem uma estrutura organizacional equivalente à estrutura de um código corretor de erros. Para realizar essa validação, propõem-se duas possíveis abordagens. A primeira consiste em mostrar que as sequências mRNA podem ser codificadas por um código corretor de erros e que o ribossomo age simplesmente como um mapa sobrejetor de códons para aminoácidos, assim a estrutura organizacional das proteínas é herdada do código corretor de erros presente nas sequências mRNA. A segunda abordagem, inspira-se no trabalho de Zyablov (ZYABLOV *et al.*, 1999), no qual mostra-se que o codificador e o modulador num sistema de transmissão digital podem ser unificados em um único codificador (*código concatenado*), assim, neste trabalho, propõe-se o conceito de *Codificador Genético Concatenado* e projetam-se códigos corretores de erros que modelam o codificador concatenado tais que algumas proteínas funcionais são palavras-código.

Para as duas abordagens mencionadas acima, usam-se códigos BCH para a construção do codificador genético e do codificador genético concatenado porque estes códigos são bem conhecidos e, além disso, pode-se obter um algoritmo sistemático relativamente

simples para a sua construção. Para a primeira abordagem projetam-se códigos BCH sobre $\mathbb{Z}_4$ e $\mathbb{F}_4$ tais que uma dada sequência mRNA é uma palavra-código, isto é, uma estrutura matemática, no contexto da teoria da informação, é identificada para essa sequência mRNA e, portanto, para a proteína na qual esta é traduzida. Na segunda abordagem, constroem-se códigos cíclicos sobre $\mathbb{Z}_{20}$ e $\mathbb{F}_4 \times \mathbb{Z}_5$, a partir de códigos BCH sobre $\mathbb{Z}_4$, $\mathbb{Z}_5$ e $\mathbb{F}_4$, tais que uma dada proteína é uma palavra-código, isto é, encontra-se uma estrutura matemática, no contexto da teoria da informação, que identifica esta proteína.

1.4 Descrição do Trabalho

Este trabalho está organizado da seguinte forma. O Capítulo 2 é introdutório, nele apresenta-se de forma sucinta alguns conceitos sobre a biologia celular e molecular com ênfase na síntese de proteínas e as principais definições e fatos referentes à teoria de códigos corretores de erros.

Os Capítulos 3, 4, 5 e 6 contêm as contribuições deste trabalho. No Capítulo 3 introduz-se e detalha-se um algoritmo para encontrar códigos BCH (incluindo a subclasse *narrow-sense BCH*, nsBCH, que tem sido utilizada em trabalhos recentes) tais que uma sequência, dada como entrada ao algoritmo, é uma palavra-código desses códigos BCH. O algoritmo é usado ao longo do trabalho na identificação de sequências mRNA e proteínas como palavras-código de códigos BCH.

No Capítulo 4, valida-se o modelo tal que o Ribossomo age como um mapa sobrejetor de códons para aminoácidos (modulador) onde as sequências mRNA são palavras-código de um código BCH. Usa-se o algoritmo do Capítulo 3 para encontrar a estrutura matemática de sequências mRNA, no contexto da teoria da informação e codificação, como palavras-código de códigos BCH. Neste capítulo, segue-se a metodologia introduzida em (FARIA, 2011; ROCHA, 2010) e demonstra-se alguns fatos e perguntas em aberto que foram sugeridas nesses trabalhos. Além disso, mostra-se que a metodologia para a identificação, junto com o algoritmo do Capítulo 3, generaliza o algoritmo usado em (FARIA, 2011; ROCHA, 2010; BRANDÃO *et al.*, 2015; FARIA *et al.*, 2012; ROCHA *et al.*, 2010; FARIA *et al.*, 2010) para a identificação de sequências de nucleotídeos (mRNA, DNA, miRNA, etc), portanto a metodologia proposta neste capítulo permite identificar ou representar uma maior quantidade de sequências biológicas que as identificadas nesses trabalhos.

No Capítulo 5, define-se os novos códigos **ECRT** (*Extended Chinese Remainder Theorem*) e demonstra-se todas as propriedades desses códigos: parâmetros do código, uma base ou conjunto de geradores, entre outras. Estes códigos são utilizados para o projeto de códigos cíclicos sobre anéis comutativos com unidade e para representar proteínas funcionais como palavras-código de um codificador genético concatenado.

No Capítulo 6, modela-se o Ribossomo e o processo de síntese de proteínas através de um **codificador genético concatenado** e valida-se o modelo através de casos de estudo o qual verifica a hipótese que as proteínas são macromoléculas que contêm informação e são usadas pela célula para a transmissão de informação.

2 Fundamentos da Síntese de Proteínas e dos Códigos Corretores de Erros

O objetivo deste capítulo é apresentar os principais conceitos utilizados no decorrer deste trabalho. Serão apresentados conceitos relacionados às proteínas, ao ribossomo, às sequências mRNA e aos códigos corretores de erros que serão usados no desenvolvimento deste trabalho.

Nas Seções 2.1 e 2.2 são apresentados, respectivamente, os conceitos básicos das proteínas e do ribossomo, código genético e o processo de transcrição e síntese de proteínas a partir das sequências de mRNA. Estes conceitos podem ser estudados em detalhe em muitos livros de biologia; recomenda-se: (BERG *et al.*, 2003; LODISH, 2008a).

Na Seção 2.3 são introduzidos os conceitos básicos para a construção de códigos lineares, cíclicos e BCH sobre corpos e anéis. Para isto, realiza-se uma revisão dos anéis comutativos com unidade, em especial os anéis $\mathbb{Z}_4$ e $\mathbb{Z}_5$ e o corpo $\mathbb{F}_4$, os quais são utilizados neste trabalho para a construção de códigos BCH; e explica-se o algoritmo rápido para o cômputo da divisão de polinômios, o qual será usado exaustivamente na identificação de sequências mRNA e aminoácidos através de códigos corretores de erros. Os conceitos apresentados nesta seção podem ser estudados com maior detalhe em (PETERSON; WELDON, 1972; LINT, 1999; SHANKAR, 1979; WALKER, 2000; DAHL, 1994).

2.1 Proteínas

Antes de introduzir as proteínas é necessário comentar que a vida começa nas células, nas quais, as proteínas realizam uma grande quantidade de funções cruciais para sua correta operação.

A célula é a unidade básica da vida em todas as formas de organismos vivos, da mais simples bactéria ao mais complexo animal, onde cada célula deve crescer, reproduzir-se, processar informações, responder a estímulos e realizar uma série impressionante de reações químicas. A vida é definida por essas capacidades. Os seres humanos e organismos multicelulares são compostos por bilhões ou trilhões de células organizadas em estruturas complexas, todavia existem outros organismos consistindo de uma única célula. Mesmo os organismos unicelulares simples exibem todas as propriedades características da vida, indicando que a célula é a unidade fundamental, (BERG *et al.*, 2003; LODISH, 2008a).

As células recebem e emitem informações

Uma célula viva monitora continuamente sua vizinhança e ajusta suas atividades e sua

composição de acordo com a necessidade. As células também se comunicam por meio do envio deliberado de sinais que podem ser recebidos e interpretados por outras células. Os sinais utilizados pelas células incluem pequenas moléculas químicas simples, gases, proteínas, luz e movimentos mecânicos. As células têm numerosas proteínas receptoras (para a detecção de sinais) e rotas elaboradas para a transmissão desses sinais em seu interior, para provocar uma resposta. Em um momento determinado, uma célula pode ser capaz de perceber apenas alguns dos sinais que existem ao seu redor e sua resposta pode mudar, dependendo desse momento. Em alguns casos, a recepção de um primeiro sinal indicará à célula o caminho específico a ser seguido em resposta a um sinal diferente subsequente.

As alterações no ambiente (por exemplo, um aumento ou diminuição nos níveis de um nutriente em particular ou nos padrões de luminosidade) quando os sinais enviados por outras células representam informações externas que as células devem processar. As respostas mais rápidas a tais sinais incluem alterações na localização ou na atividade de proteínas preexistentes.

A capacidade das células de emitir e responder a sinais é essencial para o desenvolvimento. Muitos sinais são compostos de proteínas secretadas por células específicas em locais e momentos específicos do organismo em desenvolvimento. Frequentemente, uma célula receptora deve integrar múltiplos sinais para determinar o comportamento a ser seguido, por exemplo, para diferenciar-se em um tecido em particular, dar continuidade a um processo, morrer, enviar um sinal de confirmação ou migrar. A sinalização e a transdução de sinais são atividades primordiais para as células, (BERG *et al.*, 2003; LODISH, 2008a).

Certas moléculas pequenas, monômeros, presentes na célula podem se unir para a formação de polímeros (qualquer molécula grande é composta de monômeros), pela repetição de um único tipo de ligação química (ver Figura 4). As células produzem três tipos de grandes polímeros, denominados macromoléculas: os polissacarídeos, as proteínas e os ácidos nucleicos.

Os ácidos nucleicos são polímeros lineares que contêm de centenas a milhões de nucleotídeos ligados por ligações fosfodiéster. Os polissacarídeos são polímeros lineares ou ramificados de monossacarídeos (açúcares), como a glicose, ligados por meio de ligações glicosídicas.

As proteínas são as macromoléculas mais versáteis nos sistemas vivos e servem para funções cruciais essencialmente em todos os processos biológicos. Funcionam como catalisadores, transportam e armazenam outras moléculas, tais como o oxigênio, fornecem apoio mecânico e proteção imunológica, geram movimento, transmitem impulsos nervosos, e controlam o crescimento e a diferenciação. A seguir apresentam-se algumas propriedades importantes:

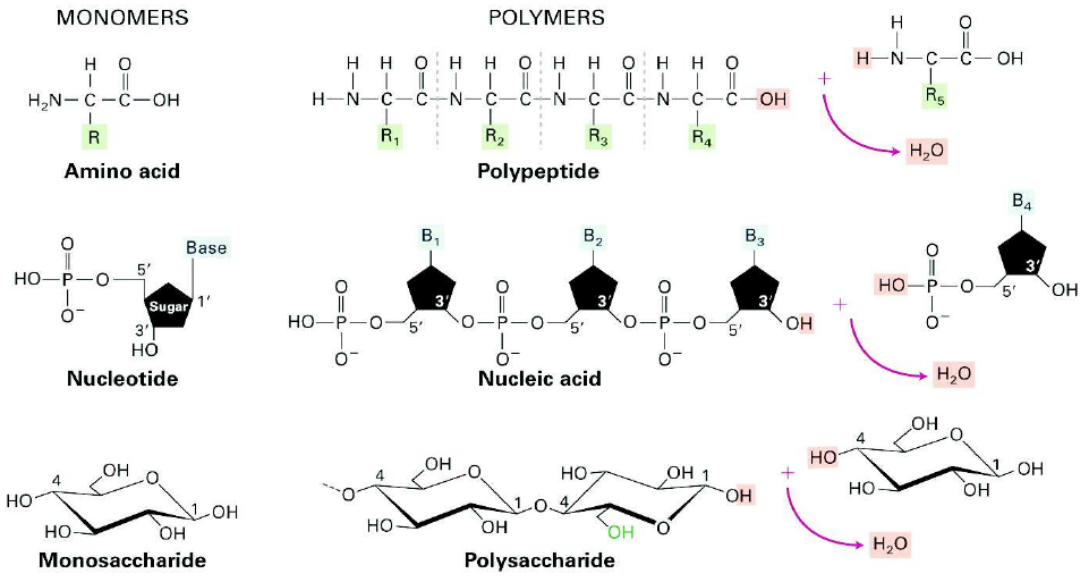


Figura 4 – Blocos químicos que constituem as células e as macroestruturas que formam (LODISH, 2008a).

- As proteínas são polímeros lineares consistindo de unidades monoméricas denominadas aminoácidos que se unem ponta-a-ponta. Elas se enovelam espontaneamente em estruturas tridimensionais que são determinadas pela sequência de aminoácidos. **A função de uma proteína depende diretamente de sua estrutura tridimensional**. Assim, as proteínas são a transição do mundo unidimensional de sequências para o mundo tridimensional de moléculas associadas a diversas atividades.

- As proteínas contêm uma ampla faixa de grupamentos funcionais (alcoóis, tióis, tioéteres, ácidos carboxílicos,...), tal que quando combinados em várias sequências, este conjunto de grupamentos funcionais contempla o amplo espectro de funções das proteínas.

- As proteínas podem interagir umas com as outras, e com outras macromoléculas biológicas, formando montagens complexas. As proteínas dentro destas montagens podem agir em sinergia, gerando capacidades não existentes nas proteínas componentes individuais.

- Algumas proteínas são bem rígidas, enquanto outras apresentam uma considerável flexibilidade. As unidades rígidas podem funcionar como elementos estruturais no citoesqueleto ou no tecido conjuntivo. Proteínas com alguma flexibilidade podem atuar como dobradiças, molas e alavancas, e para a transmissão de informação intra e inter celular (BERG *et al.*, 2003).

- Algumas proteínas apresentam semelhanças entre si, podendo, portanto, ser classificadas em famílias proteicas. Algumas centenas dessas famílias já foram identificadas. A maior parte das proteínas foi projetada para atuar em determinados locais dentro

Aminoácido	**Abreviação**	**Aminoácido**	**Abreviação**
Alanina	A	Isoleucina	I
Arginina	R	Leucina	L
Asparagina	N	Lisina	K
Aspartato (ácido aspártico)	D	Metionina	M
Cisteína	C	Prolina	P
Fenilalanina	F	Serina	S
Glicina	G	Tirosina	Y
Glutamato (ácido glutâmico)	E	Treonina	T
Glutamina (glutamida)	Q	Triptofano	W
Histidina	H	Valina	V

Tabela 2 – Aminoácidos e as abreviações com uma letra

das células ou para ser liberada no espaço extracelular. Vias celulares elaboradas asseguram que as proteínas sejam transportadas para suas localizações intracelulares adequadas ou que sejam secretadas.

- As proteínas podem servir como componentes estruturais para as células, por exemplo, formando um esqueleto interno. Podem atuar como sensores que alteram sua forma quando ocorrem mudanças na temperatura, na concentração iônica ou em outras propriedades celulares. As proteínas podem ainda importar ou exportar substâncias através da membrana plasmática. Elas podem ser enzimas que provocam determinadas reações químicas mais rapidamente. Elas podem se ligar a genes específicos, ativando-os ou desligando-os. Elas podem ser sinalizadores extracelulares, liberados por uma célula para a comunicação com outras células ou sinalizadores intracelulares, transportando informações dentro das células. Elas podem ser motores que movimentam outras moléculas, queimando energia química (ATP).

Uma pergunta que surge: como podem 20 aminoácidos formar todas as diferentes proteínas necessárias para a execução de tantas tarefas diferentes? Isso parece impossível. Porém, se uma proteína "padrão" é composta de 300 aminoácidos, existem 20^{300} possíveis sequências proteicas diferentes. Veja que algumas dessas proteínas são funcionalmente equivalentes, mas muitas são instáveis ou por algum motivo descartáveis. O número possível de proteínas é muito grande e se evidência uma alta taxa de redundância.

Os aminoácidos são as unidades estruturais básicas das proteínas. De fato, todas as proteínas em todas as espécies (bactérias, archaea e eucariontes) são construídas a partir do mesmo conjunto de 20 aminoácidos com apenas algumas exceções. Este alfabeto fundamental para a formação das proteínas tem vários bilhões de anos de idade. Os aminoácidos, junto com sua abreviação, são mostrados na Tabela 2.

2.1.0.1 Estrutura hierárquica das proteínas

A estrutura das proteínas pode ser categorizada em uma série de 4 níveis, interdependentes:

- **Estrutura primária:** É dada pela sequência de aminoácidos ao longo da cadeia polipeptídica e resulta numa longa cadeia de aminoácidos, sem se preocupar com a orientação espacial da molécula. É o nível estrutural mais simples e mais importante, pois dele deriva todo o arranjo espacial da molécula. São únicas para cada proteína. Ver Figura 5.

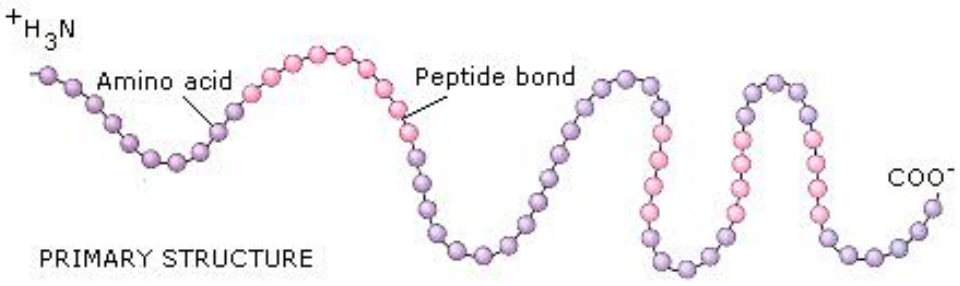


Figura 5 – Estrutura primária da proteína (BERG *et al.*, 2003).

- **Estrutura Secundária:** É dada pelo arranjo espacial de aminoácidos próximos entre si na sequência primária da proteína. O arranjo secundário de uma sequência de aminoácidos pode ocorrer de forma regular. Os dois tipos principais de arranjos secundários regulares são: α-hélice e folha β (existem também volta β e alça Ω). Ver Figura 6.

- **Estrutura Terciária:** Esta estrutura confere a atividade biológica às proteínas e descreve o dobramento final de uma cadeia. Enquanto a estrutura secundária é determinada pelo relacionamento estrutural de curta distância, a terciária é caracterizada pelas interações de longa distância entre aminoácidos. Ver Figura 7.

- **Estrutura Quaternária:** É dada pelo arranjo espacial de subunidades e a natureza de suas interações. Cada cadeia peptídica em tal proteína é chamada subunidade. Ver Figura 8.

2.1.0.2 Estrutura tridimensional das proteínas

O princípio central da biologia molecular estabelece que *a sequência de aminoácidos determina a conformação tridimensional da proteína* (BERG *et al.*, 2003) e a conformação define o funcionamento da proteína. Portanto, o funcionamento da proteína é especificado pela sequência de aminoácidos.

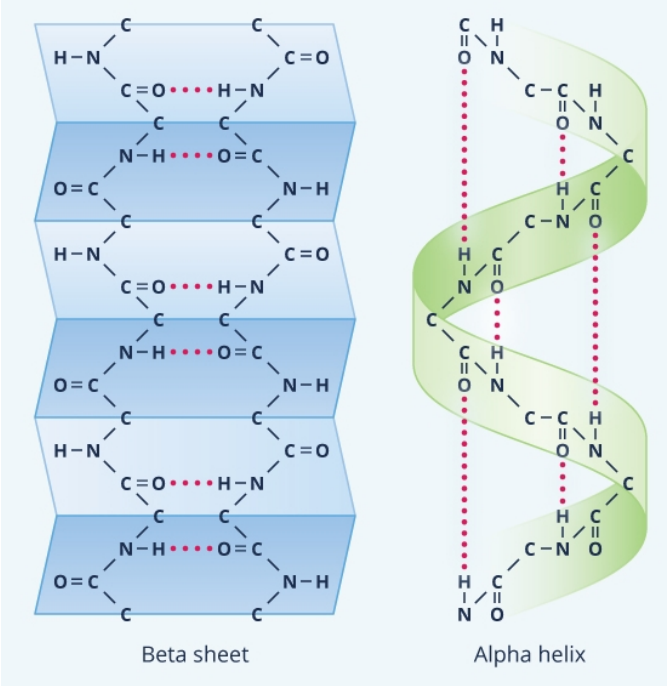


Figura 6 – Estrutura secundária da proteína (BERG *et al.*, 2003).

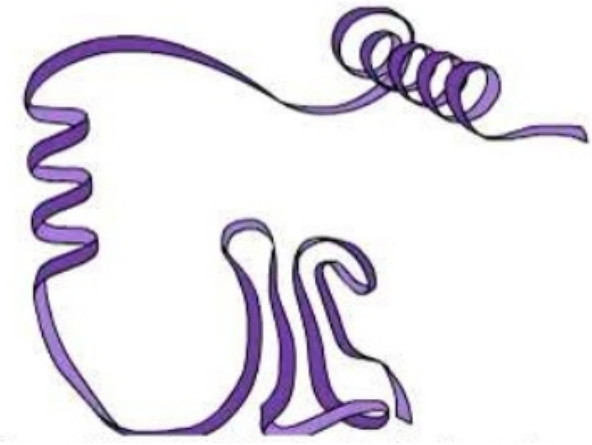


Figura 7 – Estrutura terciária da proteína (BERG *et al.*, 2003).

Em (BERG *et al.*, 2003), mostra-se que regiões locais da sequência de aminoácidos que tenham preferência estrutural significativa, mas não necessariamente estáveis por conta própria, tenderão a adotar suas estruturas favorecidas e, à medida, que se conforma a estrutura final, podem interagir umas com as outras, levando a estabilização crescente.

A célula tem mecanismos de verificação de erros, que eliminam proteínas sintetizadas ou dobradas incorretamente. As proteínas dobradas incorretamente geralmente não possuem atividade biológica e, em alguns casos, podem estar associadas a doenças. O dobramento incorreto das proteínas é evitado por dois mecanismos distintos. Primeiro, as células têm sistemas que reduzem as chances de formar proteínas mal dobradas (complexo de proteínas **chaperoninas**). Segundo, qualquer proteína incorretamente dobrada, bem

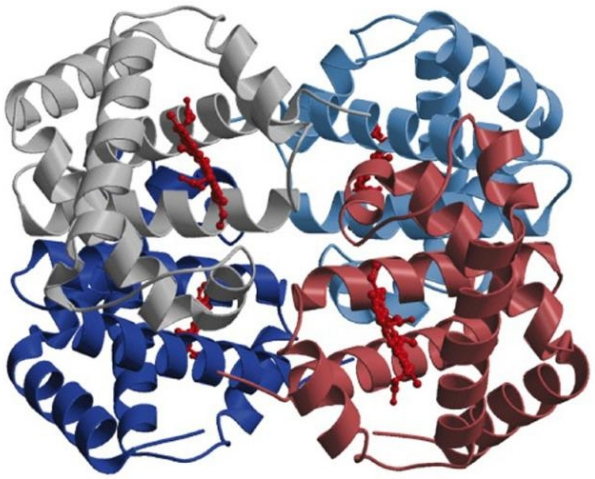


Figura 8 – Estrutura quaternária da proteína (BERG *et al.*, 2003).

como as proteínas citosólicas que não são mais necessárias à célula, são degradadas por um sistema celular especializado de descarte (**chaperonas**).

2.2 O Ribossomo e a Síntese Proteica

O ribossomo é uma máquina molecular composta de duas subunidades e a sua função é a de sintetizar proteínas a partir das sequências RNA mensageiras (tradução), sendo estas, sequências de nucleotídeos ($\{A, C, G, U\}$) transcritas do DNA. O ribossomo consiste de dois componentes principais: a pequena subunidade ribossomal, a qual lê o RNA mensageiro (mRNA), e a grande subunidade, a qual junta os aminoácidos, na ordem definida pelo mRNA, para formar uma cadeia polipeptídica ou proteína (ver Figura 9). Cada subunidade está composta por uma ou mais moléculas RNA ribossômicas (rRNA) e uma variedade de proteínas ribossômicas.

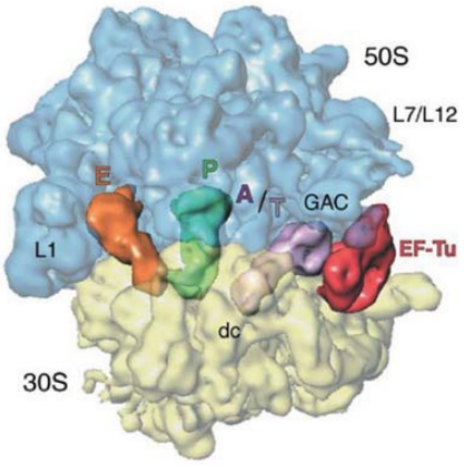


Figura 9 – Estrutura da proteína. A grande subunidade 50S e a pequena subunidade 30S (HASHEM *et al.*, 2013)

A sequência de DNA, que codifica a sequência dos aminoácidos numa proteína, é copiada para uma cadeia de RNA mensageira. Algum ribossomo, presente no citosol, liga-se a cadeia de RNA mensageiro e a utiliza para determinar a sequência correta de aminoácidos. Cada um dos aminoácidos são selecionados, recolhidos e transportados para o ribossomo por moléculas de RNA de transferência (tRNA), que entram numa parte do ribossomo e se ligam à cadeia de mRNA. Durante esta ligação ocorre a tradução correta da sequência de ácido nucleico para a sequência de aminoácidos. Para cada tripleto de codificação (códon) no mRNA existe um tRNA distinto que corresponde e que transporta o aminoácido correto para esse códon. Os aminoácidos são então ligados entre si pela grande unidade ribossomal. Uma vez que a proteína é produzida, esta se dobra para produzir uma estrutura tridimensional específica, embora, durante a síntese, algumas proteínas comecem a se dobrar em sua forma correta.

O ribossomo e o dogma central

A informação genética nos sistemas vivos está armazenada nas sequências genômicas de DNA. Uma grande parte dessas sequências codificam proteínas, as quais efetuam a maioria das tarefas funcionais em todos os organismos existentes. A informação do DNA é disponibilizada pelo processo de transcrição dos genes para sequências mRNA, as quais são traduzidas em proteínas pelo ribossomo. Este é o dogma central (CRICK, 1970) da biologia molecular em sua forma mais simples.

O mapeamentos da sequência mRNA para proteína inicia pelo códon AUG, seguido por uma sequência de códons, a qual especifica a ordem em que os aminoácidos são inseridos na proteína em formação, e a qual é seguida por um códon de finalização, indicando que a proteína está pronta para se separar do ribossomo e para, consequentemente, se enovelar e adquirir seu estado funcional. A ligação entre o mRNA e a sequência polipeptídica é o tRNA. Na célula bacteriana existem cerca de 50 tipos diferentes de moléculas tRNA, cada uma composta de 75 nucleotídeos. A sequência tRNA possui um terminal *CCA-end*, na qual um aminoácido pode estar ligado por uma ligação éster, e um anticódon que pode ser lido pelo mRNA. Para cada aminoácido existe uma enzima que reconhece o tRNA como anticódon complementar do códon mRNA (ver Figura 10).

A estrutura do ribossomo

Um ribossomo é composto por dois tipos de moléculas: RNAs e proteínas, e é, portanto, uma ribonucleoproteína (ver Figura 9). Cada ribossomo é dividido em duas subunidades: 1. uma subunidade menor que se liga a uma subunidade maior e o padrão de mRNA, e 2. uma subunidade maior que se liga ao tRNA, aos aminoácidos e à subunidade menor. Quando um ribossomo termina a leitura de uma molécula de mRNA, estas duas subunidades se separam. Os ribossomos são ribozimas, porque a atividade da peptidil transferase catalítica que liga os aminoácidos em conjunto é realizada pelo DNA ribossômico. Os ribossomos são frequentemente associados às membranas intracelulares

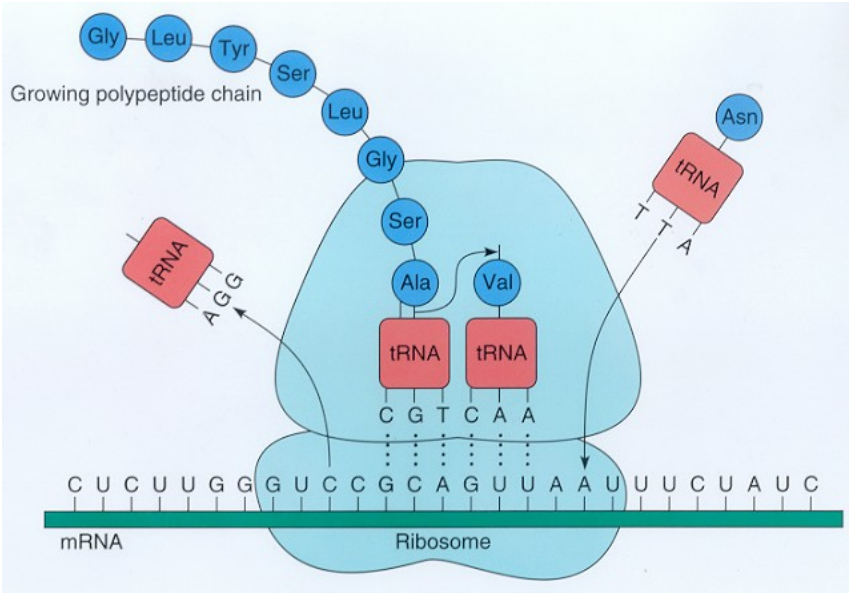


Figura 10 – Funcionamento do ribossomo e o processo da síntese de proteínas (NELSON *et al.*, 2008)

que compõem o retículo endoplasmático rugoso.

Ribossomos de bactérias, archaea e eucariotas se assemelham uns aos outros em um grau notável, evidência-se uma origem comum. Eles diferem em seu tamanho, sequência, estrutura, e a proporção de proteína para RNA. As diferenças na estrutura permitem que alguns antibióticos matem bactérias inibindo seus ribossomos, e deixando os ribossomos humanos inalterados. Os ribossomos mitocondriais das células eucarióticas, são produzidos a partir de genes mitocondriais, e funcionalmente se assemelham a muitas características daqueles em bactérias, refletindo a provável evolução da origem das mitocôndrias.

A estrutura dos ácidos nucleicos

Os ácidos nucleicos são macromoléculas que armazenam a informação genética e determinam a sequência de aminoácidos e, consequentemente, a estrutura e a função das proteínas de uma célula.

Os ácidos nucleicos contêm as informações para a produção das proteínas no local e momento adequados. A informação referente está contida no material genético. Principalmente há dois tipos de ácidos nucleicos quimicamente semelhantes e carreadores da informação das células, o DNA (ácido desoxirribonucleico) e o RNA (ácido ribonucleico). Os monômeros que formam o DNA e o RNA, denominados nucleotídeos, têm uma estrutura comum: um grupo fosfato ligado por uma ligação fosfodiéster a uma pentose (uma molécula de açúcar com cinco carbonos) que, por sua vez, está ligada a um anel cuja estrutura contém nitrogênio e carbono, a qual normalmente é conhecida como “base”. No RNA, a pentose é a ribose, e no DNA, é a desoxirribose.

As bases adenina, guanina e citosina são encontradas tanto no DNA como no

RNA. A timina é encontrada apenas no DNA e a uracila, apenas no RNA. A adenina e a guanina são purinas, contêm um par de anéis fusionados. A citosina, a timina e a uracila são pirimidinas (Figura 11). As bases são abreviadas por A, G, C, T e U, respectivamente.

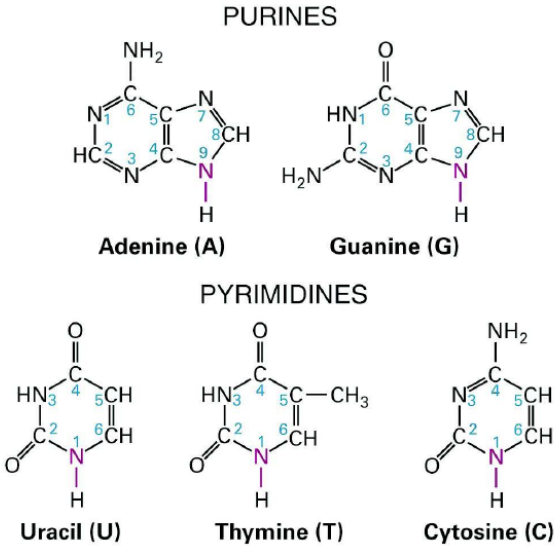


Figura 11 – Estrutura química das bases dos ácidos nucleicos (LODISH, 2008a).

O modelo do DNA introduzido em 1953 por Watson e Crick possui as seguintes características:

- Duas cadeias de ácidos nucleicos que circundam um eixo e formam a dupla hélice.
- As duas fitas de DNA são antiparalelas (bases complementares e sentido oposto) e cada uma forma uma hélice para o lado direito.
- As bases se localizam no centro da hélice, e as cadeias de fosfato estão na periferia, minimizando a repulsão entre os grupos fosfato carregados.
- Cada base está ligada a uma base complementar da fita oposta por meio de pontes de hidrogênio, onde unicamente se acomodam dois tipos de bases. A adenina deve formar o par com a timina e vice-versa, e cada guanina deve formar par com a citosina e vice-versa.

A síntese proteica

As células executam dois processos em série para converter a informação codificada no DNA (o qual contêm a informação completa da célula) em proteínas. O primeiro processo é denominado **transcrição**, a região codificante de um gene é copiada sob a forma de uma versão em fita simples de **ácido ribonucleico (RNA)** a partir da dupla fita de **DNA**. A enzima **RNA polimerase** catalisa a ligação dos nucleotídeos na cadeia de **RNA**, usando o **DNA** como molde. Em células eucarióticas, o **RNA** precursor é

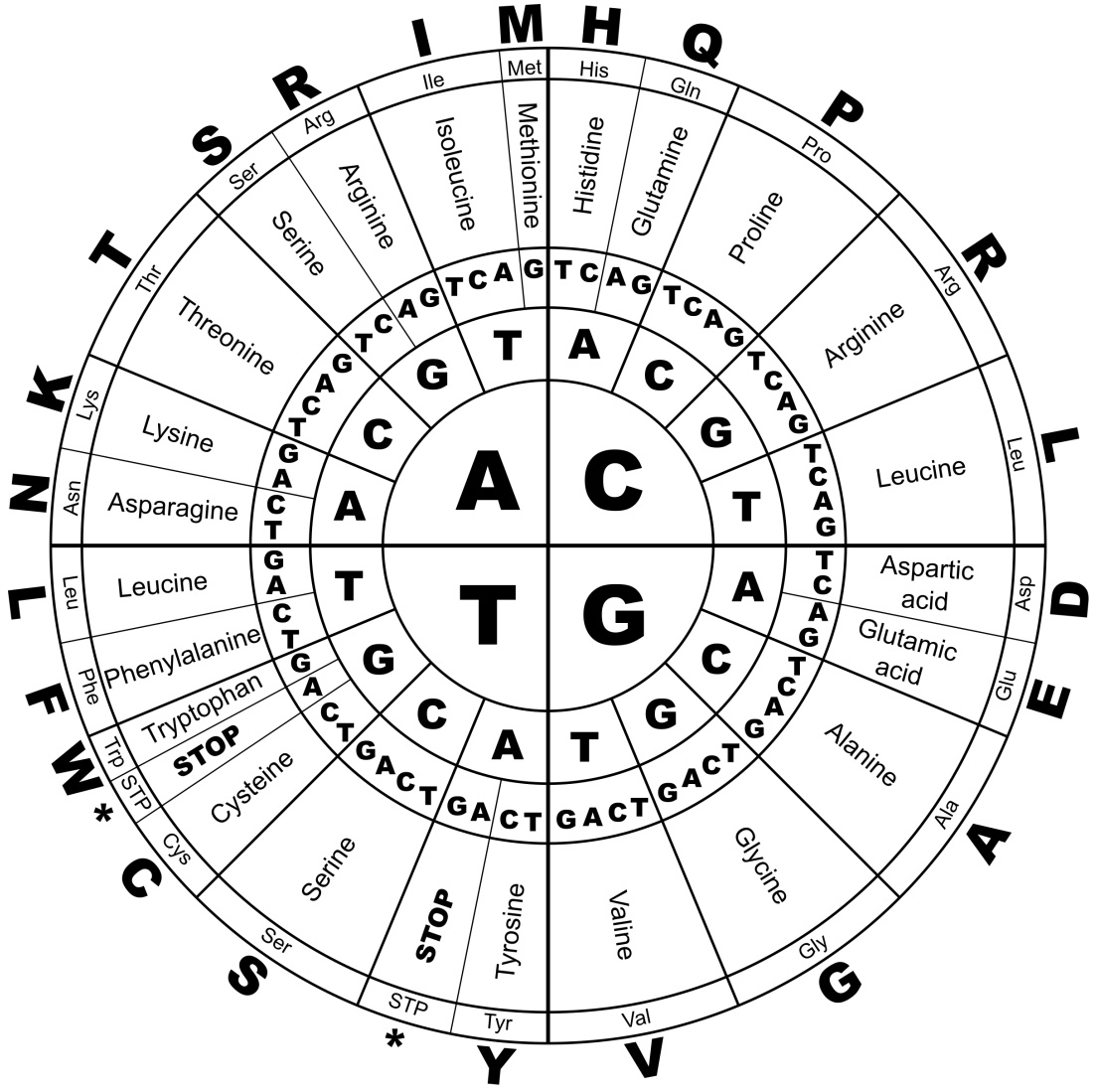


Figura 12 – Código Genético. Mapa sobrejetor de códons (tripletas ordenadas nucleotídeos) para aminoácidos.

processado em uma molécula de **RNA mensageiro (mRNA)** menor, sendo esta transportada para o citoplasma. Neste compartimento, o **ribossomo** se encarrega de efetuar o segundo processo, denominado **tradução**. Durante a tradução o ribossomo, organiza e liga os aminoácidos seguindo uma ordem estabelecida, a qual é ditada pela sequência mRNA, de acordo com o **"código genético"**, praticamente universal.

O **"código genético"** é mostrado na Figura 12 e observa-se que ele é, simplesmente, um mapa sobrejetor de códons para aminoácidos. O conjunto de nucleotídeos, alfabeto das sequências de mRNA, é denotado por $\mathbb{N} = \{A, C, G, U\}$ e o conjunto de aminoácidos, alfabeto das proteínas, denotado por $\mathbb{A} = \{A, R, N, \ldots, T, W, V\}$ (Tabela 2), então, o "código genético" pode ser definido como o mapa:
Código Genético : $(\mathbb{N} \times \mathbb{N} \times \mathbb{N}) \to \mathbb{A}$

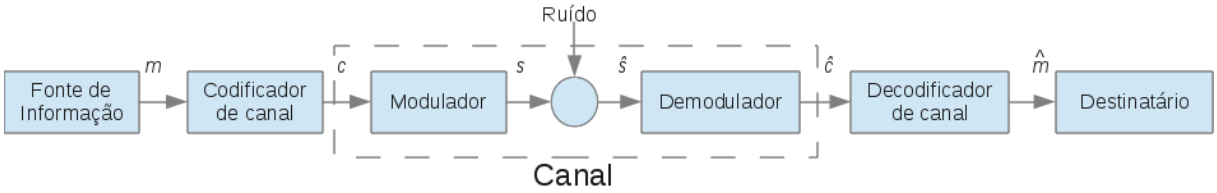


Figura 13 – Sistema de comunicação tradicional, sem considerar codificador de fonte.

2.3 Códigos Corretores de Erros

A teoria da codificação utiliza diferentes conceitos matemáticos tais como: teoria dos números, teoria dos grupos, combinatória, geometrias finitas e geometria algébrica, dentre outras. Os Códigos Corretores de Erros (CCE) são utilizados sempre que se deseja transmitir ou armazenar informação. Por exemplo, nas comunicações via satélite, nas comunicações internas de um computador, no armazenamento de dados em CD e DVD ou armazenamento óptico de dados. A teoria da codificação iniciou com os trabalhos de Golay (GOLAY, 1949), Hamming (HAMMING, 1950) e Shannon (SHANNON, 1948). O resultado mais importante surgiu, principalmente devido a Shannon, que afirmou e provou que para um determinado canal de comunicação e para taxas de transmissão menores do que a capacidade de canal, existe um código que permite uma transmissão confiável com probabilidade de erro arbitrariamente pequena.

O objetivo de um sistema de comunicação é transmitir, confiavelmente, a informação de uma fonte para um destinatário através de um canal de comunicação susceptível a ocorrência de erros. Os blocos que conformam um sistema de comunicação são mostrados na Figura 13. O codificador de canal recebe da fonte de informação a mensagem que se deseja transmitir (m) e mapeia-a numa única palavra-código c. O modulador representa a palavra-código como um sinal no espaço de sinais (s) e a transmite pelo canal de transmissão, o qual é susceptível a ocorrência de erros. O sistema de recepção realiza o processamento inverso; primeiro, o sinal recebido ($\hat{s}$) é demodulado e convertido numa sequência ($\hat{c}$); por último, o decodificador de canal decodifica $\hat{c}$ como a palavra-código mais próxima e envia para o destinatário a mensagem $\hat{m}$ que lhe corresponde. O sistema de comunicação terá enviado corretamente a informação quando $m = \hat{m}$.

O objetivo desta seção é apresentar alguns dos conceitos básicos de álgebra e dos códigos corretores de erros que são fundamentais para o entendimento do presente trabalho. Na Seção 2.3.1 mostram-se as definições e propriedades das estruturas de grupo, anel e corpo, as quais são fundamentais para a construção de códigos corretores de erro. Na Seção 2.3.2 apresentam-se os conceitos relacionados com os códigos lineares e suas principais características. Na Seção 2.3.3, as propriedades dos códigos cíclicos BCH são introduzidas e serão os principais elementos ferramentas para a modelagem da síntese de proteínas fundamentada na teoria de comunicação.

2.3.1 Estruturas algébricas e suas propriedades

Como dito anteriormente, um código corretor de erros (CCE) é um sistema que acrescenta redundância à mensagem que se deseja transmitir ou armazenar, e permite que o destinatário recupere a mensagem e detecte ou corrija erros a partir da sequência recebida ou lida. Assim, podemos definir:

Definição 2.1. *Um **código de bloco linear** $\mathcal{C}$ é um subespaço vetorial de $\mathbb{A}^n$ (ou sub-módulo), onde $\mathbb{A}$ é chamado **alfabeto** do código e n é o **comprimento** das sequências. Chamam-se **palavras-código** às sequências, no alfabeto $\mathbb{A}$, que compõem o código $\mathcal{C}$.*

O alfabeto $\mathbb{A}$ do código é um conjunto finito ou infinito de elementos. Como o objetivo dos CCE's é adicionar redundância de um modo organizado, então o alfabeto $\mathbb{A}$ deve satisfazer certas propriedades algébricas que facilitem a codificação das mensagens em palavras-código e a decodificação de palavras-código em mensagens. A seguir apresentam-se as definições das estruturas algébricas que serão usadas no desenvolvimento do trabalho.

2.3.1.1 Grupos

Definição 2.2. *Uma **Operação Binária** "$*$" sobre um conjunto $\mathbb{A}$ é uma regra que associa algum elemento de $\mathbb{A}$ a cada par ordenado (a, b) de elementos de $\mathbb{A}$. $(a*b)$ denotará o elemento associado a (a, b) através de $*$.*

Definição 2.3. *Um **Grupo Abeliano** $(\mathbb{A}, *)$ é um conjunto não vazio $\mathbb{A}$ com uma operação binária $*$ sobre $\mathbb{A}$, tal que, os seguintes axiomas são satisfeitos:*

1. *A operação $*$ é **Fechada**, i.e, $(a * b) \in \mathbb{A}\ \forall a, b \in \mathbb{A}$*
2. *A operação $*$ é **Associativa**, i.e, $(a * b) * c = a * (b * c)\ \forall a, b, c \in \mathbb{A}$*
3. *Existe um **elemento identidade** "e" em $\mathbb{A}$, tal que, $a * e = e * a = a\ \ \forall a \in \mathbb{A}$*
4. *Para todo elemento de $\mathbb{A}$ existe **elemento inverso** com relação a operação $*$ em $\mathbb{A}$, isto é, $\forall a \in \mathbb{A}\ \exists a^{-1}$ tal que $a * a^{-1} = e = a^{-1} * a$*
5. *A operação $*$ é **Comutativa**, i.e, $(a * b) = (b * a)\ \forall a, b \in \mathbb{A}$*

Usando as propriedades acima, pode-se mostrar os seguintes fatos:

- O elemento identidade de um grupo G é único.
- O elemento inverso de um dado elemento é único.

Definição 2.4. *Se G é um grupo finito, então, a ordem de G, $|G|$, é o número de elementos de G.*

Definição 2.5. *Se um subconjunto H de um grupo G é fechado sob a operação binária sobre G e se H é um grupo sob esta operação binária, então H é um subgrupo de G. Escreve-se $H \leq G$.*

Definição 2.6. *Seja G um grupo e seja $a \in G$. Então:*

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

é um subgrupo de G e é o menor subgrupo de G que contém a, ou seja, qualquer outro subgrupo que contém a contém também H.

Definição 2.7. *O grupo H é o **subgrupo cíclico de** G **gerado por** a, e denota-se por $\langle a \rangle$.*

Definição 2.8. *Dados um grupo G e um elemento $a \in G$, se ocorrer que:*

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

*então a é um **gerador** de G e o grupo $G = \langle a \rangle$ é cíclico.*

Definição 2.9. *A ordem n de um elemento a pertencente a um grupo finito G é o menor inteiro positivo tal que $a^n = e$, onde e é a identidade do grupo.*

Definição 2.10. *Considere dois grupos quaisquer G e $\hat{G}$ e a função (ou mapeamento) $\phi : G \to \hat{G}$. Dize-se que ϕ é um **homomorfismo** de G em $\hat{G}$ se:*

$$\phi(ab) = \phi(a)\phi(b)$$

para todo $a, b \in G$. (Note que o produto ab ocorre em G, enquanto que o produto $\phi(a)\phi(b)$ ocorre em $\hat{G}$).

Definição 2.11. *Um **isomorfismo de grupos** de G em $\hat{G}$ é um homomorfismo onde a função $\phi : G \to \hat{G}$ é bijetora. Dize-se que G e $\hat{G}$ são isomorfos e escreve-se: $G \cong \hat{G}$.*

2.3.1.2 Anéis e Corpos

Definição 2.12. *Um **Corpo** $(\mathbb{A}, +, \cdot)$ é um conjunto, denotado por $\mathbb{A}$, com duas operações binárias ("$+$" e "$\cdot$"), tal que os seguintes axiomas são satisfeitos:*

1. *$(\mathbb{A}, +)$ é um grupo abeliano, onde o elemento "0" representa o elemento identidade.*
2. *$(\mathbb{A} \setminus \{0\}, \cdot)$ é um grupo abeliano, onde o elemento "1" representa o elemento identidade.*
3. *A operação "$\cdot$" se distribui sobre "$+$", i.e, $a \cdot (b + c) = (a \cdot b) + (a \cdot c), \ \ \forall a, b, c \in \mathbb{A}$*

Definição 2.13. *Um **Anel associativo comutativo com identidade** $(A, +, \cdot)$ é um conjunto, denotado por $\mathbb{A}$, com duas operações binárias ("+" e "$\cdot$"), tais que,os seguintes axiomas são satisfeitos:*

1. *$(\mathbb{A}, +)$ é um grupo abeliano, onde o elemento "0" representa o elemento identidade do grupo.*
2. *$(\mathbb{A} \setminus \{0\}, \cdot)$ satisfaz as mesmas propriedades do grupo abeliano, **exceto** a existência de elemento inverso. O elemento "1" representa o elemento identidade.*
3. *A operação "$\cdot$" se distribui sobre "+", i.e, $a \cdot (b + c) = (a \cdot b) + (a \cdot c), \ \ \forall a, b, c \in \mathbb{A}$*

Observe-se que a diferença entre o Corpo e o Anel associativo comutativo é, que neste ultimo, existem elementos que **não** apresentam o seu inverso multiplicativo; estes elementos serão denominados **Divisores de Zero**, pelo seguinte fato: Para um elemento $a \in A$ (A é um anel Anel associativo comutativo), se **não** existe a^{-1} tal que $a^{-1} \cdot a = 1$, então, existe $b \neq 0 \in A$ tal que $a \cdot b = 0$.

Exemplo 2.1. *O conjunto $A = \mathbb{Z}_4 = \{0, 1, 2, 3\}$ com as operações binárias: soma módulo 4 ($\oplus_4$) e produto módulo 4 ($\otimes_4$) forma um Anel associativo comutativo.*

As operações modulo 4 podem ser feitas usando as operações dos números inteiros da seguinte maneira: $(a \oplus_4 b)$ é o resto da divisão por 4 da soma $a + b$, i.e, igual a $(a + b) \pmod 4$; e $(a \otimes_4 b)$ é o resto da divisão por 4 da multiplicação $a \cdot b$, i.e, igual a $(a \cdot b) \pmod 4$.

As Tabelas de Cayley apresentam o resultado de todas as possíveis operações que podem ser feitas num conjunto usando uma operação binária. As Tabelas 3a) e 3b) são as tabelas de Cayley para $\mathbb{Z}_4$ e as operações $\oplus_4$ e $\otimes_4$. Usam-se para verificar que o anel $(\mathbb{Z}_4, \oplus_4, \otimes_4)$ é um anel associativo comutativo. Das tabelas, pode-se ver que: o elemento "0" é a identidade do grupo $(\mathbb{Z}_4, \oplus_4)$, pois $0 \oplus_4 0 = 0$, $1 \oplus_4 0 = 1$,...; o elemento "1" é a identidade para a operação $\otimes_4$ ($1 \otimes_4 1 = 1$, $1 \otimes_4 2 = 2$, ...); e o único divisor de zero é 2 ($2 \otimes_4 2 = 0$).

Tabela 3 – Tabelas de Cayley do anel $(\mathbb{Z}_4, \oplus_4, \otimes_4)$.

(a) Tabela de Cayley do operador $\oplus_4$

$\oplus_4$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

(b) Tabela de Cayley do operador $\otimes_4$

$\otimes_4$	1	2	3
1	1	2	3
2	2	0	2
3	3	2	1

Exemplo 2.2. *O conjunto $A = \mathbb{Z}_5$ com as operações binárias: soma módulo 5 ($\oplus_5$) e produto módulo 5 ($\otimes_5$) formam a estrutura algébrica de corpo*

De maneira equivalente ao exemplo anterior, as operações são definidas usando a aritmética dos números inteiros: $(a \oplus_5 b) = (a + b) \pmod 5$ e $(a \otimes_5 b) = (a \cdot b) \pmod 5$.

As Tabelas 4a) e 4b) são as tabelas de Cayley para $\mathbb{Z}_5$ e as operações $\oplus_5$ e $\otimes_5$. Usam-se para verificar que o anel $(\mathbb{Z}_5, \oplus_5, \otimes_5)$ é um Corpo. Das tabelas, pode-se ver que: o elemento "0" é a identidade do grupo $(\mathbb{Z}_5, \oplus_5)$, pois $0 \oplus_5 0 = 0$, $1 \oplus_5 0 = 1$,...; o elemento "1" é a identidade para a operação $\otimes_5$ ($1 \otimes_5 1 = 1$, $1 \otimes_5 2 = 2, ...$); e todo elemento não nulo de $\mathbb{Z}_5$ possui um inverso multiplicativo ($2 \otimes_5 3 = 1$ e $4 \otimes_5 4 = 1$). Denota-se $\mathbb{Z}_5$ como $\mathbb{F}_5$ por ser um corpo.

Tabela 4 – Tabelas de Cayley do anel $(\mathbb{Z}_5, \oplus_5, \otimes_5)$.

(a) Tabela de Cayley do operador $\oplus_5$

$\oplus_5$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

(b) Tabela de Cayley do operador $\otimes_5$

$\otimes_5$	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

Exemplo 2.3. *O conjunto $A = \mathbb{F}_4 = \{0, 1, \alpha, 1 + \alpha\}$ com as operações binárias especificadas pelas Tabelas 5 formam um corpo.*

Tabela 5 – Tabelas de Cayley do corpo $(\mathbb{F}_4, \oplus_{\mathbb{F}_4}, \otimes_{\mathbb{F}_4})$.

(a) Tabela de Cayley do operador $\oplus_{\mathbb{F}_4}$

$+$	0	1	α	$1+\alpha$
0	0	1	α	$1+\alpha$
1	1	0	$1+\alpha$	α
α	α	$1+\alpha$	0	1
$1+\alpha$	$1+\alpha$	α	1	0

(b) Tabela de Cayley do operador $\otimes_{\mathbb{F}_4}$

$\cdot$	1	α	$1+\alpha$
1	1	α	$1+\alpha$
α	α	$1+\alpha$	1
$1+\alpha$	$1+\alpha$	1	α

As Tabelas 5a) e b) são as tabelas de Cayley para $A = \mathbb{F}_4$ e as operações $\oplus_{\mathbb{F}_4}$ e $\otimes_{\mathbb{F}_4}$. Usa-se para verificar que $(\mathbb{F}_4, \oplus_{\mathbb{F}_4}, \otimes_{\mathbb{F}_4})$ é um corpo. Das tabelas, pode-se ver que: o elemento "0" é a identidade do grupo $(\mathbb{F}_4, \oplus_{\mathbb{F}_4})$, pois $0 \oplus_{\mathbb{F}_4} 0 = 0$, $1 \oplus_{\mathbb{F}_4} 0 = 1$,...; o elemento "1" é a identidade para a operação $\cdot$ ($1 \times_{\mathbb{F}_4} 1 = 1$, $1 \otimes_{\mathbb{F}_4} \alpha = \alpha, ...$); e todo elemento não nulo de $\mathbb{F}_4$ possui um inverso multiplicativo, uma vez que $\alpha \otimes_{\mathbb{F}_4} (1 \oplus_{\mathbb{F}_4} \alpha) = 1$.

Definição 2.14. *Dize-se que Q é um subanel de um anel R se $Q \subseteq R$ e Q também forma um anel sob as operações $+$ e $\cdot$, herdadas de R.*

Definição 2.15. *Um subanel Q de um anel R é um ideal a direita (ou a esquerda) em R se $Qb \subseteq Q$ ($bQ \subseteq Q$) para todo $b \in R$. Se Q é simultaneamente um ideal à direita e à esquerda em R, dizemos que Q é um ideal em R.*

Definição 2.16. *Sejam R e $\hat{R}$ anéis. Uma função (mapeamento) $\phi : R \to \hat{R}$ é um* ***homomorfismo*** *se as condições abaixo são satisfeitas, para $a, b \in R$:*

- $\phi(a + b) = \phi(a) + \phi(b)$
- $\phi(ab) = \phi(a)\phi(b)$

Definição 2.17. *Um* ***isomorfismo*** *de R e $\hat{R}$ é um homomorfismo $\phi : R \to \hat{R}$ bijetor. Diz-se R e $\hat{R}$ são isomorfos.*

Definição 2.18. *Seja R um anel. Um R-módulo consiste de um grupo abeliano G e uma operação de multiplicação de cada elemento de G por todo elemento de R, tais que para todo $\alpha, \beta \in G$ e $r, s \in R$, as seguintes condições são satisfeitas:*

- $(r\alpha) \in G$
- $r(\alpha + \beta) = r\alpha + r\beta$
- $(r + s)\alpha = r\alpha + r\beta$
- $(rs)\alpha = r(s\alpha)$

Definição 2.19. *Um* ***subcorpo*** *é um subconjunto de um corpo que tem estrutura de corpo sob as operações herdadas do mesmo.*

Usam-se os corpos finitos na maioria das construções dos códigos corretores de erros conhecidos, estes corpos são também conhecidos como corpos algébricos de Galois ou corpos de Galois e são denotados por $GF(q)$ ou $\mathbb{F}_q$ onde $q \geq 2$ é o número de elementos do corpo. Descrevem-se a seguir um conjunto de propriedades de $\mathbb{F}_q$.

Definição 2.20. *Um* ***polinômio*** *de grau $n-1$ sobre um corpo $\mathbb{F}_q$ é escrito como:*

$$p(x) = p_{n-1}x^{n-1} + p_{n-2}x^{n-2} + \cdots + p_1x + p_0$$

onde x é uma variável e os coeficientes p_i, $0 \leq i \leq n-1$, $i \in \mathbb{Z}$, são elementos de $\mathbb{F}_q$.

Definição 2.21. *Um* ***polinômio mônico*** *é aquele cujo coeficiente líder (coeficiente da variável de maior expoente) p_{n-1} é igual a 1, a identidade multiplicativa de $\mathbb{F}_q$.*

O conjunto de todos os polinômios sobre $GF(q)$ forma um anel sob as operações usuais de soma e multiplicação de polinômios. Este anel é denotado por $GF(q)[x]$ ou $\mathbb{F}_q[x]$.

Definição 2.22. *Um elemento $\beta \in \mathbb{F}_q$ é uma raiz ou zero do polinômio $p(x) \in \mathbb{F}_q[x]$ se $p(\beta) = 0$.*

Teorema 2.1. *Se G é um subgrupo multiplicativo do grupo $(\mathbb{F}^*, \cdot)$ de elementos não nulos de um corpo $\mathbb{F}$, então G é cíclico.*

Corolário 1. *O grupo multiplicativo de todos elementos não nulos de um corpo finito sob a operação multiplicação deste corpo é cíclico.*

Definição 2.23. *Um **polinômio redutível** $q(x) \in \mathbb{F}_q[x]$ é um polinômio que pode ser fatorado por polinômios $a(x)$ e $b(x)$ em $\mathbb{F}_q[x]$ de grau maior ou igual que 1, i.e. $q(x) = a(x)b(x)$*

Um polinômio é dito **irredutível** se este não é um polinômio redutível.

Corolário 2. *Uma extensão (corpo de extensão) E de grau r de um corpo finito $\mathbb{F}_q$ é o conjunto dos polinômios sobre $\mathbb{F}_q$ módulo um polinômio irredutível de grau r.*

Teorema 2.2. *Considere uma extensão finita de grau r sobre o corpo $\mathbb{F}_q$. Então esta extensão tem q^r elementos.*

Definição 2.24. *Diz-se que um polinômio $p(x)$ sobre $\mathbb{F}_q$ é **primo** se ele for mônico e irredutível sobre $\mathbb{F}_q$.*

Teorema 2.3. *O anel de polinômios módulo um polinômio $p(x)$ sobre $\mathbb{F}_q$ é um corpo se, e somente se, $p(x)$ é um polinômio primo.*

Definição 2.25. *Um gerador do grupo multiplicativo de $\mathbb{F}_q$ é denominado um **elemento primitivo** de $\mathbb{F}_q$.*

Corolário 3. *Todo corpo finito $\mathbb{F}$ contém um elemento primitivo.*

Uma consequência imediata do Corolário anterior é que todo corpo de Galois contém um elemento β, tal que todo elemento pertencente ao grupo multiplicativo do corpo finito pode ser expresso como uma potência de β.

Os seguintes teoremas mostram a existência e unicidade dos polinômios minimais.

Definição 2.26. *Seja $GF(q_2)$ um corpo finito e $GF(q_1)$ um subcorpo de $GF(q_2)$. Seja $\beta \in GF(q_2)$. O polinômio primo $p(x)$ de menor grau sobre $GF(q_1)$, tal que $p(\beta) = 0$, é chamado **polinômio minimal** (ou **polinômio primitivo**) de β sobre $GF(q_1)$.*

Teorema 2.4. *Considere os corpos $GF(q_2)$ e $GF(q_1)$ como definidos acima. Cada elemento β de $GF(q_2)$ tem um único polinômio minimal sobre $GF(q_1)$. Mais do que isso, se β tem $p(x)$ como seu polinômio minimal e um polinômio $g(x)$ tem β como um zero, então $p(x)$ divide $g(x)$.*

Os códigos corretores de erros lineares podem ser classificados em duas classes: cíclicos e não-cíclicos. Na Seção 2.3.2 são apresentadas as definições e teoremas dos códigos de bloco lineares e na Seção 2.3.3 são introduzidos os códigos cíclicos BCH.

2.3.2 Códigos de bloco lineares

Um código de bloco sobre o alfabeto $\mathbb{A}$ é especificado por três parâmetros principais: a dimensão, a taxa e a distância mínima de Hamming.

Definição 2.27. *A **dimensão** de um código $\mathcal{C}$ é dada por $k = \log_{|\mathbb{A}|} |\mathcal{C}|$ símbolos por bloco, onde a operação $|\cdot|$ representa a cardinalidade do conjunto.*

Definição 2.28. *A **taxa** de um código $\mathcal{C}$ é dada por $r = k/n$.*

Definição 2.29. *A **distância de Hamming** $d_H(x, y)$ entre duas palavras x e $y \in A^n$ é igual ao número de componentes nas quais elas diferem. Pode-se verificar facilmente que as três propriedades da métrica são satisfeitas, o que permite usar o termo **distância**:*

1. $d_H(x, y) \geq 0$ *e* $d_H(x, y) = 0$ *se, e somente se,* $x = y$;
2. $d_H(x, y) = d_H(y, x)$;
3. $d_H(x, y) + d_H(y, z) \geq d_H(x, z)$.

Dado que se pode computar a distância entre quaisquer duas palavras em $\mathbb{A}^n$, então, pode-se computar também a distância entre quaisquer duas palavras-código. Assim, podemos definir a menor distância entre duas palavras-código que pertencem a um código de bloco.

Definição 2.30. *Seja $\mathcal{C}$ um código de bloco de comprimento n tal que $|\mathcal{C}| \geq 2$. A **Distância Mínima de Hamming de** $\mathcal{C}$, denotada por $d(\mathcal{C})$, é dada por:*

$$d(\mathcal{C}) = \min_{x,y \in \mathcal{C}, x \neq y} d_H(x, y)$$

Os parâmetros de um código de bloco $\mathcal{C}$ de comprimento n, dimensão k e distância mínima de Hamming $d = d(\mathcal{C})$ são representados por (n, k, d)-**código**.

Os códigos de bloco podem ser usados como códigos corretores de erros. A **capacidade de correção de erros** de um código (n, k, d_{min}), denotada por t, está relacionada à distância mínima do código da seguinte forma:

$$d(\mathcal{C}) \leq 2t + 1$$

Portanto, quanto maior a distância mínima do código, maior é a capacidade de correção de erros do código de bloco $\mathcal{C}$. Assim, um "Bom" código corretor de erros é

aquele que possui a maior dimensão k e a maior distância mínima de Hamming possível. O seguinte teorema (limitante de Singleton) estabelece uma relação entre a dimensão, o comprimento e a distância mínima do código, e mostra o compromisso existente entre a dimensão e a distância mínima.

Teorema 2.5. *Um código de bloco $\mathcal{C} \subseteq A^n$ com parâmetros (n, k, d) satisfaz:*

$$|\mathcal{C}| \leq |\mathbb{A}|^{n-d+1}$$

O fato de um código de bloco linear ser um subespaço vetorial, justifica que as palavras-código sejam consideradas como vetores-código com a propriedade que o resultado da soma componente a componente de dois vetores-código será um vetor-código ($\overrightarrow{c_1} + \overrightarrow{c_2} = \overrightarrow{c_3}$) e que a multiplicação por escalar também será um outro vetor código ($\alpha\ \overrightarrow{c_1} = \overrightarrow{c_2}$). Como consequência o vetor todo zero é um vetor-código e a combinação linear de vetores-código será um vetor-código e existe uma **matriz geradora** (G) na qual as linhas dessa matriz formam uma base do código linear $\mathcal{C}$.

$$G = \begin{bmatrix} g_{11} & g_{12} & g_{13} & \cdots & g_{1n} \\ g_{21} & g_{22} & g_{23} & \cdots & g_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & g_{k3} & \cdots & g_{kn} \end{bmatrix}$$

Dado que toda palavra-código pode ser formada como uma combinação linear dos k vetores linha da matriz G, então, o processo de codificação de canal pode ser reescrito como:

$$\overrightarrow{c} = \overrightarrow{u}G$$

onde $\overrightarrow{u}$ representa a mensagem que se deseja transmitir e $\overrightarrow{c}$ representa a palavra-código.

Para toda palavra-código $\overrightarrow{c}$ vale a relação:

$$\overrightarrow{c}H^T = \overrightarrow{0}$$

onde a matriz $(n-k) \times n$, denotada por H, é chamada **matriz verificação de paridade de $\mathcal{C}$**. Pode-se provar que $\overrightarrow{c}H^T = \overrightarrow{0}$ se, e somente se, $\overrightarrow{c} \in \mathcal{C}$; e que a matriz H é uma matriz geradora do chamado **Código Dual** (código de bloco linear), denotado por $\mathcal{C}^{\perp}$.

A matriz verificação de paridade pode ser obtida a partir da matriz geradora quando ela se encontra na forma sistemática. Se o código $\mathcal{C}$ é o espaço linha da matriz $G = [I_k|P]$, então o código dual $C^{\perp}$ é o espaço linha da matriz $H = [-P^T|I_{n-k}]$, onde I_k denota a matriz identidade de ordem k.

Quando o código é linear, o cálculo da distância mínima é simplificado por causa do seguinte fato:

$$d(\mathcal{C}) = \min_{x-y\in\mathcal{C}, x\neq y} d_H(x-y, \overrightarrow{0})$$
$$d(\mathcal{C}) = \min_{c\neq\overrightarrow{0}\in\mathcal{C}} d_H(c, \overrightarrow{0})$$

Onde a propriedade dos códigos lineares: $x - y \in C,\ \forall x, y \in C$, justifica o procedimento. Observa-se que $d_H(c, \overrightarrow{0})$ conta a quantidade de coordenadas de c que são diferentes de zero, o que se denomina como o **peso da palavra** c e se denota como $w(c) = d_H(c, \overrightarrow{0})$. Portanto, a distância mínima do código é $d(\mathcal{C}) = \min_{c\neq\overrightarrow{0}\in\mathcal{C}} w(c)$.

Exemplo 2.4. *O código de repetição é projetado sobre o corpo* $(\mathbb{F}_2 = \{0,1\}, \oplus_2, \otimes_2)$ *e possui os seguintes parâmetros* $(3,1,3)$.

Este código corretor de erros envia a palavra-código $\{000\}$ ou $\{111\}$ quando a mensagem a transmitir é 0 ou 1, respectivamente. Ou seja, o código é definido como: $\mathcal{C} = \{(0,0,0),(1,1,1)\} \in {\mathbb{F}_2}^3$, onde ${\mathbb{F}_2}^3 = \{(0,0,0),(0,0,1),(0,1,0),(0,1,1),(1,0,0),(1,0,1),(1,1,0),(1,1,1)\}$; e é gerado pela matriz:

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

Observa-se que o comprimento do código de bloco é $n = 3$ e que a dimensão é $k = \log_2 2 = 1$, devido a que $|\mathbb{F}_2| = 2$ e $|\mathcal{C}| = 2$, a distância mínima de $\mathcal{C}$ é $d = d(\mathcal{C}) = 3$ porque $w((1,1,1)) = 3$; portanto o código corrige $t = \lfloor \frac{3-1}{2} \rfloor = 1$ erro. A Figura 14 representa o código $\mathcal{C}$ no cubo de $n = 3$ dimensões. Cada um dos vértices do cubo representa uma sequência ou elemento de ${\mathbb{F}_2}^3$ e os vértices $(0,0,0)$ e $(1,1,1)$ representam as duas palavras-código de $\mathcal{C}$. Observa-se que o conjunto total de palavras $({\mathbb{F}_2}^3)$ foi particionado em dois subconjuntos ("nuvens"), os quais correspondem às duas palavras-código. O primeiro subconjunto é formado pelas palavras que se encontram, com respeito à palavra-código $(0,0,0)$, a uma distância menor ou igual a $t = 1$ (capacidade de correção do código), isto é, $S_0 = \{x \in \mathcal{C} \mid d_H(x,(0,0,0)) \leq t\} = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$; e o segundo subconjunto é formado como:

$$S_1 = \{x \in C \mid d_H(x,(1,1,1)) \leq t\} = \{(1,1,1),(0,1,1),(1,0,1),(1,1,0)\}.$$

Assim, o decodificador de canal decidirá pela mensagem 0 (ou 1) se, e somente se, a palavra recebida estiver no subconjunto S_0 (ou S_1).

2.3.3 Códigos cíclicos e BCH

Os códigos BCH pertencem à classe de códigos cíclicos e sua principal importância é a sua simplicidade de geração e decodificação. Isto torna estes códigos potenciais candidatos a serem utilizados em aplicações práticas.

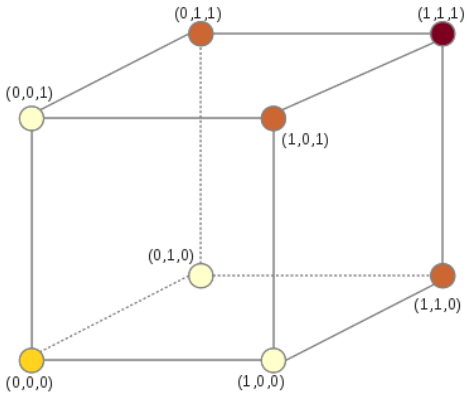


Figura 14 – Representação gráfica do código de repetição

Apesar do fato de que é sempre possível projetar um código BCH que corrija até t erros, para um t qualquer, esta afirmação deve ser interpretada com uma certa restrição, visto que as taxas dos códigos construídos são assintoticamente ruins. Em outras palavras, altas distâncias mínimas são obtidas ao custo de baixas taxas.

A seguir, apresentam-se as definições e teoremas relacionados a códigos cíclicos sobre anéis e corpos. Para as seguintes definições e teoremas, considere p um número primo e q um número inteiro maior ou igual que 4.

Definição 2.31. *Seja $\mathbb{A}$ um anel. Um módulo livre é um $\mathbb{A}$-módulo gerado por um conjunto de vetores linearmente independentes.*

Definição 2.32. *Um **código linear** (n, k) sobre $\mathbb{Z}_q$ é definido como um módulo livre de dimensão k no espaço de todas as n-uplas de ${\mathbb{Z}_q}^n$.*

Definição 2.33. *Um código linear $\mathcal{C}$ com parâmetros (n, k) sobre $\mathbb{Z}_q$ é **cíclico** se, para $v = (v_0, v_1, v_2, \ldots, v_{n-1}) \in \mathcal{C}$, todo deslocamento cíclico $v^{(1)} = (v_{n-1}, v_0, v_1, v_2, \ldots, v_{n-2}) \in \mathcal{C}$ com $v_i \in \mathbb{Z}_q$, $0 \leq i \leq n-1$.*

Os códigos cíclicos podem ser representados na forma polinomial. Portanto, considere a palavra código $v = (v_0, v_1, v_2, \ldots, v_{n-1})$ de um código cíclico $\mathcal{C}$. Esta palavra código é descrita pelo polinômio:

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

O produto entre x e $v(x)$ módulo $x^n - 1$ é dado por:

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

que corresponde à palavra código:

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

a qual é um deslocamento cíclico da palavra:

$$v = (v_0, v_1, v_2, \ldots, v_{n-1})$$

Note-se que $v^{(1)}(x)$ é obtido através do produto $x \cdot v(x)$ no anel quociente $\mathbb{A}_n = \frac{\mathbb{Z}_q[x]}{\langle x^n-1\rangle}$ onde $\langle x^n-1\rangle$ identifica o ideal gerado pelo polinômio x^n-1. Lembre que o conjunto de todas as palavras-código do código cíclico $\mathcal{C}$ é um subconjunto do anel quociente $\mathbb{A}_n$, onde o anel quociente representa todas as possíveis sequências de comprimento n.

Teorema 2.6. *Um conjunto S de elementos em $\mathbb{A}_n$ é um código cíclico se, e somente se, S é um ideal em $\mathbb{A}_n$.*

Corolário 4. *Seja $\mathcal{C}$ um ideal em $\mathbb{A}_n = \frac{\mathbb{Z}_q[x]}{\{x^n-1\}}$,i.e. um código cíclico de comprimento n. Se existir um polinômio de grau mínimo em $\mathcal{C}$, cujo coeficiente dominante é um elemento inversível em $\mathbb{Z}_q$, então o polinômio mônico de grau mínimo em $\mathcal{C}$ é único.*

Teorema 2.7. *Seja $\mathcal{C}$ um ideal em $\mathbb{A}_n$ e $g(x)$ um polinômio mônico com o menor grau em $\mathcal{C}$. Assim, $\mathcal{C} = \langle g(x)\rangle$, e portanto, o código $\mathcal{C}$ consiste de todos os múltiplos de $g(x)$. Diz-se que $\mathcal{C}$ é um ideal principal.*

Teorema 2.8. *Seja $\mathcal{C}$ um ideal principal em $\mathbb{A}_n$. Se o coeficiente dominante do polinômio de menor grau em $\mathcal{C}$, $g(x)$, é um elemento inversível, então $g(x)$ divide (x^n-1). Veja que $g(x)$ deve ser mônico para que possa dividir x^n-1.*

Teorema 2.9. *Se $g(x) \in \mathcal{C}$ e $g(x)$ divide (x^n-1), então $g(x)$ tem grau mínimo em $\mathcal{C} = \langle g(x)\rangle$.*

Os teoremas anteriores permitem a construção de códigos cíclicos sobre anéis de maneira similar à bem conhecida construção de códigos cíclicos sobre corpos. A metodologia tradicional para a construção de códigos cíclicos consiste na fatoração em polinômios irredutíveis de polinômio (x^n-1) sobre o anel de interesse; alguns polinômios irredutíveis são escolhidos e multiplicados para assim formar o polinômio gerador do código cíclico.

Note que, usando o critério da ciclicidade e de ser um módulo livre, pode-se escrever a matriz geradora do código através do polinômio gerador:

$$g(x) = g_0 + g_1x + g_2x^2 + \cdots + x^{n-k}$$

onde k é a dimensão do código.

Teorema 2.10. *Sejam C_i's códigos cíclicos sobre $\mathbb{Z}_{p_i}^{k_i}$, então, para $q = p_1^{k_1}p_2^{k_2}\ldots p_q^{k_q}$ e*

$$\mathcal{C} = \bigoplus_{i=1}^{q} C_i \ ,$$

tem-se que $\mathcal{C}$ é um código cíclico sobre $\mathbb{Z}_q$.

2.3.3.1 Códigos BCH

A seguir mostra-se a bem conhecida construção de códigos cíclicos BCH sobre corpos e a partir desta construção se deriva a construção sobre anéis locais. Em (SHANKAR, 1979) podem-se encontrar as demonstrações e mais detalhes dos seguintes fatos.

Definição 2.34. *Um código cíclico de comprimento n sobre $\mathbb{F}_p$ é denominado um* ***código BCH com distância de projeto*** *δ se o seu gerador $g(x)$ for o mínimo múltiplo comum dos polinômios minimais de $\alpha^l, \alpha^{l+1}, \ldots, \alpha^{l+\delta-2}$, para algum l, onde α é uma raiz primitiva $n-$ésima da unidade, em alguma extensão $\mathbb{F}_{p^n}$ contendo $\mathbb{F}_p$.*

Teorema 2.11. *A distância mínima do código BCH, definido acima, é pelo menos δ.*

Considere o anel de polinômios com coeficientes em $\mathbb{A}$, o qual neste trabalho serão os alfabetos $\mathbb{Z}_4$, $\mathbb{Z}_5$ e $\mathbb{F}_4$, e lembre que o anel de polinômios com coeficientes sobre $\mathbb{A}$ se denota por:

$$A[x] = \{a_0 + a_1x + a_2x + \ldots \mid a_i \in A,\ \forall i \geq 0\}$$

Veja que o anel quociente: $\mathbb{A}_n = \frac{\mathbb{A}[x]}{\langle x^n-1\rangle}$ é isomorfo como espaço vetorial a $\mathbb{A}^n$, e como visto anteriormente, um código cíclico é um ideal de $\mathbb{A}_n$ (WALKER, 2000), onde todo ideal de $\mathbb{A}_n$ é gerado por um polinômio $g(x)$ em $\mathbb{A}$ tal que $g(x)$ divide o polinômio $x^n - 1$.

No caso da Definição 2.34, α é um elemento primitivo de $\mathbb{F}_{q^n}$, então prova-se que:

Teorema 2.12. *Seja α um elemento primitivo de $\mathbb{F}_{q^n}$. O polinômio $x^n - 1$ pode ser fatorado em $\mathbb{F}_{q^n}$ e em $\mathbb{F}_q$, respectivamente:*

$$\begin{aligned} x^n - 1 =& (x - \alpha^0)(x - \alpha^1)\ldots(x - \alpha^{n-1}) \\ x^n - 1 =& F_1(x)F_2(x)\ldots F_s(x) \end{aligned}$$

onde cada polinômio $F_i(x)$ sobre $\mathbb{F}_q$ é um polinômio minimal.

Note-se que cada $F_i(x)$ divide $x^n - 1$, portanto, o múltiplo comum de alguns polinômios minimais também divide $x^n - 1$. Logo, o ideal gerado pelo polinômio gerador $g(x)$, formado pela multiplicação de alguns polinômios minimais, define um código cíclico.

A seguir se introduzem os **códigos BCH sobre anéis locais associativos comutativos**. Em especial, o anel $\mathbb{Z}_4 \simeq \mathbb{Z}_{2^2}$ é associativo comutativo e é local porque o seu único ideal maximal é $\langle 2\rangle$.

Seja $\mathbb{Z}_{p^k}[y]$ o anel de polinômios sobre o anel local $\mathbb{Z}_{p^k}$, $\phi(y)$ um polinômio irredutível de grau r sobre $\mathbb{F}_p = \mathbb{Z}_p$ (veja que $\phi(y)$ é também irredutível sobre $\mathbb{Z}_{p^k}$).

Define-se a **extensão de anel** como:

$$\mathbb{R} \simeq \frac{\mathbb{Z}_{p^k}[y]}{\phi(y)} \simeq GR(p^k, r)$$

Teorema 2.13. *Seja $\gamma \in \mathbb{R}$. Se γ gera um subgrupo cíclico de ordem $n = p^r - 1$ (denotado por G_n) no grupo das unidades de R, i.e. $G_n \subset \mathbb{R}^*$; então o grupo G_n é único.*

Teorema 2.14. *Seja $\nu \in \mathbb{R}$, tal que seu homomorfismo canônico ($\overline{\nu}$) em $\mathbb{F}_{p^r}$ é um elemento primitivo de $\mathbb{F}_{p^r}$, então:*
Existe um $\gamma \in G_n$ e algum $t \geq 1$, tal que $\gamma = \nu^t$, o qual fatora o polinômio $x^n - 1$ como:

$$\begin{aligned} x^n - 1 =& (x - \gamma^0)(x - \gamma^1)\dots(x - \gamma^{n-1}), \text{ sobre } G_n \\ x^n - 1 =& F_1(x)F_2(x)\dots F_s(x) \end{aligned}$$

onde os coeficientes de cada polinômio $F_i(x)$ estão sobre $\mathbb{Z}_{p^k}$.

Com os teoremas anteriores, pode-se demonstrar o seguinte teorema:

Teorema 2.15. *(SHANKAR, 1979) Seja $g(x)$ o polinômio gerador de um código cíclico de comprimento n sobre o alfabeto $\mathbb{Z}_{p^k}$ e sejam $\gamma^l, \gamma^{l+1}, \dots, \gamma^{l+\delta-2}$ as raízes de $g(x)$ em G_n, então, a distância mínima do código é maior ou igual a δ, onde $g(x)$ é o mínimo múltiplo comum dos polinômios minimais de $\gamma^l, \dots \gamma^{l+\delta-2}$*

2.3.3.2 Classes de códigos BCH

A seguir explicam-se as características e subclassificação dos códigos BCH que podem ser encontrados ao longo deste trabalho.

- **Códigos BCH primitivos e não primitivos:**

  Definição 2.35. *Os **códigos BCH primitivos** são aqueles códigos tais que $n = p^r - 1$, portanto, existe um elemento primitivo $\alpha \in \mathbb{F}_{p^r}$ (caso da Definição 2.34) ou $\gamma \in G_n$ (caso do Teorema 2.15), de ordem n, tais que:*

$$\begin{aligned} x^n - 1 =& (x - \alpha^0)(x - \alpha^1)\dots(x - \alpha^{n-1}) \\ x^n - 1 =& (x - \gamma^0)(x - \gamma^2)\dots(x - \gamma^{n-1}). \end{aligned}$$

  Note-se que é possível projetar um código BCH primitivo para todo $r \geq 1$, ou seja, os códigos BCH primitivos têm um comprimento que satisfaz $n = p^r - 1$, $\forall r \geq 1$.

  Exemplo 2.5. *Para $p = 2$, podem-se projetar códigos BCH primitivos de comprimento $n = 3, 7, 15, 31, 63, \dots$.*

  O anterior justifica a procura de uma maneira de flexibilizar os possíveis comprimentos, o qual é conseguido através dos **códigos BCH não-primitivos**. A ideia dos códigos não-primitivos vem do seguinte fato, o qual é verdade para qualquer um dos dois alfabetos $\mathbb{F}_p$ ou $\mathbb{Z}_{p^k}$:

Afirmação 2.1. *Se m divide n ($n = am$), então $(x^m - 1)$ divide $(x^n - 1)$.*

Assim, $x^m - 1$ deve ser fatorado por alguns polinômios $(x - \alpha^0), \ldots, (x - \alpha^{n-1})$, ou por alguns polinômios $(x - \gamma^0), \ldots, (x - \gamma^{n-1})$, dependendo do alfabeto. Prova-se, através do automorfismo de Frobenious (LANG, 1993), que se $n = am$, então o elemento $\beta = \alpha^a$ fatora o polinômio $x^m - 1$ da seguinte maneira:

$$x^m - 1 = (x - \beta^0)(x - \beta^1) \ldots (x - \beta^{m-1})$$

com coeficientes no alfabeto $\mathbb{F}_{p^r}$. De maneira equivalente, se $n = am$, então o elemento $\lambda = \gamma^a \ \in G_n$ fatora o polinômio $x^m - 1$ da seguinte maneira:

$$x^m - 1 = (x - \lambda^0)(x - \lambda^1) \ldots (x - \lambda^{m-1})$$

com coeficientes no subgrupo G_n.

Usando β ao invés de α na Definição 2.34 e usando λ ao invés de γ no Teorema 2.15, têm-se as definições dos **códigos não-primitivos** sobre o alfabeto $\mathbb{F}_p$ e $\mathbb{Z}_{p^k}$, respectivamente. A principal diferença e importância é que o comprimento do código não-primitivo é m, onde $n = am$; isto é, m é algum divisor de $n = p^r - 1$.

Exemplo 2.6. *Para $p = 2$, podem-se projetar códigos BCH não primitivos de comprimento $n = 5, 21, 39, 93, \ldots$.*

- **Códigos BCH no sentido estrito ou *narrow sense BCH codes*:**

  Definição 2.36. *Os **Códigos BCH no sentido estrito (nsBCH)** são aqueles para os quais, na Definição 2.34 e no Teorema 2.15, $l = 1$.*

  Os códigos **nsBCH** com distância de projeto δ e comprimento n, apresentam $\{\alpha, \alpha^2, \ldots, \alpha^{2t}\}$ (ou $\{\gamma, \ldots, \gamma^{2t}\}$) e seus conjugados como raízes de cada um de seus polinômios. Veja que a classe de códigos nsBCH forma uma subclasse dos códigos BCH.

- **Códigos BCH reversíveis (revBCH):**

  Definição 2.37. *Seja $c = (a_0, a_1, \ldots, a_{n-2}, a_{n-1})$ uma palavra-código num código BCH de comprimento n, $\mathcal{C}$ é dito **reversível (revBCH)**(MASSEY, 1964) se para todo $c \in \mathcal{C}$, a seguinte propriedade é satisfeita:*

  $$c' = (a_{n-1}, a_{n-2}, \ldots, a_1, a_0) \ \in \mathcal{C}$$

  Teorema 2.16. *O código BCH, denotado por $\mathcal{C}$, e gerado pelo polinômio $g(x)$ é dito **reversível** (revBCH) se:*

  $$g(x) = x^{n-1} g(x^{-1})$$

  i.e, $g(x)$ é um polinômio mônico recíproco.

O Teorema anterior pode ser expressado na mesma notação utilizada na Definição 2.34 e no Teorema 2.15 para códigos primitivos e não-primitivos.

Teorema 2.17. *(MASSEY, 1964) Seja $\beta \in \mathbb{F}_{p^r}$ (ou $\lambda \in G_n$) e seja $g(x)$ o polinômio mônico de grau mínimo tal que suas raízes são:*

$$\{\beta^{-t}, \beta^{-(t-1)}, \ldots, \beta^0\} = \{1, \beta, \beta^2, \ldots, \beta^t\} \text{ ou}$$
$$\{\lambda^{-t}, \lambda^{-(t-1)}, \ldots, \lambda^0\} = \{1, \lambda, \lambda^2, \ldots, \lambda^t\}$$

Então, $g(x)$ gera um código BCH reversível (revBCH) com distância de projeto $\delta = 2t + 2$

2.3.4 Algoritmo rápido de divisão

Como será visto no desenvolvimento deste trabalho, o Algoritmo da Divisão será utilizado intensivamente para a identificação de sequências como palavras-código de códigos BCH. Por esta razão, surge a necessidade de implementar o algoritmo da divisão da maneira mais eficiente possível. Nesta, seção mostra-se o funcionamento do algoritmo introduzido em (CAO; CAO, 2012), o qual utiliza técnica reversa e iterações de Newton.

Seja A um anel (comutativo e com identidade) e sejam $a, b \in A[x]$ polinômios de grau n e m ($\deg(a)$ e $\deg(b)$), respectivamente, onde $m \leq n$ e b é um polinômio mônico. O objetivo é encontrar os polinômios $q, r \in A[x]$, tais que: $a = qb + r$, onde $\deg(r) < \deg(b)$. Pelo fato de b ser um polinômio mônico garante a existência de q e r, unicamente.

Seja $a = a_0 + a_1 x + \cdots + a_n x^n$, onde $\text{rev}(a) = a_n + a_{n-1}x + \cdots + a_0 x^n$, então:

$$\text{rev}_n(a) = x^n a(x^{-1}) = \text{rev}_{n-m}(q) \cdot \text{rev}_m(b) + x^{n-m+1}\text{rev}(r)$$

portanto:

$$\text{rev}(a) = \text{rev}_{n-m}(q) \cdot \text{rev}_m(b) \pmod{x^{n-m+1}}$$
$$\text{rev}_{n-m}(q) = \text{rev}_n(a) \cdot \text{rev}_m(b)^{-1} \pmod{x^{n-m+1}}$$

Veja que no último passo, $\text{rev}_m(b)$ tem coeficiente constante 1, pois b é mônico, e portanto é invertível módulo x^{n-m+1}.

Assim, o problema a resolver se resume da seguinte maneira:
"Dado $f \in A[x]$ e $l \in \mathbb{N}$ com $f(0) = 1$, encontrar $g \in A[x]$ tal que $f \cdot g = 1 \pmod{x^l}$". Onde, o problema pode ser resolvido usando a iteração de Newton:

Require: $f \in A[x]$ com $f(0) = 1$ e $l \in \mathbb{N}$.
Ensure: $g \in A[x]$ que satisfaz $f \cdot g \equiv 1 \pmod{x^l}$.
1: $g_0 \leftarrow 1$, $r \leftarrow \lceil \log(l) \rceil$
2: **for** $i = 1, \ldots, r - 1$ **do**

3: $g_i \leftarrow g_{i-1} \cdot (2 - f \cdot g_{i-1}) \pmod{x^{2^i}}$
4: $g_r \leftarrow g_{r-1} \cdot (2 - f \cdot g_{r-1}) \pmod{x^l}$
5: **Return:** $g(x)$

Utilizando o algoritmo apresentado acima, completa-se o cômputo de q e r:

Require: $a, b \in A[x]$, onde A é um anel comutativo e com unidade e $b \neq 0$ é mônico.
Ensure: $q, r \in A[x]$ tais que $a = q \cdot b + r$ e $\deg(r) < \deg(b)$.
1: $g_0 \leftarrow 1$, $r \leftarrow \lceil \log(l) \rceil$
2: **if** $\deg(a) < \deg(b)$ **then**
3: **Return:** $q = 0$ e $r = a$
4: **Calcular inversa** de $\mathrm{rev}_m(b) \pmod{x^{n-m+1}}$
5: $q^* \leftarrow \mathrm{rev}_n(a) \cdot \mathrm{rev}_m(b)^{-1} \pmod{n - m + 1}$
6: **Return:** $q = \mathrm{rev}_m(q^*)$ e $r = a - bq$

Para analisar o complexidade computacional usam-se a seguinte definição e o seguinte teorema:

Definição 2.38. *Seja A um anel comutativo com unidade. A função $M : N_{>0} \to A_{>0}$ se define como o tempo de multiplicação para $A[x]$ se os polinômios em $A[x]$ de grau menor que n podem ser multiplicados usando $M(n)$ operações em A.*

Teorema 2.18. *O algoritmo computa corretamente a inversa de f modulo x^l. Este utiliza no máximo $5M(L) + l \in O(M(l))$ operações aritméticas em A. Quando l é uma potência de dois o algoritmo precisa de $3M(L) + l \in O(M(l))$ operações aritméticas em A.*

3 Algoritmo de Determinação de Códigos BCH

Um algoritmo para encontrar códigos BCH no sentido estrito (ou *Narrow-sense BCH codes*) (nsBCH) tais que uma sequência dada é uma palavra-código desses códigos foi abordado nos trabalhos (FARIA, 2011) e (ROCHA, 2010). Nestes trabalhos, o algoritmo ajudou na identificação e representação de sequências mRNA, DNA, miRNA, etc, como palavras-código de códigos nsBCH.

Neste capítulo, apresenta-se um algoritmo para encontrar códigos BCH tais que uma dada sequência pertence a esses códigos BCH. Sendo a classe de códigos nsBCH uma subclasse da classe de códigos BCH, então o algoritmo a ser introduzido permite identificar ou representar uma maior quantidade de sequências biológicas que as identificadas em (FARIA, 2011) e (ROCHA, 2010).

Este capítulo está organizado da seguinte forma. Na Seção 3.1, estabelece-se o problema e a proposta do algoritmo; na Seção 3.2, introduzem-se duas metodologias para encontrar as classes laterais ciclotômicas e os polinômios minimais, elementos que serão necessários para o cômputo do algoritmo; e na Seção 3.3 explica-se o algoritmo que encontra o código BCH com a maior distância de projeto BCH e maior cardinalidade tal que uma dada sequência é palavra-código desse código BCH.

3.1 Definição do Problema

O problema a ser resolvido pode ser escrito como: Dada s, uma sequência em $\mathbb{A}^n$, encontrar o código BCH sobre o alfabeto $\mathbb{A}$, denotado por $\mathcal{C}$, tais que $s \in \mathcal{C}$ e $\mathcal{C}$ tem a maior distância de projeto BCH, denotada por δ. Ou seja:

$$\mathcal{C} = \max_{\delta_i} \{\mathcal{C} \in \text{códigos BCH} \mid s \in \mathcal{C}\} \tag{3.1}$$

Portanto, neste capítulo, apresenta-se um algoritmo que computa um código $\mathcal{C}$ que satisfaz a Equação 3.1:

$$\mathcal{C} = \text{BCH\_One\_Seq}(s, PolsMinimais, CyclotomicCosets) \tag{3.2}$$

onde s representa a sequência $s(x)$ de comprimento n, $PolsMinimais$ é o conjunto de polinômios minimais que fatoram $x^n - 1$ e $CyclotomicCosets$ é o conjunto de classes laterais ciclotômicas módulo n e característica q, onde q é $\{2, 4, 5\}$ quando o alfabeto $\mathbb{A}$ é $\{\mathbb{Z}_4, \mathbb{F}_4, \mathbb{Z}_5\}$, respectivamente.

3.2 Polinômios Minimais e Classes Laterais Ciclotômicas

Como visto na Equação 3.2, o algoritmo *BCH_One_Seq* precisa conhecer os polinômios minimais que fatoram $x^n - 1$, as classes laterais ciclotômicas módulo n e característica q, onde q, neste trabalho, assume valores $\{2, 4, 5\}$ para os alfabetos $\{\mathbb{Z}_4, \mathbb{F}_4, \mathbb{Z}_5\}$, respectivamente; e n somente valores que satisfaçam a restrição de comprimento BCH dada pela Equação 3.3.

$$n|(q^s - 1). \tag{3.3}$$

O algoritmo *BCH_One_Seq* será usado para encontrar códigos BCH sobre os alfabetos $\mathbb{Z}_4$, $\mathbb{F}_4$ e $\mathbb{Z}_5$; porém este algoritmo pode ser facilmente estendido para anéis locais, classe à qual os corpos pertencem.

Definição 3.1. *Dado que $\mathbb{A}$ somente pode ser o anel $\mathbb{Z}_4$ ou algum dos corpos $\mathbb{F}_4$ ou $\mathbb{Z}_5$, define-se $\Gamma = (\alpha)$ como o **grupo separante de** $x^n - 1$, gerado por α, onde Γ é um subgrupo das unidades da s-ésima extensão do anel ou corpo, tal que a equação a seguir é satisfeita:*

$$x^n - 1 = (x - \alpha^0)(x - \alpha) \dots (x - \alpha^{n-1}), \tag{3.4}$$

onde α tem ordem n ($\alpha^n = 1$) e $\alpha \in GR(4, s)^*$, no caso que $\mathbb{A} = \mathbb{Z}_4$, ou $\alpha \in GF(q, s)^*$ no caso que $\mathbb{A} = \mathbb{Z}_5$ ou $\mathbb{A} = \mathbb{F}_4$.

Quando $\mathbb{A}$ é um corpo ($\mathbb{A} = \mathbb{Z}_5$ ou $\mathbb{A} = \mathbb{F}_4$); $\Gamma = (\alpha)$, o **grupo separante de** $x^n - 1$, é um subcorpo de $GF(q, s)$. A seguir, apresentam-se as metodologias para o cômputo de: $PolsMinimais$ e $CyclotomicCosets$.

3.2.1 Cômputo das classes laterais ciclotômicas

As classes laterais ciclotômicas modulo n e característica q são computadas usando a Definição 3.2.

Definição 3.2. *A i-ésima **classe lateral ciclotômica modulo** n **e característica** q é definida como:*

$$\mathcal{X}_i = \left\{ \left((i)_n, (i \cdot q)_n, (i \cdot q^2), \dots, (i \cdot q^{r_i - 1})_n \right) \right\}, \text{ para } 1 \leq i \leq n - 1, \tag{3.5}$$

onde r_i é o menor inteiro positivo tal que $(i \cdot q^{r_i})_n = (i)_n$.

Lema 3.1. *Considere i e j tais que $i \in \mathcal{X}_j$, então $\mathcal{X}_i = \mathcal{X}_j$.*

Demonstração: Como $i \in \mathcal{X}_j$, então, existe l inteiro tal que $(i)_n = (j \cdot q^l)_n$. Após multiplicar i por $q^{r_j - l}$, obtém-se $(i \cdot q^{r_j - l})_n = (j)_n$. Portanto, $j \in \mathcal{X}_i$ e $\mathcal{X}_i = \mathcal{X}_j$. ∎

Note que $\mathcal{X}_i = \mathcal{X}_j$ quando $i \in \mathcal{X}_j$; portanto, devem-se desconsiderar as classes laterais repetidas.

A importância das classes laterais ciclotômicas é que elas guardam uma estreita relação com as raízes conjugadas de um polinômio minimal $f_i(x)$. Como será visto no Lema 3.2.

Exemplo 3.1. *As classes laterais ciclotômicas modulo* 63 *e característica* 2 *são dadas por:*

$$\begin{aligned}
\mathcal{X}_1 &= \{1, 2, 4, 8, 16, 32\} \\
\mathcal{X}_3 &= \{3, 6, 12, 24, 48, 33\} \\
\mathcal{X}_5 &= \{5, 10, 20, 40, 17, 34\} \\
\mathcal{X}_7 &= \{7, 14, 28, 56, 49, 35\} \\
\mathcal{X}_9 &= \{9, 18, 36\} \\
\mathcal{X}_{11} &= \{11, 22, 44, 25, 50, 37\} \\
\mathcal{X}_{13} &= \{13, 26, 52, 41, 19, 38\} \\
\mathcal{X}_{15} &= \{15, 30, 60, 57, 51, 39\} \\
\mathcal{X}_{21} &= \{21, 42\} \\
\mathcal{X}_{23} &= \{23, 46, 29, 58, 53, 43\} \\
\mathcal{X}_{27} &= \{27, 54, 45\} \\
\mathcal{X}_{31} &= \{31, 62, 61, 59, 55, 47\}.
\end{aligned}$$

3.2.2 Cômputo dos polinômios minimais

O polinômio $x^n - 1$ é fatorado de maneira única e por polinômios não repetidos sobre Γ, o grupo separante de $x^n - 1$, e sobre $\mathbb{A}$, de acordo com a seguinte equação ((WAN; WAN, 1998)):

$$x^n - 1 = (x - \alpha^0) \dots (x - \alpha^{n-1}) = f_1(x) \dots f_m(x), \tag{3.6}$$

onde s polinômios $f_i(x)$'s são ditos **polinômios minimais** em $\mathbb{A}[x]$ e há exatamente o mesmo número de polinômios minimais e o número de classes laterais ciclotômicas módulo n e característica q.

Definição 3.3. *Um polinômio $f_i(x)$ com coeficientes em $\mathbb{A}$ é chamado* ***polinômio minimal****, se:*

$$\nexists \;\; p_1(x), p_2(x) \in \mathbb{A}[x], \text{ tal que: } \quad f_i(x) = p_1(x) \cdot p_2(x)$$

Lema 3.2. *Sejam $f_i(x) \in \frac{\mathbb{A}[x]}{\langle x^n - 1 \rangle}$ um polinômio minimal, α um elemento gerador do grupo separante de $x^n - 1$ e $\mathcal{X}_i$ uma classe lateral. Se $f_i(\alpha^i) = 0$; então:*

$$f_i(\alpha^l) = 0, \qquad \forall l \in \mathcal{X}_i \tag{3.7}$$

Demonstração:

- Quando $\mathbb{A}$ é um corpo. Sabe-se que $f_i(x) = a_0 + \cdots + a_k x^k + \cdots + a_{n-1}x^{n-1}$, onde $a_k \in \mathbb{A}$, e $f_i(\alpha^i) = a_0 + \cdots + a_{n-1}(\alpha^i)^{n-1}$. Seja $j \in \mathcal{X}_i$, portanto $j = (q^l)i$.
  Seja $f(\alpha^j) = a_0 + \cdots + a_{n-1}(\alpha^{((q^l)\cdot i)_n})^{n-1}$, então:

$$\begin{aligned} f(\alpha^j) =& a_0 + \cdots + a_k\alpha^{(ikq^l)_n} + \cdots + a_{n-1}\alpha^{((q^l)\cdot i(n-1))_n} \\ f(\alpha^j) =& a_0^{(q^l)} + \cdots + a_k^{(q^l)}\alpha^{(ikq^l)_n} + \cdots + a_{n-1}^{(q^l)}\alpha^{((q^l)\cdot i(n-1))_n} \\ f(\alpha^j) =& \left(a_0 + \cdots + a_k\alpha^{(ik)_n} + \cdots + a_{n-1}\alpha^{(i(n-1))_n}\right)^{(q^l)} \\ f(\alpha^j) =& (0)^{(q^l)} \\ f(\alpha^j) =& 0, \end{aligned}$$

  onde $a_k = a_k^{q^l}$ porque $\mathbb{A}$ é um corpo de ordem q e $(a^{q^l} + b^{q^l}) = (a+b)^{q^l}$.

- Quando $\mathbb{A}$ é $\mathbb{Z}_4$. Ver a demonstração na Proposição 6.14 em (WAN; WAN, 1998).

∎

Note que quando $\mathbb{A}$ for um corpo de ordem q, o Lema 3.2 pode ser generalizado para qualquer polinômio $f(x)$, não necessariamente minimal, e a demonstração é igual à anterior.

Passos para o cômputo das classes laterais ciclotômicas:

As classes laterais ciclotômicas são usadas para computar todos os polinômios minimais $f_i(x)$'s, da seguinte maneira:

1. Usando um polinômio primitivo $p(x)$ de grau s, realizar a extensão de anel ou de corpo (dependendo de $\mathbb{A}$).

2. Identificar o elemento α que fatora o polinômio $x^n - 1$ sobre $\Gamma = (\alpha)$, de acordo com a Equação 3.6.

3. Para cada classe lateral $\mathcal{X}_i$, computar o i-ésimo polinômio minimal $f_i(x)$ de acordo com a seguinte equação:

$$f_i(x) = \prod_{l \in \mathcal{X}_i} \left(x - \alpha^l\right) \tag{3.8}$$

Exemplo 3.2. *Encontrar os polinômios minimais sobre $\mathbb{Z}_4$ que fatoram de maneira única $x^{63} - 1$. Veja que $s = 6$, dado que $n = 63 = 2^6 - 1$.*

1. Realiza-se a extensão de anel $GR(4, 6) = \frac{\mathbb{Z}_4[x]}{\langle p(x)\rangle}$, onde $p(x) = 1 + x + x^6$ é um polinômio primitivo de grau 6 sobre $\mathbb{Z}_2$.

2. Considere $\beta = \alpha^2$, onde α é o elemento que satisfaz $p(\alpha) = 0$ sobre $\mathbb{Z}_4$. Em (SHANKAR, 1979) mostrou-se que $\Gamma = (\beta)$ e que:

$$x^{63} - 1 = (x-1)(x-\beta)(x-\beta^2)\ldots(x-\beta^{62}).$$

3. Usando as classes ciclotômicas computadas no Exemplo 3.1, computam-se os polinômios minimais:

$$\begin{aligned}
f_0(x) &= 3 + x \\
f_1(x) &= 1 + 3x + 2x^3 + x^6 \\
f_3(x) &= 1 + x + 3x^2 + 3x^4 + 2x^5 + x^6 \\
f_5(x) &= 1 + x + x^2 + 2x^4 + 3x^5 + x^6 \\
f_7(x) &= 1 + x^3 + x^6 \\
f_9(x) &= 3 + 2x + 3x^2 + x^3 \\
f_{11}(x) &= 1 + 2x + x^2 + x^3 + 3x^5 + x^6 \\
f_{13}(x) &= 1 + 3x + x^3 + x^4 + 2x^5 + x^6 \\
f_{15}(x) &= 1 + 2x + 3x^2 + 3x^4 + x^5 + x^6 \\
f_{21}(x) &= 1 + x + x^2 \\
f_{23}(x) &= 1 + 3x + 2x^2 + x^4 + x^5 + x^6 \\
f_{27}(x) &= 3 + x + 2x^2 + x^3 \\
f_{31}(x) &= 1 + 2x^3 + 3x^5 + x^6.
\end{aligned}$$

Nos Exemplos 3.1 e 3.2, foram considerados os casos em que a restrição BCH (Equação 3.3) é satisfeita com igualdade ($n = q^s - 1$); nesse caso o código BCH a ser obtido é dito *código BCH primitivo*.

Seja n o comprimento do código. Quando a Equação 3.3 pode ser reescrita como $a \cdot n = q^s - 1$, para $a > 1$, o código a ser obtido é dito *código BCH não primitivo* e, nesse caso, os *polinômios minimais* de $x^n - 1$ formam um subconjunto dos polinômios minimais de $x^{an} - 1$. De fato, se α é o elemento gerador do grupo separante de $x^{an} - 1$, i.e.,

$$x^{an} - 1 = (x-1)(x-\alpha)\cdot(x-\alpha^{an-1}),$$

então $\beta = \alpha^a$ fatora o polinômio $x^n - 1$, ou seja,

$$x^n - 1 = (x-1)(x-\beta)\ldots(x-\beta^{n-1})$$

e os polinômios minimais de $x^n - 1$ podem ser computados usando a mesma metodologia usada anteriormente. Usam-se as classes laterais ciclotômicas módulo n e característica q e a Equação 3.8 é substituída pela Equação 3.9.

$$f_i(x) = \prod_{l \in \mathcal{X}_i} \left(x - \beta^l\right). \tag{3.9}$$

Exemplo 3.3. *Encontrar os polinômios minimais sobre* $\mathbb{Z}_4$ *que fatoram de maneira única* $x^{21}-1$. *Veja que* $s=6$ *e* $a=3$, *dado que* $n\cdot a = 21\cdot 3 = 2^6-1$.

1. Realize a extensão de anel $GR(4,6) = \frac{\mathbb{Z}_4[x]}{\langle p(x)\rangle}$, onde $p(x) = 1+x+x^6$ é um polinômio primitivo de grau 6 sobre $\mathbb{Z}_2$.

2. Considere $\beta = \alpha^2$, onde α é o elemento que satisfaz $p(\alpha) = 0$ sobre $\mathbb{Z}_4$. Em (SHANKAR, 1979) mostrou-se que:

$$x^{63}-1 = (x-1)(x-\beta)(x-\beta^2)\ldots(x-\beta^{62}).$$

3. Calcule as classes laterais ciclotômicas módulo 21 e característica 2:

$$\begin{aligned}\mathcal{X}_1 &= \{1,2,4,8,16,11\}\\ \mathcal{X}_3 &= \{3,6,12\}\\ \mathcal{X}_5 &= \{5,10,20,19,17,13\}\\ \mathcal{X}_7 &= \{7,14\}\\ \mathcal{X}_9 &= \{9,18,15\}.\end{aligned}$$

4. Usando as classes ciclotômicas, compute os polinômios minimais com $\gamma = \beta^3$:

$$\begin{aligned}f_0(x) &= 3+x\\ f_1(x) &= 1+x+3x^2+3x^4+2x^5+x^6\\ f_3(x) &= 3+2x+3x^2+x^3\\ f_5(x) &= 1+2x+3x^2+3x^4+x^5+x^6\\ f_7(x) &= 1+x+x^2\\ f_9(x) &= 3+x+2x^2+x^3\end{aligned}$$

Da mesma forma em que foram computados os polinômios minimais $f_i(x)$ de x^n-1, pode-se definir a seguinte relação bijetora entre os polinômios minimais $f_i(x)$'s e as classes laterais ciclotômicas.

Definição 3.4. *Considere* Γ *como o grupo separante de* x^n-1 *sobre* $\mathbb{A}$ *e* $p(x)$ *como o polinômio primitivo usado na extensão de anel ou corpo, dependendo de* $\mathbb{A}$. *Define-se a função* Υ *como:*

$$\Upsilon \ : \ f_i(x) \mapsto \mathcal{X}_i,$$

onde $f_i(x) = \prod_{l\in\mathcal{X}_i}(x-\alpha^l)$ *e* $\Gamma = (\alpha)$.

Lema 3.3. Υ *é uma função bijetora.*

Demonstração: Dado que os $\mathcal{X}_i$'s são classes laterais e que o produto de polinômios mônicos $(x - \alpha^i)$ é biunívoco, então a demonstração segue. ∎

Exemplo 3.4. *No Exemplo 3.3,* Υ *é dado por:*

$$f_0 \mapsto \{0\}$$
$$f_1 \mapsto \mathcal{X}_1$$
$$f_3 \mapsto \mathcal{X}_3$$
$$f_5 \mapsto \mathcal{X}_5$$
$$f_7 \mapsto \mathcal{X}_7$$
$$f_9 \mapsto \mathcal{X}_9.$$

Na Figura 15, mostra-se o diagrama de blocos do procedimento necessário para identificar $\mathcal{C}$, o código BCH de comprimento n com a maior distância de projeto BCH, sobre o alfabeto $\mathbb{A}$, no qual $s(x)$ é uma palavra-código de tamanho n. O alfabeto $\mathbb{A}$ pode ser $\mathbb{Z}_4$, $\mathbb{Z}_5$ e $\mathbb{F}_4$; e q assume valores entre 2, 5 e 4, respectivamente.

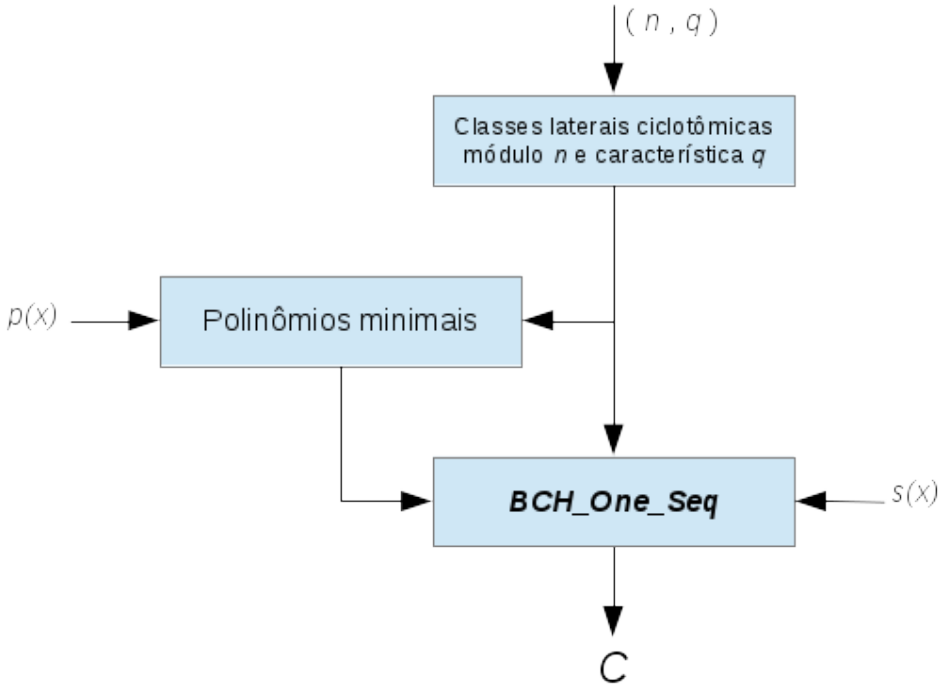


Figura 15 – Diagrama de blocos para encontrar o código BCH com a maior distância de projeto e a maior cardinalidade que contém $s(x)$.

3.3 Algoritmo para a Determinação de Códigos BCH

A ideia básica do algoritmo $\mathcal{C} = \text{BCH\_One\_Seq}(s, PolsMinimais, CyclotomicCosets)$ é mostrada no diagrama da Figura 16.

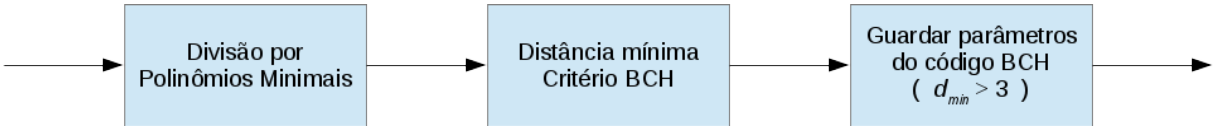


Figura 16 – Passos básicos do algoritmo BCH_One_Seq.

3.3.1 Divisão por polinômios minimais

Como visto no diagrama de blocos, o primeiro passo é realizar a divisão de polinômios entre a sequência $s(x)$ e todos os polinômios minimais. Dado que os códigos BCH são cíclicos e ideais do anel de polinômios $\frac{\mathbb{A}[x]}{\langle x^n-1\rangle}$, segue que o algoritmo BCH_One_Seq deve procurar todos os possíveis ideais nos quais $s(x)$ é uma palavra-código.

Os ideais do anel de polinômios são da forma: $I_1 = \langle f_1(x)\rangle$, $I_1 = \langle f_1(x)\rangle$, ... ,$I_m = \langle f_m(x)\rangle$ ou interseções de alguns desses ideais, onde os $f_i(x)$'s são polinômios minimais. Lembre que $(I_i \bigcap Ij) = \langle f_i(x) \cdot f_j(x)\rangle$, para $i \neq j$.

Dado que $s(x) \in I_i$ se, e somente se, $f_i(x)|s(x)$ (como visto na Seção 2.3.3), segue que podemos encontrar $\mathcal{C}_m$, o código BCH com a menor cardinalidade no qual $s(x)$ é uma palavra-código, ao computar a seguinte equação:

$$\mathcal{C}_m = \langle g_m(x)\rangle = \langle\ \{\text{mmc}\{\ldots, f_i(x), \ldots\} \mid f_i(x)|s(x)\}\ \rangle = \left\langle \prod_{f_i(x)|s(x)} f_i(x) \right\rangle, \tag{3.10}$$

onde $\mathcal{C}_m$ é um ideal gerado pelo polinômio $g_m(x)$ e $g_m(x)$ divide $(x^n - 1)$.

Ao receber uma sequência $s(x)$ e computar o algoritmo BCH_One_Seq, o primeiro passo a ser feito é realizar as divisões entre $s(x)$ e cada um dos polinômios minimais $f_i(x)$'s que fatoram o polinômio $x^n - 1$. Portanto, o algoritmo da divisão a ser usado deve ser o mais eficiente possível para reduzir o tempo de cômputo. Por esta razão, implementou-se o algoritmo rápido apresentado na Seção 2.3.4 (CAO; CAO, 2012).

3.3.2 Distância mínima e critério BCH

O segundo passo é identificar a distância mínima do código $\mathcal{C}_m$ de acordo com o critério BCH. Dado que as classes ciclotômicas, o polinômio gerador e o elemento β, tal que, $(x^n - 1) = \prod_{i=0}^{n}(x - \beta^i)$, são conhecidos, segue que, δ, a distância de projeto do código BCH, pode ser computada e, assim, encontrar um limitante inferior para a distância mínima ($d_{\mathcal{C}_m}$) do código BCH: $\delta \leq d_{\mathcal{C}_m}$.

Como visto na Seção 2.3.3, a distância de projeto é computada através das raízes de $g_m(x)$. Seja, $\beta^l, \ldots \beta^{l+\delta-2}$, a maior sequência consecutiva de raízes de $g_m(x)$, então, o código $\mathcal{C}_m$ tem distância de projeto δ. Portanto, deve-se encontrar todas as raízes de $g_m(x)$ para depois identificar a maior sequência de raízes consecutivas e o valor de δ.

Uma primeira tentativa para encontrar as *raízes de* $g_m(x)$ (Roots$(g_m(x))$) é realizar o cálculo explícito de: Roots$(g_m(x)) = \{\beta^i \mid 0 \leq i \leq n-1,\ g(\beta^i) = 0\}$. Porém, esse cálculo não é necessário porque esta informação já foi computada indiretamente na Equação 3.10, como será mostrado a seguir. Pela Equação 3.10, segue que $g_m(x) = \prod_{f_i(x)|s(x)} f_i(x)$. Portanto as raízes de $g_m(x)$ são a união das raízes de cada um dos polinômios minimais $f_i(x)$ que divide $s(x)$, ou seja,

$$\text{Roots}(g_m(x)) = \bigcup_{f_i(x)|s(x)} \text{Roots}(f_i(x)). \tag{3.11}$$

As raízes de cada $f_i(x)$ foram identificadas no cômputo dos polinômios minimais, como se observa na Equação 3.9, e são: Roots$(f_i(x)) = \{\beta^i | i \in \mathcal{X}_i\}$. Portanto, as raízes de $g_m(x)$ podem ser encontradas através da seguinte equação:

$$\text{Roots}(g_m(x)) = \left\{ \beta^j \mid j \in \bigcup_{f_i(x)|s(x)} \mathcal{X}_i \right\}. \tag{3.12}$$

onde a relação entre $f_i(x)$ e $\mathcal{X}_i$ é dada por Υ (Definição 3.4).

Usando a Equação 3.12, pode-se re-escrever o critério BCH (Teorema 2.15) para encontrar a distância de projeto com os elementos obtidos neste capítulo:

Teorema 3.1. *Sejam $g(x)$ o polinômio gerador do código $\mathcal{C}$ de comprimento n sobre o alfabeto, $\mathbb{A}$ como definido pela Equação 3.10, β uma raiz primitiva de $\Gamma = (\beta)$ e $\bigcup_{f_i(x)|s(x)} \mathcal{X}_i$ o conjunto de potências das raízes de $g(x)$. A distância mínima do código $\mathcal{C}$ é maior ou igual a δ, onde $\delta - 1$ é a cardinalidade do conjunto H, o conjunto com a maior sequência de números inteiros consecutivos módulo n no conjunto:*

$$\bigcup_{f_i(x)|s(x)} \mathcal{X}_i.$$

Usando o Teorema 3.1 sobre $g_m(x)$, obtém-se a distância de projeto δ e o conjunto H. Com estes elementos, encontra-se o código BCH de comprimento n e distância de projeto δ que contém o polinômio $s(x)$, como mostra a seguinte equação:

$$C = \langle g(x) \rangle = \left\langle \ \{\text{mmc}\{\ldots, f_i(x), \ldots\} \mid f_i(\beta^l) = 0 \text{ e } l \in H\} \ \right\rangle. \tag{3.13}$$

Aplicando os conceitos introduzidos, mostra-se o diagrama de blocos do algoritmo *BCH_One_Seq* na Figura 17. Neste diagrama, o algoritmo retorna $\mathcal{C}$ somente se a distância de projeto é maior que três. Este parâmetro pode ser escolhido dependendo da situação na qual o algoritmo for utilizado.

Como foi apresentado em (FARIA, 2011) e (ROCHA, 2010), precisa-se realizar este procedimento para cada possível polinômio primitivo de grau r, onde r é o grau

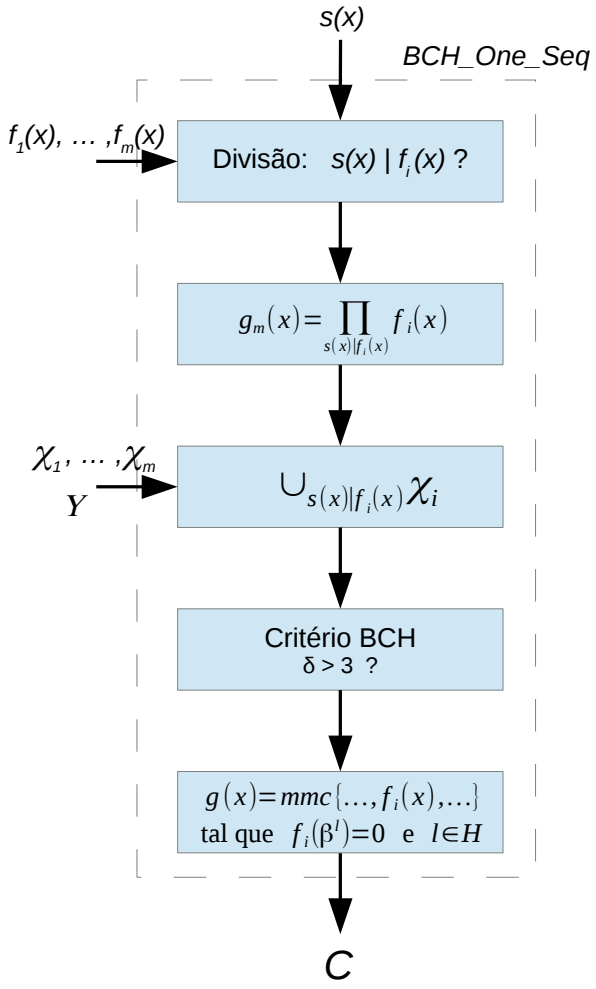


Figura 17 – Diagrama de blocos do algoritmo BCH_One_Seq.

da extensão. A razão fundamental se explica através do Exemplo 3.5, i .e., o cômputo da distância de projeto, segundo o critério BCH, não é invariante com a escolha do polinômio primitivo usado no cômputo da extensão do anel ou corpo.

Exemplo 3.5. *Considere o Exemplo 3.2. Seja $\mathcal{C}$ um código BCH com polinômio gerador $g(x) = 1 + x + x^2 + 2x^4 + 3x^5 + x^6$ tal que $g(x)|s(x)$.*

Usando o polinômio primitivo $p(x) = 1 + x + x^6$ na extensão de anel $GR(4, 6) = \frac{\mathbb{Z}_4[x]}{\langle p(x)\rangle}$, o critério BCH estabelece que $\mathcal{C}$ tem uma distância de projeto igual a 2, pois $f_5(x) = g(x)$, Roots$(g(x)) = \{5, 10, 20, 40, 17, 34\} = \mathcal{X}_5$ e a maior sequência de números consecutivos é 1, onde $1 = \delta - 1$. Usando o polinômio primitivo $\hat{p}(x) = 1 + x + x^2 + x^5 + x^6$ na extensão de anel $GR(4, 6) = \frac{\mathbb{Z}_4[x]}{\langle \hat{p}(x)\rangle}$, o critério BCH estabelece que $\mathcal{C}$ tem uma distância de projeto igual a 3. O polinômio $\hat{f}_1(x) = 1+x+x^2+2x^4+3x^5+x^6$ divide unicamente $s(x)$ e está relacionado com a classe ciclotômica $\mathcal{X}_1$. Portanto, $s(x)$ é gerado pelo código BCH com polinômio gerador $g(x) = \hat{f}_1(x) = f_5(x)$ e distância de projeto igual 3, de acordo com o critério BCH, pois Roots$(g(x)) = \{1, 2, 4, 8, 16, 32\} = \mathcal{X}_1$ é a maior sequência de números inteiros consecutivos é 2, onde $2 = \delta - 1$.

No Exemplo 3.5, quando usado o polinômio primitivo $p(x) = 1+x+x^6$ na extensão de anel $GR(4,6) = \frac{\mathbb{Z}_4[x]}{\langle p(x)\rangle}$, obtém-se o mapa Υ_1 que relaciona os polinômios minimais f_i's com as classes laterais ciclotômicas; e quando usado o polinômio primitivo $\hat{p}(x) = 1 + x + x^2 + x^5 + x^6$, obtém-se o mapa Υ_2, como mostrados na Tabela 6.

Tabela 6 – Mapas Υ_1 e Υ_2 do Exemplo 3.5.

Υ_1^{-1} $\mapsto$		Υ_2 $\mapsto$
$\mathcal{X}_0$	$f_0(x) = 3 + x$	$\mathcal{X}_0$
$\mathcal{X}_1$	$f_1(x) = 1 + 3x + 2x^3 + x^6$	$\mathcal{X}_{13}$
$\mathcal{X}_3$	$f_3(x) = 1 + x + 3x^2 + 3x^4 + 2x^5 + x^6$	$\mathcal{X}_{15}$
$\mathcal{X}_5$	$f_5(x) = 1 + x + x^2 + 2x^4 + 3x^5 + x^6$	$\mathcal{X}_1$
$\mathcal{X}_7$	$f_7(x) = 1 + x^3 + x^6$	$\mathcal{X}_7$
$\mathcal{X}_9$	$f_9(x) = 3 + 2x + 3x^2 + x^3$	$\mathcal{X}_{27}$
$\mathcal{X}_{11}$	$f_{11}(x) = 1 + 2x + x^2 + x^3 + 3x^5 + x^6$	$\mathcal{X}_5$
$\mathcal{X}_{13}$	$f_{13}(x) = 1 + 3x + x^3 + x^4 + 2x^5 + x^6$	$\mathcal{X}_{23}$
$\mathcal{X}_{15}$	$f_{15}(x) = 1 + 2x + 3x^2 + 3x^4 + x^5 + x^6$	$\mathcal{X}_3$
$\mathcal{X}_{21}$	$f_{21}(x) = 1 + x + x^2$	$\mathcal{X}_{21}$
$\mathcal{X}_{23}$	$f_{23}(x) = 1 + 3x + 2x^2 + x^4 + x^5 + x^6$	$\mathcal{X}_{31}$
$\mathcal{X}_{27}$	$f_{27}(x) = 3 + x + 2x^2 + x^3$	$\mathcal{X}_9$
$\mathcal{X}_{31}$	$f_{31}(x) = 1 + 2x^3 + 3x^5 + x^6$	$\mathcal{X}_{11}$

Isto mostra que ao considerar o polinômio primitivo $\hat{p}(x)$, pode-se encontrar um valor de distância de projeto maior para o código BCH gerado por $g(x) = 1 + x + x^2 + 2x^4 + 3x^5 + x^6$, que quando se considera o polinômio primitivo $p(x)$, i.e., a distância de projeto, segundo o critério BCH, não é invariante com a escolha do polinômio primitivo usado no cômputo da extensão do anel ou corpo. Esta descoberta é contraintuitiva dado que $GR(4,6) = \frac{\mathbb{Z}_4[x]}{\langle p(x)\rangle}$ é isomorfo a $GR(4,6) = \frac{\mathbb{Z}_4[x]}{\langle \hat{p}(x)\rangle}$.

Para contornar esta situação, em (FARIA, 2011) e (ROCHA, 2010), todos os possíveis polinômios primitivos foram considerados. Portanto, usando esta mesma técnica, o algoritmo BCH_One_Seq deve ser repetido para cada uma das extensões de anel ou corpo que resultam ao considerar todos os polinômios primitivos. A seguir, apresenta-se uma metodologia equivalente à usada em (FARIA, 2011) e (ROCHA, 2010) que permite obter os mesmos resultados, porém somente um único polinômio primitivo é necessário e $GR(4,6)$ é computado uma única vez.

Sejam $p_1(x)$ e $p_2(x)$ dois polinômios primitivos em $\mathbb{F}_2[x]$ ou $\mathbb{Z}_5[x]$ ou $\mathbb{F}_4[x]$, dependendo se o alfabeto $\mathbb{A}$ for $\mathbb{Z}_4$ ou $\mathbb{Z}_5$ ou $\mathbb{F}_4$, respectivamente. No caso quando $\mathbb{A}$ for $\mathbb{Z}_4$ e usando $p_1(x)$ e $p_2(x)$ para fazer a extensão de anel, obtêm-se:

$$GR(4,s)^* = \left(\frac{\mathbb{Z}_4[x]}{\langle p_1(x)\rangle}\right)^* \supseteq \Gamma_1 = \{1, \beta, \ldots, \beta^{n-1}\}$$

$$GR(4,s)^* = \left(\frac{\mathbb{Z}_4[x]}{\langle p_2(x)\rangle}\right)^* \supseteq \Gamma_2 = \{1, \gamma, \ldots, \gamma^{n-1}\}.$$

Veja que no caso de $\mathbb{A}$ ser um corpo, uma análise similar pode ser feita.

Dado que $p_1(x)$ e $p_2(x)$ são polinômios primitivos, segue que, tanto β como γ são elementos primitivos de Γ_1 e Γ_2, respectivamente, e a seguinte relação pode ser estabelecida:

$$(x-\beta^0)\dots(x-\beta^{n-1}) = x^n - 1 = f_1(x)\dots f_m(x) = (x-\gamma^0)\dots(x-\gamma^{n-1})$$

e esta equação verifica o fato que Γ_1 é isomorfo como grupo multiplicativo com Γ_2, o qual foi estudado em (BINI; FLAMINI, 2002). O mapa bijetor entre Γ_1 e Γ_2 é dado na forma $\gamma^j \mapsto \beta$, para um j menor que n. Assim, $\beta^i \mapsto (\gamma^j)^i = \gamma^{i\cdot j}$ e observa-se que a condição: $\gcd(j,n) = 1$ deve ser satisfeita para que $\gamma = \beta^j$ seja um elemento primitivo.

No Exemplo 3.5, a diferença no cômputo da distância de projeto ocorre porque as raízes de $g(x) = 1 + x + x^2 + 2x^4 + 3x^5 + x^6$ sobre Γ_1 são $\{\beta^i \mid i \in \mathcal{X}_5\}$, onde $\mathcal{X}_5 = \{5, 10, 17, 20, 34, 40\}$, e as raízes de $g(x) = 1 + x + x^2 + 2x^4 + 3x^5 + x^6$ sobre Γ_2 são $\{\gamma^i \mid i \in \mathcal{X}_1\}$, onde $\mathcal{X}_1 = \{1, 2, 4, 8, 16, 32\}$. Aplicando o critério BCH, confirma-se que $\mathcal{C}$ tem distância de projeto 3 porque $\{1, 2\} \subset \mathcal{X}_1$. Assim, as distintas distâncias de projeto são obtidas pela diferença entre as relações bijetoras Υ_1, entre os $\{\dots, \mathcal{X}_i, \dots\}$ e os $\{\dots, f_i(x), \dots\}$, e Υ_2, entre os $\{\dots, \mathcal{X}_i, \dots\}$ e os $\{\dots, f_i(x), \dots\}$; obtidas quando os polinômios minimais são computados através de dois diferentes polinômios primitivos (ver Seção 3.2.2).

Dado que as funções Υ_1 e Υ_2 são as responsáveis pela obtenção de diferentes distâncias de projeto, a seguir analisa-se uma metodologia simples que permite transformar Υ_1 em Υ_2. Esta metodologia está baseada no isomorfismo entre Γ_1 e Γ_2, definido pelo seguinte mapeamento:

$$\begin{aligned} Iso &: \Gamma_2 \to \Gamma_1, \quad Iso : \gamma^j \mapsto \beta \\ Iso^{-1} &: \Gamma_1 \to \Gamma_2, \quad Iso^{-1} : \beta^h \mapsto \gamma \quad \text{onde} \quad (j \cdot h)_n = 1. \end{aligned} \tag{3.14}$$

Afirmação 3.1. *Se Γ_1 e Γ_2 são isomorfos como grupo multiplicativo através do mapa da Equação 3.14, então* $\gcd(j,n) = 1$. *Além disso, se $(\beta) = \Gamma_1$ e* $\gcd(j,n) = 1$, *então, $(\beta^j) = \Gamma_1$.*

Demonstração: Por contradição, suponha que Γ_1 e Γ_2 são isomorfos através do mapa da Equação 3.14 e que $\gcd(j,n) \neq 1$. Portanto, $n = a \cdot j$, para algum $a < n$ e $j \neq 1$. Considere $\gamma^{a\cdot j}$, onde $\gamma^{a\cdot j} = \gamma^n = 1 = \beta^a\ \beta$. Isto é uma contradição porque β é um elemento de ordem n e $a < n$. Dado que $\gcd(j,n) = 1$, segue que $\exists l < n$, tal que $(l \cdot j)_n = 1$, e assim $(\beta^j)^l = \beta$ e $(\beta^j) = \Gamma_1$. ∎

Dado que a Afirmação 3.1 é uma condição necessária e não suficiente, segue que não se conhece o valor de j que define o mapa entre Γ_1 e Γ_2; mas, sabe-se que j deve

satisfazer $\gcd(j, n) = 1$. Assim, realiza-se uma busca exaustiva entre as unidades de $\mathbb{Z}_n$, até identificar o elemento j que satisfaz a Equação 3.14.

Usando o isomorfismo Iso, pode-se obter o mapeamento Υ_2 a partir do mapeamento Υ_1. Sabe-se que o conjunto $\mathcal{X}_i$ representa as raízes do polinômio $f_i(x)$ como elementos de Γ_1, portanto $f_i(x) = \prod_{i \in \mathcal{X}_i}(x - \beta^i)$. Assim, ao aplicar o isomorfismo, obtém-se:

$$
\begin{aligned}
f_i(x) &= \prod_{l \in \mathcal{X}_i} (x - \beta^l), \quad \text{sobre } \Gamma_1 \\
f_i(x) &= \prod_{l \in \mathcal{X}_i} (x - (\gamma^j)^l) \\
f_i(x) &= \prod_{l \in \mathcal{X}_i} (x - \gamma^{(j \cdot l)_n}) \\
f_i(x) &= \prod_{l \in (j \cdot \mathcal{X}_i)} (x - \gamma^l) \\
f_i(x) &= \prod_{l \in \cdot \mathcal{X}_{(i \cdot j)_n}} (x - \gamma^l), \quad \text{sobre } \Gamma_2,
\end{aligned}
$$

onde $(j \cdot \mathcal{X}_i)$ representa o conjunto $\{(j \cdot i)_n \mid i \in \mathcal{X}_i\}$ que é igual ao conjunto $\mathcal{X}_{(i \cdot j)_n}$. Portanto, Υ_2 pode ser obtido através de Υ_1 ao realizar o procedimento indicado pela Equação 3.15. De maneira equivalente, Υ_1 pode ser obtido através de Υ_2 ao realizar o procedimento indicado pela seguinte equação:

$$\Upsilon_2(f_i(x)) = \mathcal{X}_{(i \cdot j)_n}, \quad \text{onde: } \Upsilon_1(f_i(x)) = \mathcal{X}_i \tag{3.15}$$

$$\Upsilon_1(f_i(x)) = \mathcal{X}_{(i \cdot h)_n}, \quad \text{onde: } \Upsilon_2(f_i(x)) = \mathcal{X}_i. \tag{3.16}$$

Através das Equações 3.15 e 3.16, pode-se contornar o problema da variação do cálculo da distância de projeto de um código BCH com respeito ao polinômio primitivo usado no cômputo da extensão de anel. A metodologia é a seguinte:

1. Usando a Equação 3.12 e o mapeamento Υ, obtido no cômputo dos polinômios minimais; encontre $\bigcup_{f_i(x) | s(x)} \mathcal{X}_i$.
2. Identifique a distância de projeto para $\bigcup_{f_i(x) | s(x)} \mathcal{X}_i$, usando o Teorema 3.1.
3. Para todo $j \in (\mathbb{Z}_n)^*$ (i.e., $\forall j$ tal que $1 < j < n$ e $\gcd(j, n) = 1$), determine o mapeamento Υ_j a partir da função Υ e a Equação 3.15.
4. Com o mapeamento Υ_j, encontre $\bigcup_{f_i(x) | s(x)} \mathcal{X}_{(i \cdot j)_n}$.
5. Identifique a distância de projeto para $\bigcup_{f_i(x) | s(x)} \mathcal{X}_{(i \cdot j)_n}$, usando o Teorema 3.1.
6. Termine o algoritmo se todos os possíveis j's foram esgotados e retorne a distância de projeto de $\mathcal{C}$ como a máxima distância de projeto obtida durante as iterações.

A metodologia anterior aplica o Teorema 3.1 a cada uma das unidades de $(\mathbb{Z}_n)^*$. Porém, a seguir, mostra-se que nem todas as unidades devem ser consideradas e, portanto, a quantidade de iterações pode ser reduzida.

Lema 3.4. *Se* $\gcd(j,n) = 1$ *e* $j \in \mathcal{X}_1$, *segue que* $\mathcal{X}_{(i\cdot j)_n} = \mathcal{X}_i$ *para todo* i. *Portanto,* $\Upsilon = \Upsilon_j$.

Demonstração: Dado que $\gcd(j,n) = 1$ e $j \in \mathcal{X}_1$, então, $j = (1\cdot q^k)_n = (q^k)_n$ para algum $k < r_1 - 1$. Logo:

$$\begin{aligned}
\mathcal{X}_{(i\cdot j)_n} &= \Big\{\Big((j\cdot i)_n, (j\cdot i\cdot q)_n, (j\cdot i\cdot q^2), \ldots, (j\cdot i\cdot q^{r_i-1})_n\Big)\Big\}\\
\mathcal{X}_{(i\cdot j)_n} &= \Big\{\Big((q^k\cdot i)_n, (q^k\cdot i\cdot q)_n, (q^k\cdot i\cdot q^2), \ldots, (q^k\cdot i\cdot q^{r_i-1})_n\Big)\Big\}\\
\mathcal{X}_{(i\cdot j)_n} &= \Big\{\Big((q^k\cdot i)_n, (q^{k+1}\cdot i)_n, (q^{k+2}\cdot i), \ldots, (q^{k+r_i-1}\cdot i)_n\Big)\Big\}\\
\mathcal{X}_{(i\cdot j)_n} &= \Big\{\Big((q^k\cdot i)_n, (q^{k+1}\cdot i)_n, (q^{k+2}\cdot i), \ldots, (q^{k-1+r_i}\cdot i)_n\Big)\Big\}\\
\mathcal{X}_{(i\cdot j)_n} &= \Big\{\Big((i)_n, (i\cdot q), (i\cdot q^2), \ldots, (i\cdot q^{k-1})_n, (i\cdot q^k)_n, \ldots, (i\cdot q^{r_i-1})_n\Big)\Big\}\\
\mathcal{X}_{(i\cdot j)_n} &= \mathcal{X}_i,
\end{aligned}$$

onde $(i\cdot q^{r_i})_n = (i)_n$, segundo a Equação 3.5. ∎

Lema 3.5. *Dada uma sequência* $s(x)$, *se* δ *é a distância de projeto quando se aplica o Teorema 3.1 em* $\bigcup_{f_i(x)|s(x)} \mathcal{X}_i$, *então a distância de projeto é também* δ *quando se aplica o Teorema em* $\bigcup_{f_{(-i)_n}(x)|s(x)} \mathcal{X}_{(-i)_n}$. *Isto é, a distância de projeto é uma função par que depende dos* i*'s.*

Demonstração: Dado que o Teorema 3.1, aplicado em $\bigcup_{f_i(x)|s(x)} \mathcal{X}_i$, resulta em uma distância de projeto δ, segue que existe uma sequência consecutiva de raízes da forma $\{(a)_n, (a+1)_n, \ldots, (a+\delta-2)_n\}$. Quando se considera $\bigcup_{f_{(-i)_n}(x)|s(x)} \mathcal{X}_{(-i)_n}$, a sequência de raízes consecutivas se torna: $\{(-a-\delta+2)_n, \ldots, (-a-1)_n, (-a)_n\}$. Os números dentro da sequência continuam sendo consecutivos e a cardinalidade não muda. Portanto, a distância de projeto continua invariante. Veja que $(-1)_n = (n-1)_n$ e $\gcd(n-1,n) = 1$. ∎

Os Lemas 3.4 e 3.5 reduzem a quantidade de unidades de $\mathbb{Z}_n$ que devem ser consideradas. Veja o Exemplo 3.6.

Exemplo 3.6. *Considerando o Exemplo 3.1, as unidades de* $\mathbb{Z}_{63}$ *são:*

$$\begin{aligned}
(\mathbb{Z}_{63})^* = \{&1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20, 22, 23, 25, 26, 29, 31, 32, 34, 37, 38, 40,\\
&41, 43, 44, 46, 47, 50, 52, 53, 55, 58, 59, 61, 62\}.
\end{aligned}$$

Segundo a metodologia introduzida anteriormente, para cada uma das unidades j *em* $(\mathbb{Z}_{63})^*$ *deve-se computar* Υ_j *e* $\bigcup_{f_i(x)|s(x)} \mathcal{X}_{(i\cdot j)_n}$ *para depois calcular a distância de projeto.*

Aplicando o Lema 3.4, quando se considera Υ_1 ($j = 1$), não é mais necessário considerar $j \in \{2, 4, 8, 16, 32\}$ porque $\mathcal{X}_{(i \cdot j)_n} = \mathcal{X}_i$, $\forall i$, e, portanto, $\Upsilon_1 = \Upsilon_2 = \Upsilon_4 = \Upsilon_8 = \Upsilon_{16} = \Upsilon_{32}$. Aplicando o Lema 3.5, quando se considera Υ_1, não é mais necessário considerar $j \in \{62, 61, 59, 55, 47, 31\}$ porque a mesma distância de projeto BCH é obtida. Portanto, o próximo $j \in (\mathbb{Z}_{63})^*$ a ser considerado deve ser escolhido entre:

$$\{5, 10, 11, 13, 17, 19, 20, 22, 23, 25, 26, 29, 34, 37, 38, 40, 41, 43, 44, 46, 50, 52, 53, 58\}.$$

Se o seguinte número a ser considerado é 5 ($j = 5$), já não é mais necessário usar $j \in \{10, 20, 40, 17, 34, 58, 53, 43, 23, 46, 29\}$ pelos Lemas 3.4 e 3.5. Portanto, o seguinte $j \in (\mathbb{Z}_{63})^*$ a ser considerado deve ser escolhido entre:

$$\{11, 13, 19, 22, 25, 26, 37, 38, 41, 44, 50, 52, \}.$$

Se o seguinte número a ser considerado é 11 ($j = 11$), o algoritmo esgota o conjunto $(\mathbb{Z}_{63})^*$ porque já não é mais necessário utilizar $j \in \{11, 22, 44, 25, 50, 37, 52, 41, 19, 38, 13, 26\}$ pelos Lemas 3.4 e 3.5. Assim, foram necessárias três iterações para analisar o critério BCH $\forall j \in (\mathbb{Z}_{63})^*$ e, portanto, foram considerados todos os possíveis polinômios primitivos.

O Corolário 9.1 do livro (PETERSON; WELDON, 1972) estabelece que: *Um código cíclico com raízes* $\alpha^e, \alpha^{e+j}, \ldots, \alpha^e + j(\delta - 2)$ *e possivelmente outras, onde* α *é um elemento de ordem* n*, tem distância mínima* δ *ou maior, dado que* $\gcd(j, n) = 1$. Este Corolário justifica a busca de j entre as unidades de $\mathbb{Z}_n$ para encontrar a maior distância de projeto.

3.4 Considerações Finais

Neste capítulo demonstrou-se a relação que existe entre a busca de j com o uso de diferentes polinômios primitivos na construção da extensão de anel ou corpo (dependendo de $\mathbb{A}$) para encontrar a maior distância de projeto de um código BCH com polinômio gerador $g(x)$ e foram apresentados os Lemas 3.4 e 3.5 que permitem realizar uma busca eficiente entre as unidades de $\mathbb{Z}_n$. Também estabeleceu-se uma metodologia para encontrar o polinômio gerador do código BCH de comprimento n e com a maior distância de projeto tal que $s(x)$, uma sequência dada, pertence a esse código. Por último, define-se o algoritmo $\mathcal{C} = \text{BCH\_One\_Seq}(s, PolsMinimais, CyclotomicCosets)$ como mostra na Figura 18. Este algoritmo foi implementado na linguagem Python3 e todas as ferramentas matemáticas, operações nos anéis e corpos, incluindo os estendidos, foram implementadas na linguagem Python3.

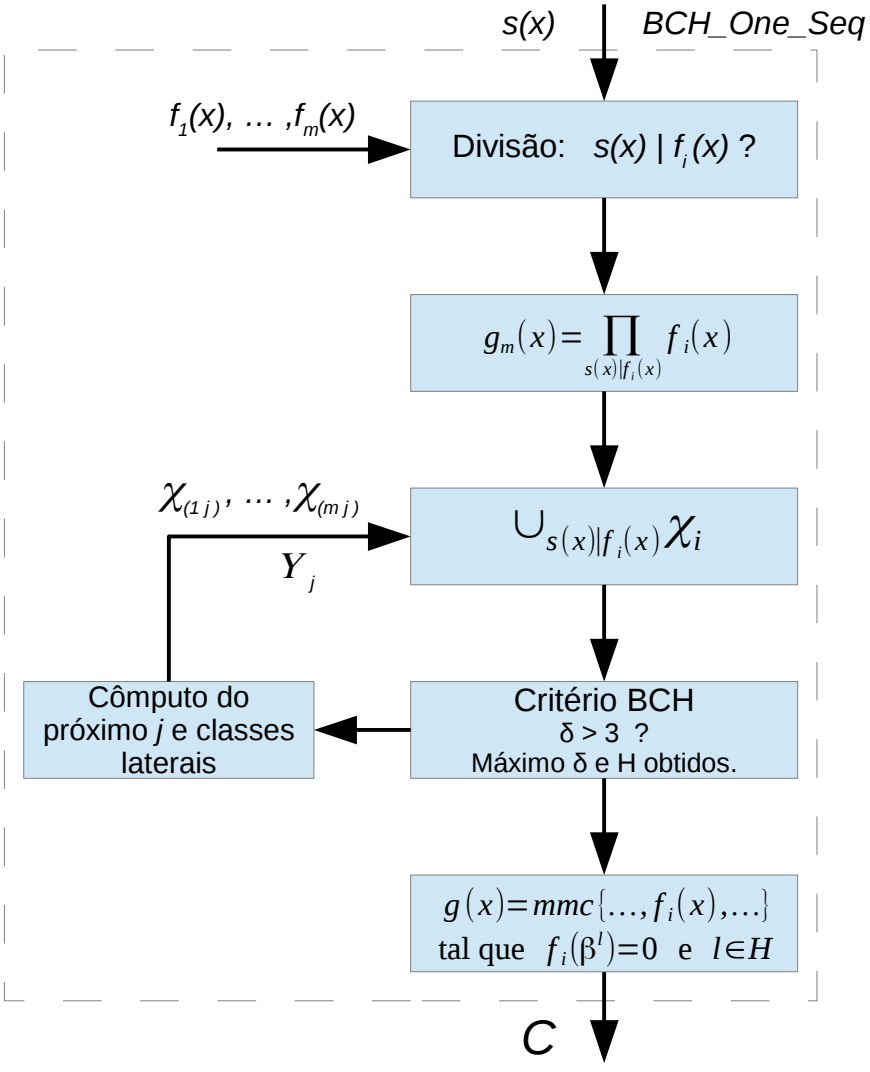


Figura 18 – Diagrama de blocos completo do algoritmo BCH_One_Seq.

4 Identificação de Sequências mRNA através de Códigos BCH

Nos trabalhos (FARIA, 2011), (ROCHA, 2010) e publicações recentes (ROCHA *et al.*, 2010; FARIA *et al.*, 2010; FARIA *et al.*, 2014; BRANDÃO *et al.*, 2015) derivadas desses trabalhos, utiliza-se um modelo para a síntese de proteínas e codificação genética baseada num sistema de comunicação digital. De acordo com o modelo proposto, o sistema de informação genético é composto por um *codificador genético*, seguido do ribossomo, os quais agem de maneira análoga ao codificador de canal e o modulador num sistema tradicional de transmissão digital. Nesses trabalhos, usam-se códigos nsBCH para modelar o codificador genético e verificar a factibilidade do modelo proposto. No desenvolvimento desses trabalhos alguns fatos e propriedades foram descobertos, mas nenhuma explicação foi dada. Neste capítulo, demonstram-se através de fatos matemáticos as propriedades descobertas nesses trabalhos e conclui-se que pelo fato de ter modelado o *codificador genético* como um código nsBCH essas propriedades devem ser satisfeitas. Adicionalmente, através das provas, encontra-se que estas propriedades também são satisfeitas quando os códigos BCH são considerados como o codificador genético. Portanto, neste capítulo propõe-se um algoritmo para identificar sequências de nucleotídeos (mRNA, DNA, entre outras) como palavras-código de códigos BCH. Algumas das propriedades também são satisfeitas e demonstradas para códigos cíclicos.

Este capítulo está organizado da seguinte maneira. Na Seção 4.1, detalha-se o modelo usado durante este capítulo para a síntese de proteínas; na Seção 4.2, apresentam-se os 24 possíveis rotulamentos e as subclasses de rotulamentos (rótulos) para os alfabetos $\mathbb{Z}_4$ e $\mathbb{F}_4$; na Seção 4.3, demonstram-se as propriedades descobertas nos trabalhos citados anteriormente; na Seção 4.4, introduz-se o algoritmo usado para identificar sequências mRNA como palavras-código de códigos BCH; e na Seção 4.5, mostram-se algumas sequências mRNA identificadas de comprimento 51, 63 e 93.

4.1 Modelo: Codificador BCH e Modulador de Sequências mRNA

O modelo para a síntese de proteínas e codificação genética baseada num sistema de transmissão digital desenvolvido em (FARIA, 2011) e (ROCHA, 2010) é composto por um *codificador genético*, seguido do ribossomo (*modulador genético*) como mostra a Figura 19. Estes blocos agem de maneira análoga ao codificador de canal e o modulador num sistema tradicional de comunicação digital (ver Figura 20). Note a similaridade entre os dois diagramas de blocos.

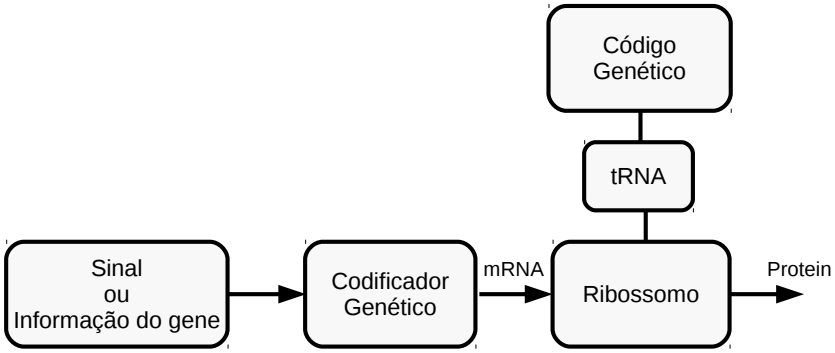


Figura 19 – Modelo para a síntese de proteínas (FARIA *et al.*, 2014).

A informação genética está representada através dos genes ou mesmo através de sinais recebidos no núcleo desde o citoplasma. Esta informação é armazenada como sequências de nucleotídeos, as quais possuem certa redundância uma vez que o sistema biológico é capaz de tolerar certos padrões de erros que podem ocorrer no armazenamento. Uma classe de sequências de nucleotídeos consiste de sequências mRNA, as quais são traduzidas pelo ribossomo em proteínas através das regras definidas pelo código genético. Este último passo é equivalente a realizar uma modulação num sistema de comunicação digital.

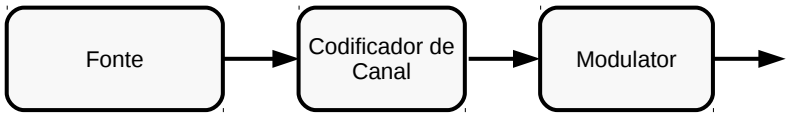


Figura 20 – Transmissor num sistema de comunicação digital tradicional.

Para verificar a factibilidade do modelo proposto, em (FARIA, 2011; ROCHA, 2010) se usou este modelo, junto com códigos corretores de erros da classe nsBCH como *codificador genético*, para representar sequências mRNA e proteínas e poder inferir sobre a informação biológica a partir deste modelo (BRANDÃO *et al.*, 2015). Aqui surge a ideia de considerar códigos BCH de maneira geral como *codificador genético* para generalizar os resultados já obtidos.

4.2 Rotulamentos: $\mathbb{Z}_4$ e $\mathbb{F}_4$

Os códigos corretores de erros precisam ter uma estrutura matemática bem definida e conhecida para poder codificar e decodificar mensagens de maneira sistemática. Sabendo que as sequências mRNA são sequências sobre o alfabeto $\mathbb{N} = \{A, C, G, T\}$ e que os únicos alfabetos de cardinalidade 4, sobre os quais se conhecem técnicas para o projeto de códigos corretores de erros, são $\mathbb{Z}_4$ e $\mathbb{F}_4$; então, deve-se estabelecer um mapa (rotulamento) entre $\mathbb{N}$ e $\mathbb{Z}_4$ ou $\mathbb{F}_4$.

Existem $4! = 24$ possíveis rotulamentos, isto é, 24 diferentes maneiras de representar cada um dos nucleotídeos como um elemento de $\mathbb{Z}_4$ ou $\mathbb{F}_4$. Cada um dos rotulamentos pode ser representado como uma possível permutação do grupo S_4. A seguir, estudam-se os possíveis rotulamentos para cada um dos alfabetos $\mathbb{Z}_4$ e $\mathbb{F}_4$.

4.2.1 Rotulamentos sobre $\mathbb{Z}_4$

Da mesma maneira como foi feito nos trabalhos (FARIA, 2011) e (ROCHA, 2010), todos os possíveis rotulamentos são considerados, dado que nenhuma informação a priori é conhecida. Nestes trabalhos, foi descoberto empiricamente que os rotulamentos podem ser agrupados em três diferentes subgrupos; os quais são denotados como Rótulo **A**, Rótulo **B** e Rótulo **C**. Cada subgrupo indica que os mesmos resultados são obtidos quando usados Rotulamentos que pertencem ao mesmo subgrupo.

Na Seção 4.3, será demonstrado que quando são considerados códigos nsBCH, existem unicamente três subgrupos de rotulamentos. Além disso, será provado que quando usados códigos cíclicos, classe de códigos que inclui os códigos BCH, existem 12 possíveis subgrupos e que, dependendo da situação, podem ser reduzidos nos Rótulos A, B e C.

Os subgrupos identificados pelos Rótulos A, B e C podem ser representados através das simetrias do quadrado, como se mostra na Tabela 7. Note também que os subgrupos podem ser obtidos através de operações simples. Sejam $e_1 = 0132$, $e_2 = 0123$ e $e_3 = 0213$ os três rotulamentos canônicos. Os Rótulos são obtidos ao multiplicar por 3 o vetor e_i (reflexão) e ao somar 1 ou 2 ou 3 a cada uma das posições do vetor e_i (rotações), como se mostra na Tabela 8.

Nos trabalhos (FARIA, 2011) e (ROCHA, 2010), cada um dos Rotulamentos A, B e C foram denominados como $\mathbb{Z}_4$**-linear**, $\mathbb{Z}_2 \times \mathbb{Z}_2$**-linear** e **Klein-linear**, respectivamente. O rotulamento $\mathbb{Z}_4$-linear respeita a complementaridade biológica pelo fato que, em todos os 8 rótulos agrupados, qualquer um dos nucleotídeos necessita caminhar duas arestas para alcançar o seu complementar (A-U e C-G). Esta característica não se encontra nos Rotulamentos B e C.

4.2.2 Rotulamentos sobre $\mathbb{F}_4$

Da mesma maneira como foi feito nos trabalhos (FARIA, 2011) e (ROCHA, 2010), todos os possíveis rotulamentos são considerados. Nestes trabalhos, foi descoberto empiricamente que os resultados obtidos são invariantes à escolha do rotulamento, isto é, basta considerar um único rotulamento dos 24 possíveis rotulamentos porque os mesmos resultados são obtidos.

Na Seção 4.3, será demonstrado que quando são considerados códigos nsBCH, os resultados são invariantes à escolha do rotulamento. Além disso, será provado que

Tabela 7 – Simetrias do quadrado e Rótulos entre $\mathbb{Z}_4$ e $\mathbb{N} = \{A, C, G, T\}$

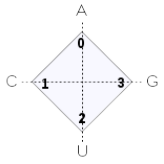
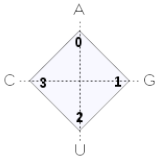
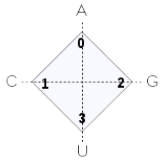
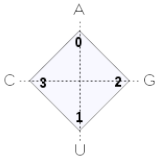
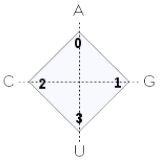
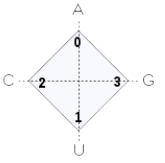
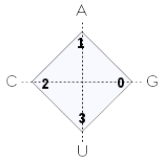
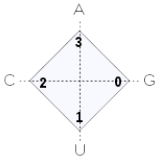
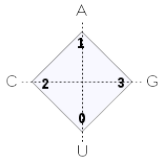
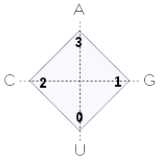
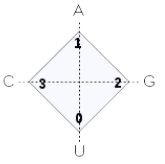
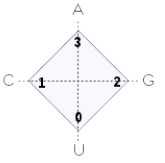
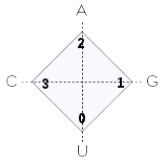
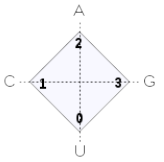
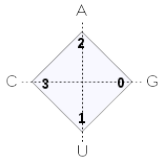
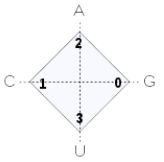
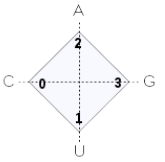
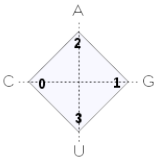
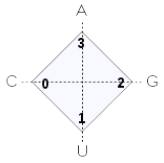
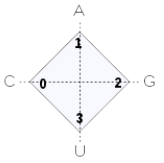
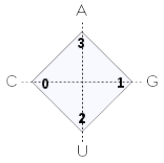
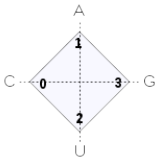
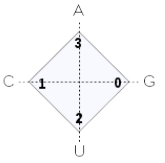
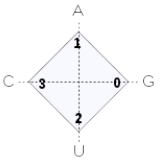
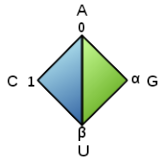
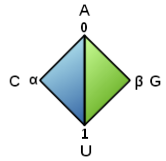
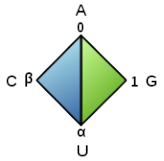
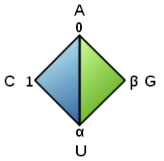
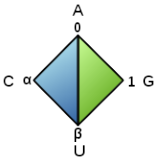
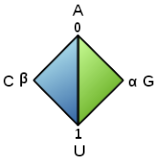
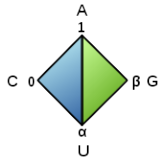
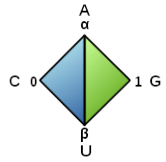
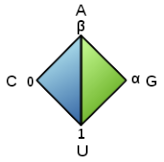
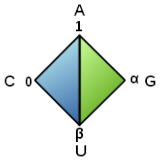
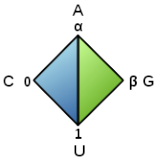
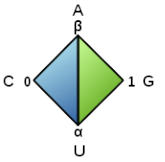
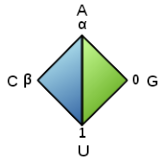
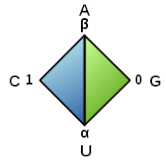
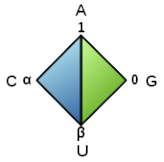
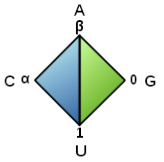
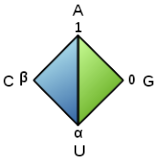
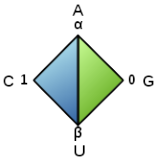
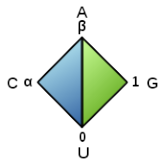
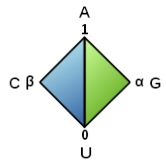
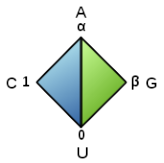
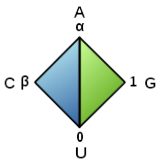
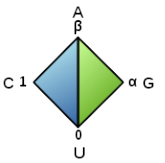
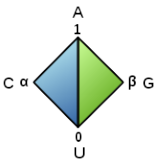
	Rotulo **A**		Rotulo **B**		Rotulo **C**	
0						
	e_1	$3 \cdot e_1$	e_2	$3 \cdot e_2$	e_3	$3 \cdot e_3$
1						
	$(e_1 + 1)$	$3 \cdot (e_1 + 1)$	$(e_2 + 1)$	$3 \cdot (e_2 + 1)$	$(e_3 + 1)$	$3 \cdot (e_3 + 1)$
2						
	$(e_1 + 2)$	$3 \cdot (e_1 + 2)$	$(e_2 + 2)$	$3 \cdot (e_2 + 2)$	$(e_3 + 2)$	$3 \cdot (e_3 + 2)$
3						
	$(e_1 + 3)$	$3 \cdot (e_1 + 3)$	$(e_2 + 3)$	$3 \cdot (e_2 + 3)$	$(e_3 + 3)$	$3 \cdot (e_3 + 3)$

Tabela 8 – Rotulamentos entre $\mathbb{Z}_4$ e $\mathbb{N} = \{A, C, G, T\}$

	Rotulo **A**		Rotulo **B**		Rotulo **C**	
0	A C G U 0 1 3 2	A C G U 0 3 1 2	A C G U 0 1 2 3	A C G U 0 3 2 1	A C G U 0 2 1 3	A C G U 0 2 3 1
	e_1	$3 \cdot e_1$	e_2	$3 \cdot e_2$	e_3	$3 \cdot e_3$
1	A C G U 1 2 0 3	A C G U 3 2 0 1	A C G U 1 2 3 0	A C G U 3 2 1 0	A C G U 1 3 2 0	A C G U 3 1 2 0
	$(e_1 + 1)$	$3 \cdot (e_1 + 1)$	$(e_2 + 1)$	$3 \cdot (e_2 + 1)$	$(e_3 + 1)$	$3 \cdot (e_3 + 1)$
2	A C G U 2 3 1 0	A C G U 2 1 3 0	A C G U 2 3 0 1	A C G U 2 1 0 3	A C G U 2 0 3 1	A C G U 2 0 1 3
	$(e_1 + 2)$	$3 \cdot (e_1 + 2)$	$(e_2 + 2)$	$3 \cdot (e_2 + 2)$	$(e_3 + 2)$	$3 \cdot (e_3 + 2)$
3	A C G U 3 0 2 1	A C G U 1 0 2 3	A C G U 3 0 1 2	A C G U 1 0 3 2	A C G U 3 1 0 2	A C G U 1 3 0 2
	$(e_1 + 3)$	$3 \cdot (e_1 + 3)$	$(e_2 + 3)$	$3 \cdot (e_2 + 3)$	$(e_3 + 3)$	$3 \cdot (e_3 + 3)$

Tabela 9 – Simetrias do tetraedro e Rótulos entre $\mathbb{F}_4$ e $\mathbb{N} = \{A, C, G, T\}$

	Rótulo **A**			Rótulo **B**		
0	 e	 $e \cdot \alpha$	 $e \cdot \beta$	 $\overline{e}$	 $\overline{e} \cdot \alpha$	 $\overline{e} \cdot \beta$
1	 $(1+e)$	 $(1+e) \cdot \alpha$	 $(1+e) \cdot \beta$	 $(1+\overline{e})$	 $(1+\overline{e}) \cdot \alpha$	 $(1+\overline{e}) \cdot \beta$
2	 $(\alpha+e)$	 $(\alpha+e) \cdot \alpha$	 $(\alpha+e) \cdot \beta$	 $(\beta+\overline{e})$	 $(\beta+\overline{e}) \cdot \alpha$	 $(\beta+\overline{e}) \cdot \beta$
3	 $(\beta+e)$	 $(\beta+e) \cdot \alpha$	 $(\beta+e) \cdot \beta$	 $(\alpha+\overline{e})$	 $(\alpha+\overline{e}) \cdot \alpha$	 $(\alpha+\overline{e}) \cdot \beta$

quando são usados códigos cíclicos existem 8 possíveis subgrupos, os quais, dependendo da situação, podem ser reduzidos a um único grupo.

Os subgrupos identificados pelos Rótulos A e B podem ser representados através das simetrias do tetraedro, como se mostra na Tabela 9, onde o Rótulo A representa as simetrias pares do tetraedro e o Rótulo B as ímpares. Note também que os subgrupos podem ser obtidos através de operações simples. Se $e = 01\alpha\beta$ é o rotulamento canônico, os Rótulos são obtidos ao computar o conjugado $\bar{e}$ (trocar α por β), ao multiplicar por α ou β o vetor e (rotações com o vértice 0 fixo) e ao somar 1 ou α ou β em cada uma das posições do vetor e (as três diferentes reflexões), como se mostra na Tabela 10.

4.3 Propriedades e Teoremas na Identificação de Sequências

Nesta seção, demonstram-se as propriedades e fatos que foram descobertos em (FARIA, 2011) e (ROCHA, 2010) e que não foram provados; assim, também, apresentam-se generalizações dessas descobertas. Estas propriedades serão exploradas na implementação do algoritmo de identificação de sequências mRNA como palavras-código de códigos corretores de erros.

Tabela 10 – Rotulamentos entre $\mathbb{F}_4$ e $\mathbb{N} = \{A, C, G, T\}$

	Rótulo **A**			Rótulo **B**		
0	A C G U 0 1 α β e	A C G U 0 α β 1 $e \cdot \alpha$	A C G U 0 β 1 α $e \cdot \beta$	A C G U 0 1 β α $\overline{e}$	A C G U 0 α 1 β $\overline{e} \cdot \alpha$	A C G U 0 β α 1 $\overline{e} \cdot \beta$
1	A C G U 1 0 β α $(1+e)$	A C G U α 0 1 β $(1+e) \cdot \alpha$	A C G U β 0 α 1 $(1+e) \cdot \beta$	A C G U 1 0 α β $(1+\overline{e})$	A C G U α 0 β 1 $(1+\overline{e}) \cdot \alpha$	A C G U β 0 1 α $(1+\overline{e}) \cdot \beta$
2	A C G U α β 0 1 $(\alpha+e)$	A C G U β 1 0 α $(\alpha+e) \cdot \alpha$	A C G U 1 α 0 β $(\alpha+e) \cdot \beta$	A C G U β α 0 1 $(\beta+\overline{e})$	A C G U 1 β 0 α $(\beta+\overline{e}) \cdot \alpha$	A C G U α 1 0 β $(\beta+\overline{e}) \cdot \beta$
3	A C G U β α 1 0 $(\beta+e)$	A C G U 1 β α 0 $(\beta+e) \cdot \alpha$	A C G U α 1 β 0 $(\beta+e) \cdot \beta$	A C G U α β 1 0 $(\alpha+\overline{e})$	A C G U β 1 α 0 $(\alpha+\overline{e}) \cdot \alpha$	A C G U 1 α β 0 $(\alpha+\overline{e}) \cdot \beta$

4.3.1 Subgrupos simétricos de rótulos em $\mathbb{Z}_4$

Ao estudar a Tabela 8, observa-se que os subgrupos de rótulos, identificados por A_0, B_0 e C_0 (linhas da tabela), são obtidos pela multiplicação do rotulamento líder com cada uma das unidades do anel $\mathbb{Z}_4$.

Assim, denote a sequência de nucleotídeos a ser analisada como nuc e a sequência sobre $\mathbb{Z}_4$, obtida através do rótulo e_s, como nuc_{e_s}. Esta notação pode ser estendida para qualquer rótulo. Por exemplo, $nuc_{(2+e_s)3}$ é a sequência sobre $\mathbb{Z}_4$, obtida a partir de nuc, pelo uso do rótulo $(2+e_s)3$, como se ilustra na Tabela 8. Veja que s, somente pode tomar valores entre 1, 2 e 3, o qual representa os rótulos A, B e C, respectivamente.

Afirmação 4.1. *Usando a notação acima, se $nuc_{(i+e_s)j} \in \mathcal{C}$, então $nuc_{(i+e_s)k} \in \mathcal{C}$, onde $k = a \cdot i$, $a \in (\mathbb{Z}_4)^*$ e $\mathcal{C}$ é um código linear.*

Demonstração. Considere a sequência $nuc_{(i+e_s)j}$ na forma polinomial:

$$nuc_{(i+e_s)j} = b_0 + b_1 x + \cdots + b_{n-1} x^{n-1} \in \mathcal{C}.$$

Sabe-se que a relação entre os rótulos $nuc_{(i+e_s)j}$ e $nuc_{(i+e_s)k}$ é uma multiplicação por a. Assim, $nuc_{(i+e_s)k}$ pode ser escrito como:

$$\begin{aligned} nuc_{(i+e_s)k} =& ab_0 + ab_1 x + \cdots + ab_{n-1} x^{n-1} \\ nuc_{(i+e_s)k} =& a(b_0 + b_1 x + \cdots + b_{n-1} x^{n-1}). \end{aligned}$$

Como o código $\mathcal{C}$ é linear, segue que $nuc_{(i+e_s)k} \in \mathcal{C}$. □

Afirmação 4.2. *Usando a notação acima, se $nuc_{(i+e_s)j} \in \mathcal{C}$, onde $\mathcal{C}$ é um código cíclico gerado por um polinômio $f(x)$ ($\mathcal{C} = \langle f(x) \rangle$) tal que $f(x) \neq x - 1$, então $nuc_{(k+e_s)j} \in \mathcal{C}$, onde $k = a + i$, $a \in \mathbb{Z}_4$.*

Demonstração. Considere a sequência $nuc_{(i+e_s)j}$ na forma polinomial:

$$nuc_{(i+e_s)j} = b_0 + b_1x + \cdots + b_{n-1}x^{n-1} \in \mathcal{C}.$$

Sabe-se que a relação entre os rótulos $nuc_{(i+e_s)j}$ e $nuc_{(k+e_s)j}$ é uma soma ($k = a + i$). Assim $nuc_{(k+e_s)i}$ pode ser escrito como:

$$\begin{aligned} nuc_{(k+e_s)j} =& (a + b_0) + (a + b_1)x + \cdots + (a + b_{n-1})x^{n-1} \\ nuc_{(k+e_s)j} =& (1 + x + \cdots + x^{n-1}) + (b_0 + b_1x + \cdots + b_{n-1}x^{n-1}). \end{aligned}$$

Como o código $\mathcal{C}$ é cíclico, segue que $f(x)|(x^n - 1)$ e que o polinômio $x^n - 1$ pode ser sempre fatorado da seguinte maneira:

$$x^n - 1 = (1 + x + x^2 + \cdots + x^{n-1})(x - 1).$$

Portanto, se $f(x)$ for diferente de $x - 1$, então:

$$f(x)|(1 + x + \cdots + x^{n-1}) \text{ e } (1 + x + \cdots + x^{n-1}) \in \mathcal{C} = \langle f(x) \rangle.$$

Assim, $nuc_{(k+e_s)j} \in \mathcal{C}$. □

Utilizando as Afirmações 4.1 e 4.2, prova-se a existência única dos rótulos A, B e C quando o polinômio gerador do código cíclico não é composto pelo polinômio $x-1$. Este é o caso dos resultados apresentadas em (FARIA, 2011) e (ROCHA, 2010) e de trabalhos recentes derivados destas resultados. Nestes trabalhos, somente foram considerados códigos nsBCH, nos quais o polinômio $x - 1$ não é considerado.

A seguir, estuda-se a influência do polinômio $x - 1$ na identificação de sequências de nucleotídeos e na subdivisão dos rótulos A, B e C.

Ao realizar a divisão polinomial de $nuc(x) = b_0 + b_1x + \cdots + b_{n-1}x^{n-1}$ com $x - 1$ sobre o anel $\mathbb{Z}_4$, obtém-se: $nuc(x) = q(x)(x - 1) + r(x)$, onde o grau de $r(x)$ é menor que 1, isto é, $r(x) \in \mathbb{Z}_4$. Ao aplicar o algoritmo da divisão, demonstra-se que:

$$r(x) = \sum_{i=0}^{n-1} b_i, \quad \text{onde as somas são feitas no anel } \mathbb{Z}_4.$$

Portanto, o polinômio $x - 1$ divide o polinômio $nuc(x)$ se, e somente se, $r(x) = 0$, isto é, o polinômio $x - 1$ faz parte do código cíclico $\mathcal{C}$ que identifica a sequência de nucleotídeos $nuc(x)$ sobre algum determinado rotulamento se, e somente se, a soma de todos os coeficientes sobre $\mathbb{Z}_4$ é zero.

Analisando o polinômio $x - 1$, obtêm-se as quatro subdivisões para cada um dos rótulos A, B e C.

Afirmação 4.3. *Considere uma sequência de nucleotídeos, a qual é representada através dos seguintes quatro rótulos:* e_s, (e_s+1), (e_s+2) *e* (e_s+3). *Assim, obtêm-se quatro polinômios sobre* $\mathbb{Z}_4$*:* $nuc(x)_{e_s}$, $nuc(x)_{(e_s+1)}$, $nuc(x)_{(e_s+2)}$ *e* $nuc(x)_{(e_s+3)}$. *Somente um único polinômio dentre os polinômios* $nuc(x)_{e_s}$, $nuc(x)_{(e_s+1)}$, $nuc(x)_{(e_s+2)}$ *e* $nuc(x)_{(e_s+3)}$ *é divisível pelo polinômio* $x-1$.

Demonstração. Considerando $nuc(x)_{e_s} = b_0 + b_1x + \cdots + b_{n-1}x^{n-1}$; obtêm-se:

- $nuc(x)_{(e_s+1)} = (b_0+1) + (b_1+1)x + \cdots + (b_{n-1}+1)x^{n-1}$
- $nuc(x)_{(e_s+2)} = (b_0+2) + (b_1+2)x + \cdots + (b_{n-1}+2)x^{n-1}$
- $nuc(x)_{(e_s+3)} = (b_0+3) + (b_1+3)x + \cdots + (b_{n-1}+3)x^{n-1}$.

Ao computar o resto da divisão com cada um dos polinômios por $x-1$, denotados por $r(x)_{e_s}$, $r(x)_{(e_s+1)}$, $r(x)_{(e_s+2)}$ e $r(x)_{(e_s+3)}$, obtêm-se:

- $r(x)_{(e_s+0)} = b_0 + b_1 + \cdots + b_{n-1} = \sum_{l=0}^{n-1} b_l = (\sigma)_4$
- $r(x)_{(e_s+1)} = (b_0+1) + (b_1+1) + \cdots + (b_{n-1}+1) = \sum_{l=0}^{n-1} (b_l+1) = (\sigma + 1\cdot n)_4$
- $r(x)_{(e_s+2)} = (b_0+2) + (b_1+2) + \cdots + (b_{n-1}+2) = \sum_{l=0}^{n-1} (b_l+2) = (\sigma + 2\cdot n)_4$
- $r(x)_{(e_s+3)} = (b_0+3) + (b_1+3) + \cdots + (b_{n-1}+3) = \sum_{l=0}^{n-1} (b_l+3) = (\sigma + 3\cdot n)_4$.

Para que $\frac{\mathbb{Z}_4[x]}{\langle x^n-1\rangle}$ seja um anel de ideais principais e para que x^n-1 seja fatorado unicamente por polinômios minimais é necessário que n seja ímpar, que são características importantes para o projeto de códigos cíclicos, em especial para a decodificação. Assim, n pode ser expressado como $n = 2q+1$, para algum q inteiro positivo. Logo: $(n)_4 = 1$ ou $(n)_4 = 3$. Assim,

- $r(x)_{(e_s+0)} = (\sigma)_4$
- $r(x)_{(e_s+1)} = (\sigma)_4 + 1$ ou $r(x)_{(e_s+1)} = (\sigma)_4 + 3$
- $r(x)_{(e_s+2)} = (\sigma)_4 + 2$ ou $r(x)_{(e_s+2)} = (\sigma)_4 + 2$
- $r(x)_{(e_s+3)} = (\sigma)_4 + 3$ ou $r(x)_{(e_s+3)} = (\sigma)_4 + 1$.

Observe que: $(\sigma)_4 = 0$ ou $(\sigma)_4 = 1$ ou $(\sigma)_4 = 2$ ou $(\sigma)_4 = 3$, mas é um valor único (as conjunções são exclusivas). Portanto, se $(\sigma)_4 = 0$ então somente $r(x)_{(e_s+0)} = 0$, se $(\sigma)_4 = 1$ então somente $r(x)_{(e_s+3)} = 0$, se $(\sigma)_4 = 2$ então somente $r(x)_{(e_s+2)} = 0$ e se $(\sigma)_4 = 3$ então somente $r(x)_{(e_s+1)} = 0$. □

De acordo com as Afirmações 4.1, 4.2 e 4.3, demonstra-se que quando usados códigos cíclicos para representar sequências de nucleotídeos, como por exemplo: DNA, mRNA, microRNA, etc, existem 12 possíveis subclasses de Rótulos; os quais são: A_1, A_2, A_3, A_4, B_1, B_2, B_3, B_4, C_1, C_2, C_3 e C_4.

As subclasses denotadas por A, B e C, representam os rotulamentos $\mathbb{Z}_4$**-linear**, $\mathbb{Z}_2 \times \mathbb{Z}_2$**-linear** e **Klein-linear**, respectivamente; os quais não parecem estar relacionados entre si. Para cada uma das subclasses A, B e C, existe uma subdivisão em 4 subclasses: 1, 2, 3 e 4; a qual indica para qual dos 4 possíveis rotulamentos o polinômio obtido, após o mapeamento para $\mathbb{Z}_4$, é divisível pelo polinômio $x-1$, i.e., para qual dos 4 possíveis rotulamentos a paridade é zero. Assim, observa-se que existe uma relação entre os rótulos A_1, A_2, A_3 e A_4 e o algoritmo que será mostrado na Seção 4.4. A relação é que os polinômios sobre $\mathbb{Z}_4$ obtidos após o mapeamento através dos Rótulos A_1, A_2, A_3 e A_4 são divididos pelos mesmos polinômios minimais e o polinômio $x-1$ sempre divide unicamente um desses polinômios, i.e., somente um desses polinômios tem paridade zero. Este fato também ocorre para os Rótulos B_1, B_2, B_3, B_4, C_1, C_2, C_3 e C_4.

Como mencionado anteriormente, quando os códigos cíclicos "*narrow sense*" BCH são usados, o polinômio $x-1$ não é considerado pois as subclasses A_1, A_2, A_3 e A_4 levam aos mesmos resultados e podem ser agrupados num único rótulo: Rótulo A. Isto foi observado nos trabalhos (FARIA, 2011) e (ROCHA, 2010) mas não foi dado nenhuma explicação. Lembre que o mesmo fenômeno ocorre para os Rótulos B_1, B_2, B_3 e B_4, e para os Rótulos C_1, C_2, C_3 e C_4.

Por último, nos trabalhos (FARIA, 2011) e (ROCHA, 2010) descobriu-se que os resultados obtidos ao estudar uma sequência de nucleotídeos na direção 5'-3' são os mesmos obtidos ao estudar a sequência complementar, denotada por 3'-5', na direção 5'-3' (anti-paralela). Este fato é relevante quando são estudadas as sequências DNA, onde o processo de replicação é importante. A seguir explica-se brevemente a maneira em que a informação é armazenada no DNA e colocam-se algumas definições que permitirão entender a afirmação que formaliza a descoberta:

- Uma sequência DNA é formada por duas sequências complementares de nucleotídeos, denotadas por: $nuc(x)$ (5'-3') e $cnuc_(x)$ (3'-5'). As sequências são complementares porque $cnuc(x)$ é obtida quando sobre $nuc(x)$ se aplicam as seguintes substituições biológicas: $A \mapsto U$, $U \mapsto A$, $G \mapsto C$ e $C \mapsto G$
- Considera-se que as sequências $nuc(x)$ e $cnuc(x)$ armazenam informação pois as sequências $nuc(x)$ e $x^{n-1}cnuc(x^{-1})$ (sequência $cnuc(x)$ lida na direção oposta ou direção 5'-3') são usadas indiferentemente nos processos de replicação e transcrição.

Afirmação 4.4. *No alfabeto $\mathbb{Z}_4$, se uma dada sequência de nucleotídeos lida na direção 5'-3' e denotada por $nuc(x)$ pertence ao código $\mathcal{C} = \langle g(x) \rangle$ com parâmetros (n, k, d) através*

do Rótulo A_i (ou B_i ou C_i), então a sequência antiparalela de nucleotídeos, denotada por $x^{n-1}cnuc(x^{-1}) = rev(cnuc(x))$[1], pertence ao código $\mathcal{C}_2 = \langle rev(g(x))\rangle$ com parâmetros (n, k, d) através do Rótulo $A_{(i+2)}$ (ou $B_{(i+2)}$ ou $C_{(i+2)}$).

Demonstração. Considere a sequência de nucleotídeos $nuc(x)$ e utilize o rotulamento e_i, que pertence ao Rótulo A_0 ou B_0 ou C_0, para mapeá-la numa sequência sobre $\mathbb{Z}_4$, denotada como $nuc_{e_i}(x) \in \mathbb{Z}_4^n$. Observe, nas Tabelas 7 e 8, que $cnuc_{e_i+2}(x) = nuc_{(e_i+2)}$. Assim, estudar a sequência 5'-3' pelo rotulamento e_i é equivalente a ter estudado a sequência $cnuc_{(e_i+2)} \in \mathbb{Z}_4^n$. Portanto os mesmos códigos BCH são obtidos, i.e., mesmo polinômio gerador $g(x)$. Sejam $\{\alpha^i, \ldots, \alpha^{d-2}\}$ as raízes de $g(x)$, as quais são também raízes de $cnuc_{(e_i+2)}(x)$. Assim, as raízes de rev$(g(x))$ são $\{\alpha^{-i}, \ldots, \alpha^{-(d-2)}\}$, as quais são também raízes de rev$(cnuc_{(e_i+2)}(x))$. Portanto, rev$(cnuc_{(e_i+2)}(x)) \in \mathcal{C}_2$ e $\mathcal{C}_2$ tem parâmetros (n, k, d). □

4.3.2 Subgrupos simétricos de rótulos em $\mathbb{F}_4$

Ao estudar a Tabela 10, observa-se que os subgrupos de rótulos, identificados por A_0 e B_0 (linhas da tabela), são obtidos pela multiplicação do rotulamento líder com cada uma das unidades do corpo $\mathbb{F}_4$.

Assim, denote a sequência de nucleotídeos a ser analisada como nuc e a sequência sobre $\mathbb{F}_4$, obtida através do rótulo e, como nuc_e. Esta notação pode ser estendida para qualquer rótulo. Por exemplo, $nuc_{(\alpha+e)\beta}$ é a sequência sobre $\mathbb{F}_4$, obtida a partir de nuc, pelo uso do rótulo $(\alpha + e)\beta$, como se ilustra na Tabela 10.

Afirmação 4.5. *Usando a notação acima, se $nuc_{(i+e)j} \in \mathcal{C}$, então $nuc_{(i+e)k} \in \mathcal{C}$, onde $k = a \cdot i$ com $a \in (\mathbb{F}_4)^*$ e $\mathcal{C}$ um código linear.*

Demonstração. Considere a sequência $nuc_{(i+e)j}$ na forma polinomial:

$$nuc_{(i+e)j} = b_0 + b_1x + \cdots + b_{n-1}x^{n-1} \in \mathcal{C}$$

Sabe-se que a relação entre os rótulos $nuc_{(i+e)j}$ e $nuc_{(i+e)k}$ é uma multiplicação por a, assim $nuc_{(i+e_s)k}$ pode ser escrito como:

$$\begin{aligned} nuc_{(i+e)k} =& ab_0 + ab_1x + \cdots + ab_{n-1}x^{n-1} \\ nuc_{(i+e)k} =& a(b_0 + b_1x + \cdots + b_{n-1}x^{n-1}). \end{aligned}$$

Como o código $\mathcal{C}$ é linear, segue que $nuc_{(i+e)k} \in \mathcal{C}$ □

Afirmação 4.6. *Usando a notação acima, se $nuc_{(i+e_s)j} \in \mathcal{C}$, onde $\mathcal{C}$ é um código cíclico gerado por um polinômio $f(x)$ ($\mathcal{C} = \langle f(x)\rangle$) tal que $f(x) \neq x - 1$, então $nuc_{(k+e)j} \in \mathcal{C}$, onde $k = a + i$ com $a \in \mathbb{F}_4$.*

[1] O operador rev$(\cdot)$ age de acordo ao definido na Seção 2.3.4

Demonstração. Considere a sequência $nuc_{(i+e)j}$ na forma polinomial:

$$nuc_{(i+e)j} = b_0 + b_1x + \cdots + b_{n-1}x^{n-1} \in \mathcal{C}.$$

Sabe-se que a relação entre os rótulos $nuc_{(i+e)j}$ e $nuc_{(k+e)j}$ é uma soma ($k = a + i$), assim $nuc_{(k+e)i}$ pode ser escrito como:

$$\begin{aligned} nuc_{(k+e)j} =& (a + b_0) + (a + b_1)x + \cdots + (a + b_{n-1})x^{n-1} \\ nuc_{(k+e)j} =& (1 + x + \cdots + x^{n-1}) + (b_0 + b_1x + \cdots + b_{n-1}x^{n-1}). \end{aligned}$$

Como o código $\mathcal{C}$ é cíclico segue que $f(x)|(x^n - 1)$ e que o polinômio $x^n - 1$ pode ser sempre fatorado da seguinte maneira:

$$x^n - 1 = (1 + x + x^2 + \cdots + x^{n-1})(x - 1).$$

Portanto, se $f(x)$ for diferente de $x - 1$, então

$$f(x)|(1 + x + \cdots + x^{n-1}) \text{ e } (1 + x + \cdots + x^{n-1}) \in \mathcal{C} = \langle f(x) \rangle.$$

Assim, $nuc_{(k+e)j} \in \mathcal{C}$. □

Analogamente à Seção 4.3.1, o polinômio $x-1$ também tem um papel muito importante na classificação dos rotulamentos e seu comportamento é similar quando considerado o alfabeto $\mathbb{Z}_4$. Assim, observa-se que os Rótulos A e B podem ser divididos em 4 subgrupos a saber: A_1, A_2, A_3 e A_4; e B_1, B_2, B_3 e B_4, respectivamente. A seguir, estabelece-se e demonstra-se a afirmação que justifica o que foi mencionado anteriormente.

Afirmação 4.7. *Considere uma sequência de nucleotídeos, a qual é representada através dos seguintes quatro rótulos:* e, $(e+1)$, $(e+\alpha)$ *e* $(e+\beta)$. *Assim, obtêm-se quatro polinômios sobre* $\mathbb{F}_4$*:* $nuc(x)_e$, $nuc(x)_{(e+1)}$, $nuc(x)_{(e+\alpha)}$ *e* $nuc(x)_{(e+\beta)}$. *Para* n *ímpar, somente um único polinômio dentre os polinômios* $nuc(x)_e$, $nuc(x)_{(e+1)}$, $nuc(x)_{(e+\alpha)}$ *e* $nuc(x)_{(e+\beta)}$ *é divisível pelo polinômio* $x - 1$.

Demonstração. Considerando $nuc(x)_e = b_0 + b_1x + \cdots + b_{n-1}x^{n-1}$; obtêm-se:

- $nuc(x)_{(e+1)} = (b_0 + 1) + (b_1 + 1)x + \cdots + (b_{n-1} + 1)x^{n-1}$
- $nuc(x)_{(e+\alpha)} = (b_0 + \alpha) + (b_1 + \alpha)x + \cdots + (b_{n-1} + \alpha)x^{n-1}$
- $nuc(x)_{(e+\beta)} = (b_0 + \beta) + (b_1 + \beta)x + \cdots + (b_{n-1} + \beta)x^{n-1}$.

Ao computar o resto da divisão de cada um dos polinômios por $x - 1$, denotados por $r(x)_e$, $r(x)_{(e+1)}$, $r(x)_{(e+\alpha)}$ e $r(x)_{(e+\beta)}$, obtêm-se:

- $r(x)_{(e+0)} = b_0 + \cdots + b_{n-1} = \sum_{l=0}^{n-1} b_l = \sigma = ((\sigma_1)_2, (\sigma_2)_2)$

- $r(x)_{(e+1)} = (b_0 + 1) + \cdots + (b_{n-1} + 1) = \sum_{l=0}^{n-1} (b_l + 1) = ((\sigma_1 + 1)_2, (\sigma_2)_2)$
- $r(x)_{(e+\alpha)} = (b_0 + \alpha) + \cdots + (b_{n-1} + \alpha) = \sum_{l=0}^{n-1} (b_l + \alpha) = ((\sigma_1)_2, (\sigma_2 + 1)_2)$
- $r(x)_{(e+\beta)} = (b_0 + \beta) + \cdots + (b_{n-1} + \beta) = \sum_{l=0}^{n-1} (b_l + 3) = ((\sigma_1 + 1)_2, (\sigma_2 + 1)_2)$.

onde $((\sigma_1)_2, (\sigma_2)_2)$ é a representação em $(\mathbb{F}_2)^2$ de σ. Se n é um número ímpar, então n pode ser expressado como $n = 2q + 1$, para algum q inteiro positivo e $(n)_2 = 1$. Assim:

- $r(x)_{(e+0)} = ((\sigma_1)_2, (\sigma_2)_2)$
- $r(x)_{(e+1)} = ((\sigma_1 + 1)_2, (\sigma_2)_2)$
- $r(x)_{(e+\alpha)} = ((\sigma_1)_2, (\sigma_2 + 1)_2)$
- $r(x)_{(e+\beta)} = ((\sigma_1 + 1)_2, (\sigma_2 + 1)_2)$.

Observe que: $((\sigma_1)_2, (\sigma_2)_2) = (0, 0)$ ou $((\sigma_1)_2, (\sigma_2)_2) = (1, 0)$ ou $((\sigma_1)_2, (\sigma_2)_2) = (0, 1)$ ou $((\sigma_1)_2, (\sigma_2)_2) = (1, 1)$, mas é um valor único (as conjunções são exclusivas). Portanto, se $((\sigma_1)_2, (\sigma_2)_2) = (0, 0)$ então somente $r(x)_{(e+0)} = 0$, se $((\sigma_1)_2, (\sigma_2)_2) = (1, 0)$ então somente $r(x)_{(e+1)} = 0$, se $((\sigma_1)_2, (\sigma_2)_2) = (0, 1)$ então somente $r(x)_{(e+\alpha)} = 0$ e se $((\sigma_1)_2, (\sigma_2)_2) = (1, 1)$ então somente $r(x)_{(e+\beta)} = 0$. □

De acordo com os trabalhos (FARIA, 2011) e (ROCHA, 2010), é necessário somente estudar um único Rótulo porque todos os outros 23 rótulos levam ao mesmo resultado. Porém, esta característica somente pode ser observada quando forem usados códigos nsBCH com distância de projeto igual a 3, como os usados nos trabalhos mencionados. No caso geral, quando forem utilizados códigos BCH para identificar sequências de nucleotídeos existem duas subclasses: A e B, como mostra a Tabela 10. Porém, será provado, a seguir, que esses dois rótulos estão relacionados e, portanto, os resultados obtidos através do rótulo B podem ser reproduzidos quando se usa o rótulo A ao aplicar a transformação conjugado.

Para poder demonstrar a relação entre os rótulos A e B basta aplicar o conjugado, porém, necessitando das afirmações a seguir. Antes porém, observe a relação existente entre os rótulos líderes de A e de B (e e $\overline{e}$, respectivamente).

Definição 4.1. *Considere $\sigma(\cdot)$, o mapa conjugado: $\sigma : \{0 \mapsto 0, 1 \mapsto 1, \alpha \mapsto \beta, \beta \mapsto \alpha\}$, definido como: $\sigma(a) = a^2$, onde $a \in \mathbb{F}_4$.*

Note que:

- $\sigma^2(a) = a$
- $\sigma(e) = \overline{e}$.

Afirmação 4.8. *O operador σ é um homomorfismo de anéis.*

Demonstração.

- $\sigma(a+b) = (a+b)^2 = a^2 + b^2 = \sigma(a) + \sigma(b)$
- $\sigma(a \cdot b) = (a \cdot b)^2 = a^2 \cdot b^2 = \sigma(a) \cdot \sigma(b)$.

□

Afirmação 4.9. *Se $p(x)|g(x)$, então $\sigma(p(x))|\sigma(g(x))$, onde $\sigma(\cdot)$ quando aplicado em polinômios significa que $\sigma(\cdot)$ é aplicado em cada um dos seus coeficientes.*

Demonstração. Como $p(x)|g(x)$, segue que $\exists q(x)$ tal que $p(x)q(x) = g(x)$. Aplicando o homomorfismo: $\sigma(p(x)q(x)) = \sigma(g(x))$. Portanto, $\sigma(p(x))|\sigma(g(x))$. □

Afirmação 4.10. *Sejam $n = 4^s - 1$ para algum l e $p(x) = a_0 + \dots a_{n-1}x^{n-1} \in \mathbb{F}_4[x]$ um polinômio sobre $\mathbb{F}_4$ com raízes: $\{\lambda^{l\cdot 1}, \lambda^{l\cdot 4}, \lambda^{l\cdot 16}, \dots\}$, onde λ satisfaz:*

$$GF(4^s)^* = \{\lambda, \lambda^2, \dots, \lambda^{4^s-1}, \lambda^{4^s} = 1\}.$$

Se as raízes de $p(x)$ são $\{\lambda^l, \lambda^{4\cdot l}, \dots\}$, então, para algum inteiro l, as raízes de $\sigma(p(x)) = \sigma(a_0) + \dots + \sigma(a_{n-1})x^{n-1}$ são:

$$\{(\lambda^l)^2, (\lambda^{l\cdot 4})^2, (\lambda^{l\cdot 16})^2, \dots\}.$$

Demonstração. Sebe-se que existe l tal que $p(\lambda^l) = 0$, porém, antes de demonstrar a afirmação anterior, precisa-se estender o homomorfismo $\sigma(\cdot)$ para o corpo $GF(4^s)$ o qual é uma extensão de $\mathbb{F}_4$.

1. Para $q(x) \in \mathbb{F}_4[x]$, define-se $\tilde{\sigma}(q(x)) = q(x)^2$, o qual é uma extensão de $\sigma(\cdot)$.
   $\tilde{\sigma}(\cdot)$ é um homomorfismo de anéis.
   - $\tilde{\sigma}(q_1(x) + q_2(x)) = (q_1(x) + q_2(x))^2 = q_1(x)^2 + q_2(x)^2 = \tilde{\sigma}(q_1(x)) + \tilde{\sigma}(q_2(x))$.
   - $\tilde{\sigma}(q_1(x)q_2(x)) = (q_1(x)q_2(x))^2 = q_1(x)^2 q_2(x)^2 = \tilde{\sigma}(q_1(x))\tilde{\sigma}(q_2(x))$.
2. Sabe-se que $GF(4^s) \cong \frac{\mathbb{F}_4[x]}{\langle P_{irr}(x)\rangle}$, onde $P_{irr}(x)$ é um polinômio irredutível de grau s sobre $\mathbb{F}_4$. Com estes elementos, usa-se o $\overline{(\cdot)}$ como o homomorfismo canônico que leva de $\mathbb{F}_4[x]$ em $GF(4^s)$. Define-se $\hat{\sigma}(\lambda^l) = \lambda^{2l}$ sobre $GF(4^s)$ para $1 \leq l \leq 4^s - 1$. Note que o mapa $\sigma(\cdot)$ respeita o subcorpo $\mathbb{F}_4$, pois: $(\mathbb{F}_4)^* = \{\alpha, \alpha^2, \alpha^3 = 1\}$ para $\alpha = \lambda^{\frac{4^s-1}{3}}$.
   - $\hat{\sigma}(\lambda^l) = \hat{\sigma}(\overline{q(x)}) = \left(\overline{q(x)}\right)^2 = \overline{(q(x)^2)} = \overline{\tilde{\sigma}(q(x))}$ para alguns $q(x)$ e l tal que $\overline{q(x)} = \lambda^l$.

$\hat{\sigma}(\cdot)$ é um homomorfismo de anéis.

- $\hat{\sigma}(\lambda^{l_1} + \lambda^{l_2}) = \overline{\tilde{\sigma}(q_1(x) + q_2(x))} = \overline{\tilde{\sigma}(q_1(x)) + \tilde{\sigma}(q_2(x))} = \hat{\sigma}(\lambda^{l_1}) + \hat{\sigma}(\lambda^{l_2})$.
- $\hat{\sigma}(\lambda^{l_1}\lambda^{l_2}) = \overline{\tilde{\sigma}(q_1(x)q_2(x))} = \overline{\tilde{\sigma}(q_1(x))\tilde{\sigma}(q_2(x))} = \hat{\sigma}(\lambda^{l_1})\hat{\sigma}(\lambda^{l_2})$.

As definições anteriores permitem completar a demonstração:

$$\begin{aligned}
\hat{\sigma}(p(\lambda^{2\cdot l})) =& \hat{\sigma}(a_0) + \hat{\sigma}(a_1)\lambda^{2\cdot l} + \cdots + \hat{\sigma}(a_{n-1})\lambda^{2\cdot l(n-1)} \\
\hat{\sigma}(p(\lambda^{2\cdot l})) =& \hat{\sigma}(a_0) + \hat{\sigma}(a_1\lambda^{l}) + \cdots + \hat{\sigma}(a_{n-1}\lambda^{l(n-1)}) \\
\hat{\sigma}(p(\lambda^{2\cdot l})) =& \hat{\sigma}(a_0 + a_1\lambda^{l} + \cdots + a_{n-1}\lambda^{l(n-1)}) \\
\hat{\sigma}(p(\lambda^{2\cdot l})) =& \hat{\sigma}(0) = 0.
\end{aligned}$$

□

Afirmação 4.11. *$c \in \mathcal{C} = \langle g(x) \rangle$ sobre $\mathbb{F}_4$, onde $\mathcal{C}$ é um código BCH com parâmetros (n, k, d) se, e somente se, $\sigma(c) \in \mathcal{C}' = \langle \sigma(g(x)) \rangle$, onde $\mathcal{C}'$ é um código BCH com parâmetros (n, k, d).*

Demonstração.

- Veja que provar a ida ($\Rightarrow$) implica em provar a volta ($\Leftarrow$) pois $\sigma^2(a) = a$.
- ($\Rightarrow$) Usando a Afirmação 4.9 e sabendo que $g(x)$ é fatorado por polinômios minimais, então, $p(x)|g(x)$ implica que $\sigma(p(x))|\sigma(g(x))$. Pela Afirmação 4.10, se $p(x)$ tem raízes $\{\lambda^l, \lambda^{4\cdot l}, \dots\}$, então $\sigma(p(x))$ tem raízes: $\{\lambda^{2\cdot l}, \lambda^{8\cdot l}, \dots\} = \{\beta^l, \beta^{4\cdot l}, \dots\}$ para $\beta = \lambda^2$.

  Dado que $\mathcal{C}$ é um código BCH com distância de projeto δ, segue que $g(x) = p_1(x) \dots p_t(x)$ e as raízes de $g(x)$ satisfazem o seguinte:

$$ROOTS(g(x)) = \bigcup_i ROOTS(p_i(x)),$$

  onde $ROOTS(g(x)) \supseteq \{\lambda^e, \lambda^{e+j}, \dots \lambda^{e+j(\delta-2)}\}$. Aplicando as Afirmações 4.9 e 4.10, obtêm-se:

  - $\sigma(g(x)) = \sigma(p_1(x)) \dots \sigma(p_t(x))$
  - $ROOTS(\sigma(g(x))) = \bigcup_i ROOTS(\sigma(p_i(x)))$,

  onde:

$$\begin{aligned}
ROOTS(\sigma(g(x))) \supseteq& \{\lambda^{2e}, \lambda^{2(e+j)}, \dots \lambda^{(2(e+j(\delta-2))}\} \\
ROOTS(\sigma(g(x))) \supseteq& \{\beta^{e}, \beta^{e+j}, \dots \beta^{e+j(\delta-2)}\}. \qquad (4.1)
\end{aligned}$$

  Pela Equação 4.1, conclui-se que o código $\mathcal{C}' = \langle \sigma(g(x)) \rangle$ é BCH e tem distância de projeto δ. Por último, note que o homomorfismo $\sigma(\cdot)$ não troca o grau do polinômio,

portanto, $\mathcal{C}'$ tem dimensão k e comprimento n, o qual é justificado pelo seguinte fato:

$$g(x)h(x) = x^n - 1 \quad \Rightarrow \quad \sigma(g(x))\sigma(h(x)) = \sigma(x^n - 1) = x^n - 1$$

□

A Afirmação 4.11 mostra que os resultados obtidos pelo Rótulo A podem ser usados para deduzir os resultados que se obteriam através do Rótulo B. Sem considerar o polinômio $x-1$, se uma sequência de nucleotídeos for identificada como palavra-código de um código BCH ($\mathcal{C} = \langle g(x) \rangle$) através de algum dos rotulamentos de A, então, essa mesma sequência pode ser identificada como uma palavra-código do código BCH: $\mathcal{C}' = \langle \sigma(g(x)) \rangle$ através de algum dos rotulamentos de B; além disso, os parâmetros de $\mathcal{C}$ e de $\mathcal{C}'$ são os mesmos, i.e., igual comprimento, dimensão e distância de projeto. Nos trabalhos (FARIA, 2011) e (ROCHA, 2010), como somente códigos nsBCH com distância de projeto igual a 3 foram capazes de identificar sequências de nucleotídeos então as raízes dos polinômios geradores sempre eram da forma $\{\lambda, \lambda^4, \dots\} \bigcup \{\lambda^2, \lambda^8, \dots\}$ ($g(x) = p(x)\sigma(p(x))$) e, portanto, os $g(x)$ sempre têm coeficientes sobre $\mathbb{F}_2$ e são invariantes à transformação do conjugado ($\sigma(g(x)) = g(x)$). A explicação anterior demonstra a razão pela qual, quando considerados códigos nsBCH com distância de projeto igual a três na identificação de sequências de nucleotídeos, basta estudar um único rotulamento que os mesmos resultados serão obtidos para os outros 23 rotulamentos.

As Afirmações 4.7 e 4.11 mostram que no caso de identificar sequências de nucleotídeos através de códigos BCH, de maneira geral, sobre $\mathbb{F}_4$, obtêm-se 8 subclasses de rotulamentos: A_1, A_2, A_3, A_4, B_1, B_2, B_3 e B_4. Porém existe a relação do conjugado entre os rotulamentos A e B, e portanto basta analisar o Rótulo A e deduzir o código para o Rótulo B. Esta caraterística será aproveitada pelo algoritmo de identificação de sequências de nucleotídeos que será apresentado na Seção 4.4.

Para cada uma das subclasses A e B existe uma subdivisão em 4 subclasses: 1, 2, 3 e 4; a qual indica para qual dos 4 possíveis rotulamentos o polinômio obtido, após o mapeamento para $\mathbb{F}_4$, é divisível pelo polinômio $x - 1$; i.e., para qual dos 4 possíveis rotulamentos a paridade é zero. De maneira análoga ao que foi usado com o alfabeto $\mathbb{Z}_4$ o algoritmo que será mostrado na Seção 4.4 aproveitará esta característica para reduzir a redundância computacional.

Analogamente à Seção 4.3.1, nos trabalhos (FARIA, 2011) e (ROCHA, 2010) descobriu-se que os resultados obtidos ao estudar uma sequência de nucleotídeos na direção 5'-3' são os mesmos aos obtidos ao estudar a sequência antiparalela. Lembre que o procedimento para obter a sequência antiparalela ($\text{rev}(cnuc(x))$) a partir da sequência 5'-3' ($nuc(x)$) basta simplesmente trocar as bases: $A \mapsto U$, $U \mapsto A$, $G \mapsto C$ e $C \mapsto G$ na sequência e de ler a sequência na direção oposta.

Afirmação 4.12. *Considere o alfabeto como sendo* $\mathbb{F}_4$*. Se uma dada sequência de nucleotídeos lida na direção 5'-3' e denotada por* $nuc(x)$ *pertence ao código* $\mathcal{C} = \langle g(x) \rangle$ *com parâmetros* (n, k, d) *através do Rótulo* A_i *(ou* B_i*), então a sequência antiparalela de nucleotídeos, denotada por* $rev(cnuc(x))$*, pertence ao código* $\mathcal{C}_2 = \langle rev(g(x)) \rangle$ *com parâmetros* (n, k, d) *através do Rótulo* $A_{(3-i)}$ *(ou* $B_{(3-i)}$*).*

Demonstração. Considere a sequência de nucleotídeos $nuc(x)$ e utilize o rotulamento e_i, que pertence ao Rótulo A_0 ou B_0, para mapeá-la numa sequência sobre $\mathbb{F}_4$, denotada como $nuc_{e_i}(x) \in \mathbb{F}_4^n$. Observe, nas Tabelas 9 e 10, que $nuc_{e_i}(x) = cnuc_{(3-e_i)}$. Assim, estudar a sequência 5'-3' pelo rotulamento e_i é equivalente a ter estudado a sequência $cnuc_{(3-e_i)} \in \mathbb{F}_4^n$, e portanto os mesmos códigos BCH são obtidos, i.e., o mesmo polinômio gerador $g(x)$. Sejam $\{\alpha^i, \ldots, \alpha^{d-2}\}$ as raízes de $g(x)$, as quais são também raízes de $cnuc_{(3-e_i)}(x)$. Assim, as raízes de $\text{rev}(g(x))$ são $\{\alpha^{-i}, \ldots, \alpha^{-(d-2)}\}$, as quais são também raízes de $\text{rev}(cnuc_{(3-e_i)}(x))$. Portanto, $\text{rev}(cnuc_{(3-e_i)}(x)) \in \mathcal{C}_2$ e $\mathcal{C}_2$ tem parâmetros (n, k, d). □

4.4 Algoritmo para Identificação de Sequências mRNA

Utilizando as propriedades dos códigos BCH, a subdivisão dos rótulos e as propriedades provadas na seção anterior; pode-se propor um algoritmo eficiente que identifique sequências de nucleotídeos como palavras-código de códigos BCH, as quais são moduladas através do código genético e, portanto, representam a informação redundante usada na síntese de proteínas. O diagrama de blocos dos algoritmos ($BCH\_OneNuc\_\mathbb{Z}_4$ e $BCH\_OneNuc\_\mathbb{F}_4$) usados para identificar sequências sobre $\mathbb{Z}_4$ e sobre $\mathbb{F}_4$ são ilustrados nas Figuras 21 e 22, respectivamente.

Os dois algoritmos são similares, a menos de diferenças no bloco de aplicação do rotulamento. No caso do algoritmo $BCH\_OneNuc\_\mathbb{Z}_4$, o algoritmo deve ser executado três vezes para cada um dos seguintes três rotulamentos: e_1, e_2 e e_3. Estas três execuções são suficientes para obter todos os resultados que seriam obtidos para cada um dos outros 21 rotulamentos. No caso do algoritmo $BCH\_OneNuc\_\mathbb{F}_4$, o algoritmo deve ser executado uma única vez para o rotulamento e e esta execução é suficiente para obter todos os resultados que seriam obtidos quando aplicados em cada um dos outros 23 rotulamentos. Estas características aproveitam as propriedades dos rotulamentos apresentadas nas afirmações das Seções 4.3.1 e 4.3.2.

A seguir explicam-se cada um dos blocos que compõem os algoritmos:

- **Aplicação do rotulamento:** Este bloco recebe uma sequência de nucleotídeos de tamanho n e um rotulamento (veja as Tabelas 8 e 10). No bloco, a sequência

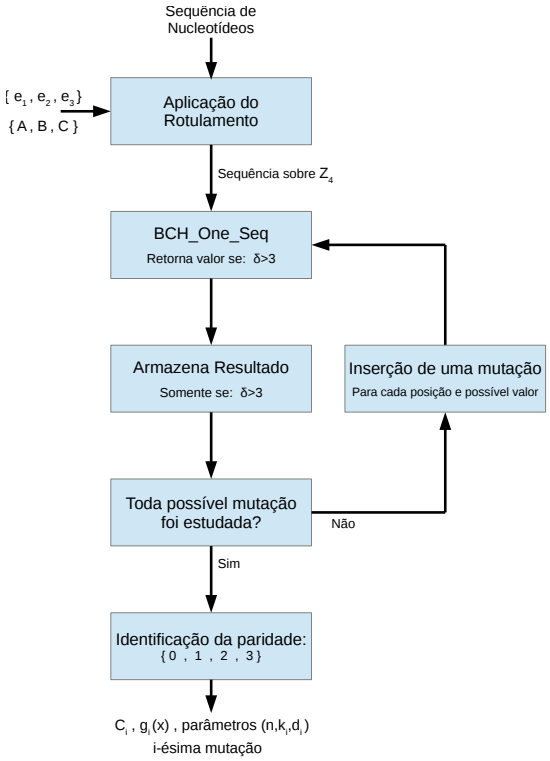


Figura 21 – $BCH\_OneNuc\_\mathbb{Z}_4(nuc)$.

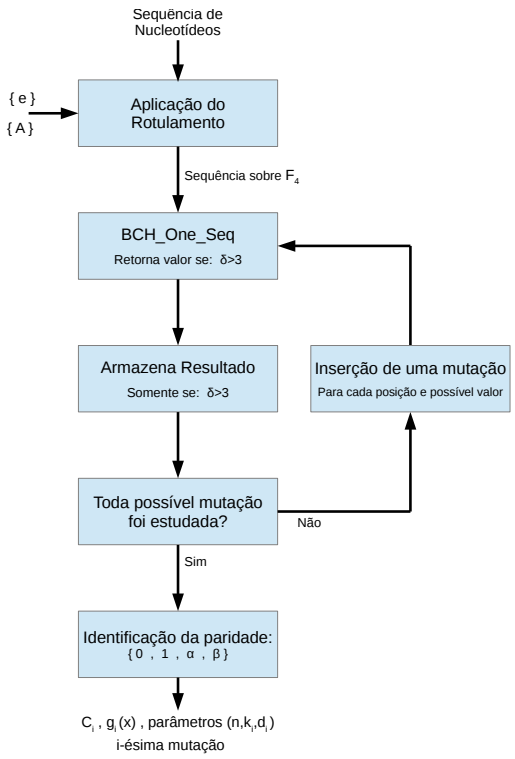


Figura 22 – $BCH\_OneNuc\_\mathbb{F}_4(nuc)$.

de nucleotídeos é mapeada numa sequência no alfabeto $\mathbb{Z}_4$ ou $\mathbb{F}_4$, dependendo da situação. A saída é uma sequência sobre o alfabeto $\mathbb{Z}_4$ ou $\mathbb{F}_4$.

- *BCH_One_Seq*: Revise o funcionamento deste algoritmo no Capítulo 3. Este bloco recebe uma sequência sobre $\mathbb{Z}_4$ ou $\mathbb{F}_4$. Este bloco encontra o polinômio gerador do maior código BCH, $\mathcal{C}$, de comprimento n e com a maior distância de projeto, tal que a sequência de entrada pertence a $\mathcal{C}$. As saídas são os parâmetros do código BCH, o polinômio gerador, $g(x)$, e os polinômios minimais que fatoram $g(x)$.

- **Armazena resultado:** Neste bloco, o resultado é armazenado, somente se o código obtido tiver uma distância de projeto maior que 3.

- **Toda possível mutação foi estudada?** Este bloco controla o fluxo do algoritmo e verifica se cada uma das três possíveis mutações em cada uma das possíveis posições (SNPs) foram consideradas. Ele retorna "Não" quando alguma mutação ainda não foi considerada e retorna "Sim" quando todas as possíveis mutações de um único nucleotídeo já foram consideradas.

- **Inserção de uma mutação:** Este bloco insere uma única mutação na sequência de nucleotídeos com respeito à sequência original (SNP). Dado que somente há três possíveis mutações para cada uma das posições, segue que o algoritmo *BCH_One_Seq* é executado: $(n \cdot 3) + 1$ vezes. Esta é a justificativa de procurar um algoritmo rápido e de poupar a maior quantidade de rotulamentos.

- **Identificação da paridade:** Este bloco computa a paridade da sequência de nucleotídeos sem mutação depois de ter sido mapeada sobre $\mathbb{Z}_4$ ou $\mathbb{F}_4$. Determina-se para qual das 4 subclasses o polinômio $x-1$ é um divisor ao verificar para qual dos quatro rotulamentos a paridade é zero. Por exemplo, para $\mathbb{Z}_4$, se o Rótulo B está sendo considerado, então este bloco encontra para qual dos Rótulos: B_1, B_2, B_3 ou B_4, obtém-se paridade zero. O Rótulo obtido seria o único que leva em uma sequência que é divisível pelo polinômio $x-1$.

Note que nos algoritmos anteriores escolheu-se uma distância de projeto maior que 3 como critério de decisão se o código devia ser armazenado e considerado. Este número foi determinado experimentalmente, pois encontrou-se que somente códigos BCH com distâncias de projeto pequenas eram capazes de identificar sequências de nucleotídeos. Por outro lado, códigos com distâncias de projeto menores que três não foram considerados porque são códigos muito pobres e não são uteis para mostrar que o sistema biológico tem capacidade de detecção ou correção de erros. Códigos com distância de projeto igual a três também não foram considerados porque em muitos dos casos os códigos dependiam do polinômio $x-1$ o qual parece não ter muita importância porque, como provado nas Afirmações 4.3 e 4.7, qualquer sequência sempre é divisível pelo polinômio $x-1$ para algum Rótulo 1, 2, 3 e 4.

O fato que o polinômio $x-1$ parece não ter muita importância não quer dizer que este nunca é considerado, pois, como será visto na Seção 4.5, existem sequências para as quais o polinômio $x-1$ é necessário para obter uma distância de projeto maior que 3. Além disso, algumas dessas sequências são ditas reversíveis porque ambas sequências: 1) quando lida da esquerda para direita e 2) quando lida da direita para esquerda, pertencem ao mesmo código.

Utilizando esta noção de sequências reversíveis, prova-se que se uma dada sequência de nucleotídeos ($nuc(x)$) é identificada como palavra-código de um código BCH ($\mathcal{C}$) com parâmetros (n,k,δ), então a sequência invertida ($nuc_{Rev}(x)$) também é identificada como palavra-código de um código BCH ($\mathcal{C}'$) com parâmetros (n,k,δ). Define-se, um *código reversível* quando a relação $\mathcal{C}=\mathcal{C}'$ é satisfeita, e assim, $nuc(x)\in\mathcal{C}$ implica que $nuc_{Rev}(x)\in\mathcal{C}$.

Afirmação 4.13. *Se a sequência de nucleotídeos $nuc(x)$ é identificada pelo código $\mathcal{C}$ com parâmetros (n,k,δ) através de algum rotulamento, denotado por rot, então a sequência $nuc_{Rev}(x)$ é identificada por um código $\mathcal{C}'$ com parâmetros (n,k,δ) através do rotulamento rot; onde δ é a distância de projeto.*

Demonstração. Seja $\langle g(x)\rangle=\mathcal{C}$, onde $ROOTS(g(x))=\{\lambda_1,\ldots,\lambda_L\}=ROOTS(nuc(x))$, o grau de $g(x)$ é L e $rot(nuc(x))\in\mathcal{C}$. Dado que $nuc_{Rev}(x)=x^{n-1}nuc(1/x)$, segue que as

raízes de $nuc_{Rev}(x)$ são:

$$ROOTS(nuc_{Rev}(x)) = \{\lambda_1^{-1}, \ldots, \lambda_L^{-1}\}.$$

Como $\mathcal{C}$ é um código BCH com distância de projeto δ, segue que existe um gerador do grupo separante (λ), tal que: $\{\lambda^l, \lambda^{l+1}, \ldots, \lambda^{l+\delta-2}\} \subseteq ROOTS(nuc(x))$. Considere o código $\mathcal{C}'$ gerado pelo polinômio $g'(x) = \prod_i^L (x - \lambda_i^{-1})$. Logo:

- Grau de $g(x)$ é igual ao grau de $g'(x)$ (mesma dimensão para $\mathcal{C}$ e $\mathcal{C}'$).
- $\{\lambda^{-l}, \lambda^{-l-1}, \ldots, \lambda^{-l-\delta+2}\} \subseteq ROOTS(nuc - Rev(x))$ (igual distância de projeto).

□

Observe na demonstração anterior: para que $nuc(x) \in \mathcal{C}$ e $nuc_{Rev}(x) \in \mathcal{C}' = \mathcal{C}$ é necessário que $ROOTS(nuc(x)) = ROOTS(nuc_{Rev}(x))$.

4.4.1 Exemplos: Algoritmo sobre sequências de nucleotídeos

Nesta seção, apresenta-se alguns exemplos de sequências de nucleotídeos identificadas como palavras-código de códigos BCH através dos algoritmos apresentados na Seção 4.4 (Figuras 21 e 22).

Exemplo 4.1. *Considere a sequência mRNA identificada na Base de Dados do NCBI (*National Center for Biotechnology Information*) como: "gi|334188617|Arabidopsis thaliana conserved peptide upstream open reading frame 38, complete cds" que possui a seguinte representação sobre nucleotídeos:*
Ont: *AUGUGUAUUGCCGUAUACCGUAAAGUUUUGAGCUUGAAUCUGUAUUGCCGUGUGAUACUGUAG*
Aplica-se o algoritmo BCH_OneNuc_$\mathbb{Z}_4$ (Figura 21) na sequência mRNA

- **Aplicação do rotulamento:** Para o alfabeto $\mathbb{Z}_4$, os três rotulamentos e_1, e_2 e e_3 devem ser utilizados:

  e_1 : 023232022311320201132000322223031223002123202231132323020123203
  e_2 : 032323033211230301123000233332021332003132303321123232030132302
  e_3 : 031313033122130302213000133331012331003231303312213131030231301

- *BCH_One_Seq***:** Este bloco é aplicado nas 3 sequências mostradas acima e, para nenhuma dessas sequências, se encontra um código BCH tal que essas sequências sejam palavras-código. Portanto, consideram-se todas as possíveis mutações em uma única posição (SNPs) para identificar a sequência com a mutação como palavra-código. Este procedimento faz sentido porque os códigos que serão encontrados têm

distância maior que três, e assim, a sequência original estaria na nuvem de sequências corrigíveis. A mutação é inserida pelo bloco: **Inserção de uma mutação**.

Ao analisar as mutações, chega-se em uma única sequência obtida através do rótulo e_1 com uma única diferença com respeito a original é identificada como palavra-código de um código BCH com distância 3:

$\widehat{e}_1$: 023232022311320201132000322223031223001123202231132323020123203

As características do código são mostradas a seguir:

- Polinômio gerador: $g(x) = 1 + 2x^3 + 3x^5 + x^6$.
- Polinômio Minimais que compõem $g(x)$: $pm_1(x) = 1 + 2x^3 + 3x^5 + x^6$.
- As raízes de $g(x)$ são: $\{\lambda, \lambda^2, \lambda^4, \lambda^8, \lambda^{16}, \lambda^{32}\}$; mostrando que este é um código nsBCH.

Estas informações são armazenadas no bloco **Armazena resultado**.

- **Identificação da paridade:** A paridade da sequência com a mutação é: 0. Portanto, usando as afirmações da Seção 4.3.1, chega-se que a sequência é identificada através de qualquer rotulamento do Rótulo A como palavra-código de um código nsBCH de distância 3 com polinômio gerador $g(x) = 1 + 2x^3 + 3x^5 + x^6$. Porém, dado que a paridade é 0, segue que somente através do Rótulo A_1, obtém-se um código BCH de distância 4 com as seguintes características:
  - Polinômio gerador: $g(x) = 3 + x + 2x^3 + 2x^4 + x^5 + 2x^6 + x^7$.
  - Polinômio Minimais que compõem $g(x)$: $pm_1(x) = 1+2x^3+3x^5+x^6$, $pm_2(x) = 3 + x$.
  - As raízes de $g(x)$ são: $\{\lambda^0 = 1, \lambda, \lambda^2, \lambda^4, \lambda^8, \lambda^{16}, \lambda^{32}\}$.

Para finalizar, a seguinte sequência de nucleotídeos com um SNP é identificada como palavra-código de um código BCH (Ver Tabela 11):

Mut: AUGUGUAUUGCCGUAUACCGUAAAGUUUUGAGCUUGAA**C**CUGUAUUGCCGUGUGAUACUGUAG

Observe que a mutação sugerida pelo código BCH é uma mutação silenciosa pelo fato de que a tradução da sequência original (Ont) para aminoácidos (Oaa) é igual à tradução da sequência gerada pelo código (Gnt) para aminoácidos (Gaa).

Exemplo 4.2. *Considere a sequência mRNA identificada na Base de Dados NCBI como: "gi|899225|B.napus mRNA for mitochondrial malate dehydrogenase" que possui a seguinte representação sobre nucleotídeos:*

Ont: *UUCAGAUCCGCGCUUGUCCGAUCCUCCGCCUCGGCGAAGCAGUCGCUUCUCCGCCGCAGCUUC*

Aplica-se o algoritmo BCH_OneNuc_$\mathbb{F}_4$ (Figura 22) na sequência mRNA.

Tabela 11 – Análise sobre $\mathbb{Z}_4$ da sequência mRNA: "gi|334188617|Arabidopsis thaliana"

Código $\mathbb{Z}_4$-linearidade nsBCH (63,56,4) sobre $\mathbb{Z}_4$ e Rótulo A_1

Polinômio Gerador: $g(x) = 3 + x + 2x^3 + 2x^4 + x^5 + 2x^6 + x^7$

Rotulamento: $(A, C, G, U) \to (0, 1, 3, 2)$

Oaa:	M	C	I	A	V	Y	R	K	V	L	S	L	**N**	L	Y
Ont:	AUG	UGU	AUU	GCC	GUA	UAC	CGU	AAA	GUU	UUG	AGC	UUG	AA**U**	CUG	UAU
Olb:	023	232	022	311	320	201	132	000	322	223	031	223	00**2**	123	202
Glb:	023	232	022	311	320	201	132	000	322	223	031	223	00**1**	123	202
Gnt:	AUG	UGU	AUU	GCC	GUA	UAC	CGU	AAA	GUU	UUG	AGC	UUG	AA**C**	CUG	UAU
Gaa:	M	C	I	A	V	Y	R	K	V	L	S	L	**N**	L	Y

Oaa:	C	R	V	I	L	Stop
Ont:	UGC	CGU	GUG	AUA	CUG	UAG
Olb:	231	132	323	020	123	203
Glb:	231	132	323	020	123	203
Gnt:	UGC	CGU	GUG	AUA	CUG	UAG
Gaa:	C	R	V	I	L	Stop

- **Aplicação do rotulamento:** Para o alfabeto $\mathbb{F}_4$, somente o rotulamento e deve ser utilizado:

$$e: \quad \beta\beta 10\alpha 0\beta 11\alpha 1\alpha 1\beta\beta\alpha\beta 11\alpha 0\beta 11\beta 11\alpha 11\beta 1\alpha\alpha 1\alpha 00\alpha 10\alpha\beta 1\alpha 1\beta\beta 1\beta 11\alpha 11\alpha 10\alpha 1\beta\beta 1$$

- *BCH_One_Seq***:** Este bloco é aplicado na sequência mostrada acima e, para essa sequência, não se encontra um código BCH tal que essa sequência seja palavra-código. Portanto, consideram-se todas as possíveis mutações em uma única posição (SNPs) para identificar a sequência com a mutação como palavra-código. Como no exemplo anterior, este procedimento faz sentido porque os códigos que serão encontrados têm distância maior que três, e assim, a sequência original estaria na nuvem de sequências corrigíveis. A mutação é inserida pelo bloco: **Inserção de uma mutação**.

  Ao analisar as mutações, chega-se que a seguinte sequência com uma única diferença com respeito a original é identificada como palavra-código de um código BCH com distância 4:

$$\widehat{e}: \quad \boldsymbol{\alpha}\beta 10\alpha 0\beta 11\alpha 1\alpha 1\beta\beta\alpha\beta 11\alpha 0\beta 11\beta 11\alpha 11\beta 1\alpha\alpha 1\alpha 00\alpha 10\alpha\beta 1\alpha 1\beta\beta 1\beta 11\alpha 11\alpha 10\alpha 1\beta\beta 1$$

  As características do código são mostradas a seguir:

  - Polinômio gerador: $g(x) = \beta + x + \alpha x^2 + x^4 + \alpha x^5 + x^7$.
  - Polinômio Minimais que compõem $g(x)$: $pm_1(x) = \beta + x + \alpha x^2 + x^3$, $pm_2(x) = \alpha + \beta x + x^2 + x^3$ e $pm_3(x) = \beta + x$.
  - As raízes de $g(x)$ são: $\{\lambda^5, \lambda^{13}, \lambda^{17}, \lambda^{19}, \lambda^{20}, \lambda^{21}, \lambda^{52}\}$; mostrando que este **não** é um código nsBCH e será denotado como código **BCH**.

  Estas informações são armazenadas no bloco **Armazena resultado**.

- **Identificação da paridade:** A paridade da sequência com a mutação é: 0. Porém, como visto na Seção 4.3.2, a paridade somente afeta na adição da raiz $\lambda^0 = 1$ e na adição do fator $x - 1$ no polinômio gerador, o qual não aumenta a distância de projeto do código BCH. Portanto, o polinômio gerador para qualquer Rótulo não considera a paridade e, portanto, esse mesmo polinômio gerador $g(x) = \beta + x + \alpha x^2 + x^4 + \alpha x^5 + x^7$ é obtido quando considerado todos os Rótulos A_1, A_2, A_3 e A_4, i.e., qualquer rotulamento no Rótulo A.

  Pela Afirmação 4.11, sabe-se que a diferença entre o Rótulo A e o Rótulo B é o conjugado, portanto se algum rotulamento do Rótulo B é considerado, então o polinômio gerador obtido será: $\overline{g(x)} = \alpha + x + \beta x^2 + x^4 + \beta x^5 + x^7$ e, novamente, a adição do polinômio $x-1$ não aumenta a distância de projeto. Os mesmos resultados são obtidos quando considerados todos os Rótulos: B_1, B_2, B_3 e B_4.

Para finalizar, a seguinte sequência de nucleotídeos com um SNP é identificada como palavra-código de um código BCH (Ver Tabela):

Mut: GUCAGAUCCGCGCUUGUCCGAUCCUCCGCCUCGGCGAAGCAGUCGCUUCUCCGCCGCAGCUUC

Observe que a mutação sugerida pelo código **BCH** não é uma mutação silenciosa visto que a tradução da sequência original (Ont) para aminoácidos (Oaa) difere num aminoácido à tradução da sequência gerada pelo código (Gaa).

Tabela 12 – Análise sobre $\mathbb{F}_4$ da sequência mRNA: "gi|899225|B.napus for mitochondrial malate dehydrogenase"

Código BCH (63,56,4) sobre $\mathbb{F}_4$ e Rótulo A

Polinômio Gerador: $g(x) = \beta + x + \alpha x^2 + x^4 + \alpha x^5 + x^7$

Rotulamento: $(A, C, G, U) \to (0, 1, \alpha, \beta)$

Oaa:	F	R	S	A	L	V	R	S	S	A	S	A	K	Q	S
Ont:	UUC	AGA	UCC	GCG	CUU	GUC	CGA	UCC	UCC	GCC	UCG	GCG	AAG	CAG	UCG
Olb:	$\beta\beta1$	$0\alpha0$	$\beta11$	$\alpha1\alpha$	$1\beta\beta$	$\alpha\beta1$	$1\alpha0$	$\beta11$	$\beta11$	$\alpha11$	$\beta1\alpha$	$\alpha1\alpha$	00α	10α	$\beta1\alpha$
Glb:	$\alpha\beta1$	$0\alpha0$	$\beta11$	$\alpha1\alpha$	$1\beta\beta$	$\alpha\beta1$	$1\alpha0$	$\beta11$	$\beta11$	$\alpha11$	$\beta1\alpha$	$\alpha1\alpha$	00α	10α	$\beta1\alpha$
Gnt:	GUC	AGA	UCC	GCG	CUU	GUC	CGA	UCC	UCC	GCC	UCG	GCG	AAG	CAG	UCG
Gaa:	V	R	S	A	L	V	R	S	S	A	S	A	K	Q	S

Oaa:	L	L	R	R	S	F
Ont:	CUU	CUC	CGC	CGC	AGC	UUC
Olb:	$1\beta\beta$	$1\beta1$	$1\alpha1$	$1\alpha1$	$0\alpha1$	$\beta\beta1$
Glb:	$1\beta\beta$	$1\beta1$	$1\alpha1$	$1\alpha1$	$0\alpha1$	$\beta\beta1$
Gnt:	CUU	CUC	CGC	CGC	AGC	UUC
Gaa:	L	L	R	R	S	F

Exemplo 4.3. *Considere a sequência identificada na Base de Dados NCBI como: "gi|217937|Mitocôndria - F1-AT Pase delta subunit" que possui a seguinte representação sobre nucleotídeos:*

Ont: *AUGUUCAGGCACUCUUCUCGACUCCUAGCUCGCGCCACCACAAUGGGGUGGCGUCGCCCCUUC*

Aplica-se o algoritmo BCH_OneNuc_$\mathbb{F}_4$ (Figura 22) na sequência de nucleotídeos.

- **Aplicação do rotulamento:** Para o alfabeto $\mathbb{F}_4$, somente o rotulamento e deve ser utilizado:

$$e: \quad 0\beta\alpha\beta\beta 10\alpha\alpha 101\beta 1\beta\beta 1\beta 1\alpha 01\beta 11\beta 0\alpha 1\beta 1\alpha 1\alpha 110110100\beta\alpha\alpha\alpha\beta\alpha\alpha 1\alpha\beta 1\alpha 1111\beta\beta 1$$

- *BCH_One_Seq***:** Este bloco é aplicado na sequência mostrada acima e, para essa sequência, não se encontra um código BCH tal que essa sequência seja palavra-código. Portanto, consideram-se todas as possíveis mutações em uma única posição (SNPs) para identificar a sequência com a mutação como palavra-código. A mutação é inserida pelo bloco: **Inserção de uma mutação**.

  Ao analisar as mutações, chega-se que a seguinte sequência com uma única diferença com respeito a original é identificada como palavra-código de um código BCH com distância 3:

$$\widehat{e}: \quad 0\beta\alpha\beta\beta 10\alpha\alpha 101\beta 1\beta\beta 1\beta 1\alpha 01\beta 11\beta 0\alpha 1\beta 1\alpha 1\alpha 110110100\beta\alpha\alpha\alpha\beta\alpha\mathbf{0}1\alpha\beta 1\alpha 1111\beta\beta 1$$

  As características do código são mostradas a seguir:

  - Polinômio gerador: $g(x) = 1 + x^2 + x^3 + x^5 + x^6$.
  - Polinômio Minimais que compõem $g(x)$: $pm_1(x) = \alpha + \alpha x + \alpha x^2 + x^3$ e $pm_2(x) = \beta + \beta x + \beta x^2 + x^3$.
  - As raízes de $g(x)$ são: $\{\lambda^1, \lambda^2, \lambda^4, \lambda^8, \lambda^{16}, \lambda^{32}\}$; mostrando que este é um código nsBCH.

  Estas informações são armazenadas no bloco **Armazena resultado**.

- **Identificação da paridade:** A paridade da sequência com a mutação é: β. A paridade adiciona a raiz $\lambda^0 = 1$ e o fator $x-1$ no polinômio gerador, o qual aumenta a distância de projeto do código BCH. Portanto, a sequência é identificada através de qualquer rotulamento do Rótulo A como palavra-código de um código nsBCH de distância 3 com polinômio gerador $g(x) = 1 + x^2 + x^3 + x^5 + x^6$. Porém, dado que a paridade é β, segue que somente através do Rótulo A_4, obtém-se um código BCH de distância 4 com as seguintes características:

  - Polinômio gerador: $g(x) = 1 + x + x^2 + x^4 + x^5 + x^7$.
  - Polinômio Minimais que compõem $g(x)$: $pm_0(x) = 1 + x$, $pm_1(x) = \alpha + \alpha x + \alpha x^2 + x^3$ e $pm_2(x) = \beta + \beta x + \beta x^2 + x^3$.
  - As raízes de $g(x)$ são: $\{\lambda^0 = 1, \lambda^1, \lambda^2, \lambda^4, \lambda^8, \lambda^{16}, \lambda^{32}\}$; mostrando que este é um código **nsBCH**.

  Pela Afirmação 4.11, sabe-se que a diferença entre o Rótulo A e o Rótulo B é o conjugado, e portanto se algum rotulamento do Rótulo B é considerado, então o

polinômio gerador do código nsBCH obtido será: $\overline{g(x)} = 1+x^2+x^3+x^5+x^6 = g(x)$. Esta é a justificativa de ter um único Rótulo (todos os 24 rotulamentos levam aos mesmos resultados) quando são considerados códigos nsBCH. De maneira similar ao Rótulo A, a adição do polinômio $x-1$ aumenta a distância de projeto, e assim, somente através do Rótulo B_4, obtém-se um código **nsBCH** de distância 4 com polinômio gerador: $\overline{g(x)} = g(x) = 1+x+x^2+x^4+x^5+x^7$.

Para finalizar, a seguinte sequência de nucleotídeos com um SNP é identificada como palavra-código de um código nsBCH (Ver Tabela):

Mut: AUGUUCAGGCACUCUUCUCGACUCCUAGCUCGCGCCACCACAAUGGGGUG**A**CGUCGCCCCUUC

Observe que a mutação sugerida pelo código nsBCH não é uma mutação silenciosa.

Tabela 13 – Análise sobre $\mathbb{F}_4$ da sequência: "gi|217937|Mitocôndria - F1-AT Pase delta subunit"

Código nsBCH (63,57,3) sobre $\mathbb{F}_4$ e qualquer Rótulo

Polinômio Gerador: $g(x) = 1+x^2+x^3+x^5+x^6$

Rotulamento: $(A, C, G, U) \to (1, 0, \beta, \alpha)$

Oaa:	M	F	R	H	S	S	R	L	L	A	R	A	U	U	M
Ont:	AUG	UUC	AGG	CAC	UCU	UCU	CGA	CUC	CUA	GCU	CGC	GCC	ACC	ACA	AUG
Olb:	$1\alpha\beta$	$\alpha\alpha0$	$1\beta\beta$	010	$\alpha0\alpha$	$\alpha0\alpha$	$0\beta1$	$0\alpha0$	$0\alpha1$	$\beta0\alpha$	$0\beta0$	$\beta00$	100	101	$1\alpha\beta$
Glb:	$1\alpha\beta$	$\alpha\alpha0$	$1\beta\beta$	010	$\alpha0\alpha$	$\alpha0\alpha$	$0\beta1$	$0\alpha0$	$0\alpha1$	$\beta0\alpha$	$0\beta0$	$\beta00$	100	101	$1\alpha\beta$
Gnt:	AUG	UUC	AGG	CAC	UCU	UCU	CGA	CUC	CUA	GCU	CGC	GCC	ACC	ACA	AUG
Gaa:	M	F	R	H	S	S	R	L	L	A	R	A	U	U	M

Oaa:	G	**W**	R	R	P	F
Ont:	GGG	UG**G**	CGU	CGC	CCC	UUC
Olb:	$\beta\beta\beta$	$\alpha\beta\boldsymbol{\beta}$	$0\beta\alpha$	$0\beta0$	000	$\alpha\alpha0$
Glb:	$\beta\beta\beta$	$\alpha\beta\mathbf{1}$	$0\beta\alpha$	$0\beta0$	000	$\alpha\alpha0$
Gnt:	GGG	UG**A**	CGU	CGC	CCC	UUC
Gaa:	G	**Stop**	R	R	P	F

Exemplo 4.4. *Considere a sequência identificada na Base de Dados NCBI como: "gi/632733/- N. tabacum-Pathogen and wound-inducible antifungal protein CBP20* " que possui a seguinte representação sobre nucleotídeos:*

Ont: *GGAAAGCUAAGUACACUUUUAUUUGCUCUGGUCCUCUAUGUCAUAGCCGCAGGAGCUAAUGCA*

Aplica-se o algoritmo $BCH\_OneNuc\_\mathbb{F}_4$ (Figura 22) na sequência de nucleotídeos.

- **Aplicação do rotulamento:** Para o alfabeto $\mathbb{F}_4$, somente o rotulamento e deve ser utilizado:

  e : $\alpha\alpha000\alpha1\beta00\alpha\beta0101\beta\beta\beta\beta0\beta\beta\beta\alpha1\beta1\beta\alpha\alpha\beta11\beta1\beta0\beta\alpha\beta10\beta0\alpha11\alpha10\alpha\alpha0\alpha1\beta00\beta\alpha10$

- *BCH_One_Seq***:** Este bloco é aplicado na sequência mostrada acima e, para essa sequência, não se encontra um código BCH tal que essa sequência seja palavra-código. Portanto, consideram-se todas as possíveis mutações em uma única posição

(SNPs) para identificar a sequência com a mutação como palavra-código. A mutação é inserida pelo bloco: **Inserção de uma mutação**.

Ao analisar as mutações, chega-se que a seguinte sequência com uma única diferença com respeito a original é um múltiplo dos polinômios minimais $pm_1(x) = \beta + \beta x + \beta x^2 + x^3$ e $pm_2(x) = \alpha + x + x^2 + x^3$:

$$\widehat{e}: \quad \alpha\mathbf{1}000\alpha1\beta00\alpha\beta0101\beta\beta\beta\beta0\beta\beta\beta\alpha1\beta1\beta\alpha\alpha\beta11\beta1\beta0\beta\alpha\beta10\beta0\alpha11\alpha10\alpha\alpha0\alpha1\beta00\beta\alpha10$$

Se o polinômio gerador fosse a multiplicação desses dois polinômios minimais ($g(x) = pm_1(x)pm_2(x)$), então as raízes de $g(x)$ seriam:

$$\{\lambda^1, \lambda^4, \lambda^{16}, \lambda^{47}, \lambda^{59}, \lambda^{62}\}.$$

Observe que o código gerado por $g(x)$ tem distância de projeto 2 e que se a raiz $\lambda^0 = 1$ é adicionada (i.e., o fator $x-1$ é adicionado ao polinômio gerador), o novo polinômio gerador corresponderia a um código BCH de distância 4. Esta característica ocorre somente para um dos Rótulos A_1, A_2, A_3 ou A_4, como visto na Afirmação 4.7. Portanto, quando esta situação ocorre, o bloco **Armazena resultado** armazena as informações anteriores.

- **Identificação da paridade:** A paridade da sequência com a mutação é: 0. Portanto, somente para o Rótulo A_0, o fator $x - 1$ no polinômio gerador é adicionado e a $\lambda^0 = 1$ se torna uma raiz de $g(x)$. Este fator aumenta a distância de projeto do código BCH para 4. Assim, descobre-se que a sequência é identificada somente através do Rótulo A_0 como palavra-código de um código BCH. Os parâmetros do código são:
  - Polinômio gerador: $g(x) = 1 + \beta x + \beta x^2 + \beta x^5 + \beta x^6 + x^7$.
  - Polinômio Minimais que compõem $g(x)$: $pm_0(x) = 1 + x$, $\beta + \beta x + \beta x^2 + x^3$ e $pm_2(x) = \alpha + x + x^2 + x^3$.
  - As raízes de $g(x)$ são: $\{\lambda^0 = 1, \lambda^1, \lambda^4, \lambda^{16}, \lambda^{47}, \lambda^{59}, \lambda^{62}\}$.

  Pela Afirmação 4.11, sabe-se que a diferença entre o Rótulo A e o Rótulo B é o conjugado, e portanto somente quando o Rótulo B_0 é considerado, obtém-se um código BCH com distância de projeto 4 e polinômio gerador: $\overline{g(x)} = 1 + \alpha x + \alpha x^2 + \alpha x^5 + \alpha x^6 + x^7$. Note que o código BCH anterior satisfaz as propriedades descritas na Afirmação 4.13, portanto este código é dito reversível e a sequência lida da direita para esquerda ($nuc_{Rev}(x)$) também é identificada pelo mesmo código BCH. Esta classe de códigos será identificada como **revBCH**.

Para finalizar, a seguinte sequência de nucleotídeos com um SNP é identificada como palavra-código de um código revBCH (Ver Tabela):

Mut: G**C**AAAGCUAAGUACACUUUUAUUUGCUCUGGUCCUCUAUGUCAUAGCCGCAGGAGCUAAUGCA

Observe que a mutação sugerida pelo código revBCH não é uma mutação silenciosa.

Tabela 14 – Análise sobre $\mathbb{F}_4$ da sequência: "gi|217937|Mitocôndria - F1-AT Pase delta subunit"

Código nsBCH (63,57,3) sobre $\mathbb{F}_4$ e qualquer Rótulo

Polinômio Gerador: $g(x) = 1 + x^2 + x^3 + x^5 + x^6$

Rotulamento: $(A, C, G, U) \rightarrow (0, 1, \alpha, \beta)$

Oaa:	**G**	K	L	S	U	L	L	F	A	L	V	L	Y	V	I
Ont:	G**G**A	AAG	CUA	AGU	ACA	CUU	UUA	UUU	GCU	CUG	GUC	CUC	UAU	GUC	AUA
Olb:	$\alpha\boldsymbol{\alpha}0$	00α	$1\beta0$	$0\alpha\beta$	010	$1\beta\beta$	$\beta\beta0$	$\beta\beta\beta$	$\alpha1\beta$	$1\beta\alpha$	$\alpha\beta1$	$1\beta1$	$\beta0\beta$	$\alpha\beta1$	$0\beta0$
Glb:	$\alpha\mathbf{1}0$	00α	$1\beta0$	$0\alpha\beta$	010	$1\beta\beta$	$\beta\beta0$	$\beta\beta\beta$	$\alpha1\beta$	$1\beta\alpha$	$\alpha\beta1$	$1\beta1$	$\beta0\beta$	$\alpha\beta1$	$0\beta0$
Gnt:	G**C**A	AAG	CUA	AGU	ACA	CUU	UUA	UUU	GCU	CUG	GUC	CUC	UAU	GUC	AUA
Gaa:	**A**	K	L	S	U	L	L	F	A	L	V	L	Y	V	I

Oaa:	A	A	G	A	N	A
Ont:	GCC	GCA	GGA	GCU	AAU	GCA
Olb:	$\alpha11$	$\alpha10$	$\alpha\alpha0$	$\alpha1\beta$	00β	$\alpha10$
Glb:	$\alpha11$	$\alpha10$	$\alpha\alpha0$	$\alpha1\beta$	00β	$\alpha10$
Gnt:	GCC	GCA	GGA	GCU	AAU	GCA
Gaa:	A	A	G	A	N	A

4.5 Sequências mRNA Identificadas

Nesta seção, são apresentadas as sequências mRNA identificadas como palavras-código de códigos BCH sobre $\mathbb{Z}_4$ e $\mathbb{F}_4$ e as características e parâmetros desses códigos.

Apesar do algoritmo ser capaz de analisar sequências mRNA de comprimentos que se encontram no seguinte conjunto:

$$\begin{aligned}\{\text{divisores de: } 2^s - 1 \mid s = 1, 2, \dots\} = \{&3, 5, 7, 9, 11, 13, 15, 17, 21, 23, 31, 33, 35, 39, 43, 45,\\ &51, 63, 65, 73, 85, 89, 91, 93, 105, 117, 127, 129, 195,\\ &255, 273, 315, 341, 381, 455, 511, 585, 819, 1023, 1365,\\ &2047, 4095, 5461, 8191, 16383, \dots, \text{entre outros}\},\end{aligned}$$

foram somente consideradas sequências de comprimento 51, 63 e 93, além da sequência de comprimento 1024, identificada como "A. thaliana - Mitocôndria - Malate dehydrogenase 1 – GI: 30695458", da qual obtêm-se novos resultados sobre o alfabeto $\mathbb{F}_4$ e os resultados já conhecidos sobre $\mathbb{Z}_4$ são reproduzidos. Estes comprimentos foram considerados porque o tempo de cômputo do algoritmo é considerável e porque os códigos BCH obtidos são primitivos (comprimentos 63 e 1024) e não primitivos (comprimentos 51 e 93).

No banco de dados do NCBI existem muitas sequências mRNA, entre as quais existem sequências classificadas como: *hypothetical*, *partial*, *predicted*, *patent* e *TSA*; as quais não são consideradas neste trabalho porque estas sequências não são encontradas em nenhum organismo vivo.

Para discriminar estas sequências no NCBI, por exemplo, para sequências mRNA de comprimento 63, o seguinte comando de busca foi utilizado no NCBI:

((((((("biomol mrna"[Properties]) AND **63**[Sequence Length]) NOT hypothetical) NOT partial) NOT predicted) NOT patent) NOT TSA

Com estes comandos as sequências mRNA de comprimento 51, 63 e 93 foram obtidas.

4.5.1 Sequências mRNA identificadas sobre $\mathbb{Z}_4$

A Tabela 16 mostra algumas sequências que foram associadas com algum código BCH sobre o alfabeto $\mathbb{Z}_4$. Na Tabela 17, os resultados obtidos (tipo de código BCH, polinômio gerador, distância de projeto, rótulo e a mutação com sua posição) ao aplicar o algoritmo são armazenados. Estas tabelas estão relacionadas através da coluna *Index*. Assim, por exemplo, a sequência identificada pelo índice 0: "Mus musculus cDNA, clone:Y2G0119C06, strand:unspecified" foi identificada como palavra-código de um código **nsBCH** como se mostra na seguinte tabela:

Tabela 15 – Análise sobre $\mathbb{Z}_4$ da sequência: "Mus musculus cDNA, clone:Y2G0119C06, strand:unspecified".

Código nsBCH (51,42,4) sobre $\mathbb{Z}_4$ e Rótulo C_3

Polinômio Gerador: $g(x) = 3 + 2x + x^2 + 2x^3 + 3x^4 + x^9$

Rotulamento: $(A, C, G, U) \to (3, 1, 0, 2)$

Oaa:	R	Stop	Stop	R	P	P	Stop	E	S	L	**L**	S	R	N	M
Ont:	CGG	UGA	UGA	CGG	CCA	CCA	UAG	GAA	UCG	CUC	**U**UG	AGU	AGG	AAC	AUG
Olb:	100	203	203	100	113	113	230	033	210	121	**2**20	302	300	331	320
Glb:	100	203	203	100	113	113	230	033	210	121	**1**20	302	300	331	320
Gnt:	CGG	UGA	UGA	CGG	CCA	CCA	UAG	GAA	UCG	CUC	**C**UG	AGU	AGG	AAC	AUG
Gaa:	R	Stop	Stop	R	P	P	Stop	E	S	L	**L**	S	R	N	M

Oaa:	V	U
Ont:	GUA	ACG
Olb:	023	310
Glb:	023	310
Gnt:	GUA	ACG
Gaa:	V	T

Ao analisar a Tabela 17, nota-se que todos os códigos BCH associados com as sequências mRNA são do tipo **nsBCH**, indiferentemente dos comprimentos (códigos BCH primitivos e não primitivos). Porém, este fato não garante que quando usado o alfabeto $\mathbb{Z}_4$, somente os códigos nsBCH são capazes de identificar uma sequências mRNA; caso contrario, bastaria um único contraexemplo para tal.

Tabela 16 – Sequências mRNA identificadas como palavras-código de códigos BCH sobre $\mathbb{Z}_4$.

Index	Accession.Version	Detalhes da sequência
		Sequência mRNA
0	AK210649.1	Mus musculus cDNA, clone:Y2G0119C06, strand:unspecified
	CGGTGATGACGGCCACCATAGGAATCGCTCTTGAGTAGGAACATGGTAACG	
1	AJ717925.1	Nicotiana tabacum cDNA-AFLP-fragment BSTT14-85, cultivar Bright Yellow 2
	CAAGGTTTTGGTCTTTCTTTTTTGGAGATTGGTTGTGCTATCTTAGCTCCA	
2	AJ419552.1	Clarkia arcuata mRNA for cytosolic phosphoglucose isomerase, leader region, splice variant 2, strain 8615 LDG
	ACCTTCTTCAGGTACTCTCAGTCATTTTTGCGATTTAGTTCAAGTCTAATG	
3	AJ132114.1	Homo sapiens mRNA for T-cell receptor delta chain, CDR3 region clone D4
	CTTGGGGAACGACGGATCCTATACAGGGGGAGTGGATTGAGAACCGATAAA	
4	NM_001126020.2	Arabidopsis thaliana conserved peptide upstream open reading frame 38 mRNA, complete cds
	ATGTGTATTGCCGTATACCGTAAAGTTTTGAGCTTGAATCTGTATTGCCGTGTGATACTGTAG	
5	NM_001124073.2	Arabidopsis thaliana conserved peptide upstream open reading frame 23 mRNA, complete cds
	ATGGATCGGTGTGACACCTCAGGATTTGTAAGCGGCGGTGGAGATGCTTATAAGGTCTTCTAA	
6	AB642228.1	Solanum tuberosum mRNA, cDNA AFLP fragment TDF41, complete sequence, cultivar: Sarpo Mira
	TAACTCCATATCCTTCAAACTAAGAACCCAATTATAAGGATCACGACCCATATTTGCGAATTG	
7	GU785016.1	Arachis diogoi clone ADAF43 unknown mRNA
	TTACATTTCCATTGCACGTGCATCGACAAATGGCTGCTTCTAAATGCCACCTGTCCTCTCTGC	
8	M17542.1	HUMBCRB Human bcr/abl protein gene (product of translocation t(22q11; 9q34)), exons 1 and 2
	GATGGCGAGGGCGCCTTCCATGGAGACGCAGATGGCTCGTTCGGAACACCACCTGGATACGGC	
9	M17541.1	HUMBCRA Human bcr/abl fusion protein (product of translocation t(22q11; 9q34)), exons 1 and 2
	GATGGCGAGGGCGCCTTCCATGGAGACGCAGAAGCCCTTCAGCGGCCAGTAGCATCTGACTTT	
10	X71032.1	H.sapiens mRNA for T cell receptor joining segment alpha chain, wnVIII.1
	GCAAGCCGGGTATTGGTGAGAATTTTGTCTTTGGTCCCGGAACCAGATTGTCCGTGCTGCCCT	
11	EF209047.1	Palorus ratzeburgii clone 5race68 satellite sequence
	TCTCTGCAAACGCTCTTAAATCATTCGCATTTCGTATTTAAGGCACGATTTTGTTTCGGATTC	
12	AJ617222.1	Nicotiana plumbaginifolia cDNA-AFLP fragment, clone Np464
	TGCTGCAACTATTGCATGGCACATGGACTTCGTGCCATCTGGACGGTATATTTTTGAGATGGG	
13	AJ617187.1	Nicotiana plumbaginifolia cDNA-AFLP fragment, clone Np415
	CTGCGTACCTAATGCCACGAGTTCCCCAGGTCATCCTCTTCCCATTCCTACCATTACTATCCA	
14	AJ617034.1	Nicotiana plumbaginifolia cDNA-AFLP fragment, clone Np197
	ATGTCACAAAGCTCAAGATTCATAGCCCAATCAGGTTCATCCAGGGTTTCAGCTGTGGCTTCC	
15	AK219759.1	Mus musculus cDNA, clone:Y2G0149I10, strand:plus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000055322, based on BLAT search
	CGTAGCGCCCGCCAAGTCTACCTGAGTGACCCCTGCCCTGGCCTCTACCTGGCTGGCCCTGCG	
16	AK218250.1	Mus musculus cDNA, clone:Y2G0144I12, strand:plus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000026743, based on BLAT search

Continua na seguinte página

Tabela 16 – *Continuação da página anterior*

Index	Accession.Version	Detalhes da sequência
		Sequência mRNA
	CGTCAGGAGGCTGTCGCGCACAGACTTGACTGACTACCTCAACAGACATTACAAAGCCCCCCG	
17	AK211528.1	Mus musculus cDNA, clone:Y2G0122B02, strand:plus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000059290, based on BLAT search
	CGGATCACCAGCCTGCTGCTGCAGGTCACCAGTCAGGCAGAAGTCCCATCAGCCATGGCTGCG	
18	AK209415.1	Mus musculus cDNA, clone:Y2G0115A17, strand:minus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000039480, based on BLAT search
	AGCAGAAAAAAGAACTGCTCCAGAAAGGTGCTACTGGTGTCACAAGCACTGAAGGTTGGGTGG	
19	AK206820.1	Mus musculus cDNA, clone:Y2G0106A16, strand:minus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000009705, based on BLAT search
	CGGCTGGTTGTGGGCCAAGTCCTGCATATTCACTAATAAATCAGACATGAAACAAAACAGTCG	
20	AK202522.1	Mus musculus cDNA, clone:Y1G0141M22, strand:minus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000030637, based on BLAT search
	CGGCCAGGGCGTGGCCTCCCCAAGGCTGTGGTGCCCCTTCTGGCTCCCCCAGGTCAGGTCCGC	
21	AK185726.1	Mus musculus cDNA, clone:Y0G0135A12, strand:plus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000020522, based on BLAT search
	CTGAGTAGCAAGATGGGTATGGAGGCCGTGATGGCGCTGCTAGAGGCCACGCCTGACACGCCG	
22	AK185512.1	Mus musculus cDNA, clone:Y0G0134E24, strand:minus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000020775, based on BLAT search
	CGGGAGCAGGGGGATACCGCGGCGGCAGCGGCTGCTGCTGCAGGACGAGCCCAGGGGACACCG	
23	AK182298.1	Mus musculus cDNA, clone:Y0G0121I03, strand:minus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000030724, based on BLAT search
	GTCAGCCAGCCTCGTCCTCTCCCATCCCCAGGTTCATGTGAACTTGCTGCTCCTTGAAGCCCG	
24	AK178647.1	Mus musculus cDNA, clone:Y0G0107H07, strand:unspecified
	CGGCCTCTCTCGGATGTCAGAGGATTCGCCGGGCCGCGGCGGCCTCGGAAGATGGGAGCTGCG	
25	AJ224270.1	Homo sapiens mRNA for T cell receptor beta chain V-D-J junctional region (TCRBV6BJ2S7)
	TGTGCCAGCAGCTTAGTAAAAGGACAGGGGCCCTCGGTAGTAACTTACGAGCAGTACTTCGGG	
26	AJ224229.1	Homo sapiens mRNA for T cell receptor beta chain V-D-J junctional region (TCRBV6BJ2S1)
	TGTGCCAGCAGCTTAGCGACCGGGACTAGCGGGGGGTGGTTGCCAAGCGAGCAGTTCTTCGGG	
27	AM232658.1	Catharanthus roseus cDNA-AFLP fragment, clone CRG434
	TTTTATTATATCTGTTCATTTCCTTTCAGATTTCTTGAATTTTAGTTTACTCAGGACTCATCA	
28	AM232605.1	Catharanthus roseus cDNA-AFLP fragment, clone CRG9
	GTAGTGCAGGCATAATCCACAAGTGTTTCTTGGGAAGAAGAGTTGTTTACTCAGGACTCATCA	
29	DQ460189.1	Nicotiana tabacum cDNA-AFLP fragment H-N_BT3M14-117 sequence
	GCAGTATCCCCATTTACAGACACAGGCTGATATAGTTATTTATGTATTTACTCAGGACTCATC	
30	DQ455269.1	Medicago truncatula cDNA-AFLP fragment BT11M11_200 sequence
	TTTGCTTGGTTTGATGTTACTTTATCTGATCTTAGGAACTCTTAGATATTTTACTCAGGACTC	
31	AJ717897.1	Nicotiana tabacum cDNA-AFLP-fragment BSTT14-100D, cultivar Bright Yellow 2
	GGCGGAATGTAACTGCAAGCGGAGCTCGCCCAAATCCCCAAGGCTCTTTCAGATATGGCTCAA	
32	AJ717848.1	Nicotiana tabacum cDNA-AFLP-fragment BSTT12-2-100, cultivar Bright Yellow 2
	ATACGAAGGTTCAGTGCTAGTAGCTGAACCCCGTTGCTTGGGAATTGATAGTTTGGGTGACAG	
33	X04549.1	Beef heart mitochondrial 3S transfer RNA-Ser (GCU)

Continua na seguinte página

Tabela 16 – *Continuação da página anterior*

Index	Accession.Version	Detalhes da sequência
		Sequência mRNA
	GAAAAAGTATGCAAGAACTGCTAATTCTATGCTCCCATATCTAATAGTATGGCTTTTTCGCCA	
34	D28445.1	HUMCN104 Homo sapiens mRNA for 2',3'-cyclic nucleotide 3'-phosphodiesterase, 5'UTR region
	AACAGAGGCTTCTCCCGAAAAAGCCACACATTCCTGCCCAAGATCTTCTTCCGCAAGATGTCA	
35	AY431119.1	Aedes aegypti ASAP ID: 37269 unknown mRNA sequence
	AGTCAAAATCGTGTTCGTGTTTTAACAAACAGATTCAAATCTAATAACTATTAAAGAAATAAA	
36	Z49946.1	H.sapiens mRNA for T cell receptor alpha chain region (TCRAV4S1AJ13S2)
	TGCTGTGTACTACTGCATCCTGAGAGACAAGACGATAACTATGGTCAGAATTTTGTCTTTGGT	
37	X60144.1	Human J-alpha segment J-alpha FR8 mRNA for J-alpha region of T-cell receptor
	AAATCGGTGAATAGGCAAACAACCTCTTCTTTGGGACTGGAACGAGACTCACCGTTATTCCCT	
38	U75788.1	REU75788 Raja eglanteria T cell antigen receptor gamma mRNA, junctional region
	TATTACTGTGCTTATTGCAAAAGAGGGGTATCAGGTACATGGAGAAAAATATTCGGCAGCGGG	
39	U75780.1	REU75780 Raja eglanteria T cell antigen receptor beta mRNA, junctional region
	TATTTTTGCGCTGCTAAAGAAGCGCGGGCAGGAAACAATGCAGAAGCCTATTTCGGAAAGGGA	
40	U75776.1	REU75776 Raja eglanteria T cell antigen receptor beta mRNA, junctional region
	TATTTTTGCGCTGCTAAAGAAGCTATAAACGGTCGGACCAATGAAGCGATCTTCGGCAGCGGG	
41	X72130.1	H.sapiens (patient HoP, clone 1) mRNA for T-cell receptor delta chain V-J region
	GCTCTTGGGGATCCACACGCCAAATCTTTGCGAATATACCCTGGGGGATACTCCGATAAACTC	
42	Z27187.1	H.sapiens rearranged mRNA for TCR delta chain (VJ)
	GCTCTTGGGGACCCCTCCAATCTTCCTACGATAAGGGGGATACCGCCGTACACCGATAAACTC	
43	X74018.1	H.sapiens (1) Vdelta1-Jdelta1 mRNA for TCR delta
	GCTCTTGGGGAACTCGGCCTGCCTCGTCCTCCTTACTGGGGGATCCCTCCGACCGATAAACTC	
44	X14937.1	Mouse mRNA for T-cell receptor beta-chain V(beta)14-J(beta)2.2
	TGTGCCTGGAGTCTAGCGGGGGAGCAGCTCTACTTTGGTGAAGGCTCAAAGCTGACAGTGCTG	
45	X14936.1	Mouse mRNA for T-cell receptor beta-chain V(beta)14-J(beta)1.1
	TGTGCCTGGAGTCTCAGACAGAACACAGAAGTCTTCTTTGGTAAAGGAACCAGACTCGTTGTA	
46	X02975.1	Mouse mRNA fragment for T-cell receptor alpha chain J-C (TA 20)
	CTCAGCAGCCTCTTCTTTGGTGATGGGACGCAGCTGGTGGTGAAGCCCAACATCCAGAACCCA	
47	X74012.1	H.sapiens (10) Vdelta1-Jdelta1 mRNA for TCR delta
	GCTCTTGGGGATACCTCGAGTAGCCATTACTGGATCGGCTGGCCTTCATCCACCGATAAACTC	
48	L32451.1	HUMTCVD1DB Human (clone: 3cfa09) T-cell receptor delta-chain (V-delta-1) mRNA
	CTCTTGGGGAACCCCCCTCTTCCTTTACTCTCCCTGTAGGCAGATGGTACACCGATCTTTGGA	
49	L06870.1	HUMTCRAJE Human rearranged T-cell receptor alpha-chain joining region (TCRA) mRNA
	GTGAGAGGGGAGAACTTCAACAAATTTTACTTTGGATCTGTGACCAAACTCAATGTAAAACCA	
50	AK253422.1	Gryllus bimaculatus mRNA, GBcontig00283

Continua na seguinte página

Tabela 16 – *Continuação da página anterior*

Index	**Accession.Version**	**Detalhes da sequência**
	Sequência mRNA	
	ATCGCCTCGACAGTCGGTCTTTGGCATGGATGTAGCGGAAACGCTGCACGTAGTAGAAGACCAGCCACGCCAAGGAAATCATCATGAGGACGA	
51	AK262126.1	Gryllus bimaculatus mRNA, GBcontig08987
	CATCTGATGCAACATGGAGAAATTCCTGTAATACATACTAACAGAGCAAATATTTGTGTGTGCAGTTAAGTAGGCTCAGAGATTTTTAGTGAA	
52	NM_104202.2	Arabidopsis thaliana malate dehydrogenase 1
	ATGTTCAGATCTATGCTCGTCCGATCTTCTGCCT...GAACTCAAGTCCTCCATAGAAAAGGGAGTCAAGTTTGCCAACCAG	

Tabela 17 – Propriedades dos códigos BCH sobre $\mathbb{Z}_4$ que identificam as sequências mRNA da Tabela 16 como palavras-código. A coluna *Index* relaciona cada uma das sequências mRNA com os códigos obtidos, **N** é comprimento da sequência, **Pos** indica a posição da mutação e **Mut** especifica a mutação.

Index	**N**	**Pos**	**Mut**	**Rotulo**	$g(x)$	δ	**Tipo**
0	51	30	T→C	C_3	$3+2x+x^2+2x^3+3x^4+x^9$	4	nsBCH
1	51	24	G→A	B_2	$3+x^5+2x^6+3x^7+2x^8+x^9$	4	nsBCH
2	51	19	A→G	C_2	$3+2x+x^2+2x^3+3x^4+x^9$	4	nsBCH
3	51	28	G→A	A_1	$3+2x+x^2+2x^3+3x^4+x^9$	4	nsBCH
4	63	38	T→C	A_0	$3+x+2x^3+2x^4+x^5+2x^6+x^7$	4	nsBCH
5	63	21	G→A	B_2	$3+3x+x^2+x^4+x^5+2x^6+x^7$	4	nsBCH
5	63	12	G→T	C_0	$3+3x+x^2+x^4+x^5+2x^6+x^7$	4	nsBCH
5	63	51	A→T	A_0	$3+3x+x^2+x^4+x^5+2x^6+x^7$	4	nsBCH
6	63	18	A→T	C_1	$3+x^3+2x^4+3x^5+2x^6+x^7$	4	nsBCH
7	63	15	A→G	B_2	$3+x^3+2x^4+3x^5+2x^6+x^7$	4	nsBCH
8	63	24	G→C	C_1	$3+2x+3x^2+3x^3+3x^5+x^6+x^7$	4	nsBCH
9	63	62	T→G	A_3	$3+x+2x^3+2x^4+x^5+2x^6+x^7$	4	nsBCH
10	63	38	G→T	A_1	$3+2x+3x^2+2x^3+2x^4+3x^6+x^7$	4	nsBCH
11	63	51	T→C	A_1	$3+x+2x^3+2x^4+x^5+2x^6+x^7$	4	nsBCH
12	63	21	C→G	A_0	$3+3x+x^2+x^4+x^5+2x^6+x^7$	4	nsBCH
12	63	47	A→G	C_3	$3+3x+x^2+x^4+x^5+2x^6+x^7$	4	nsBCH
13	63	56	C→T	A_3	$3+x+2x^3+2x^4+x^5+2x^6+x^7$	4	nsBCH
14	63	3	T→C	B_0	$3+3x+x^2+x^4+x^5+2x^6+x^7$	4	nsBCH
14	63	29	A→T	A_0	$3+2x+3x^2+3x^3+3x^5+x^6+x^7$	4	nsBCH
15	63	6	G→T	C_1	$3+3x+x^2+x^4+x^5+2x^6+x^7$	4	nsBCH
16	63	54	A→C	B_1	$3+2x+x^2+2x^3+3x^4+x^7$	4	nsBCH
17	63	11	C→A	C_2	$3+3x+x^2+x^4+x^5+2x^6+x^7$	4	nsBCH
18	63	51	A→T	C_2	$3+2x+3x^2+3x^3+3x^5+x^6+x^7$	4	nsBCH
19	63	46	A→C	C_2	$3+2x+x^2+2x^3+3x^4+x^7$	4	nsBCH
19	63	4	T→G	B_3	$3+2x+3x^2+2x^3+2x^4+3x^6+x^7$	4	nsBCH
20	63	0	C→A	B_0	$3+x+2x^3+2x^4+x^5+2x^6+x^7$	4	nsBCH
20	63	53	T→A	A_1	$3+x+2x^3+2x^4+x^5+2x^6+x^7$	4	nsBCH
21	63	30	A→T	A_2	$3+x+2x^3+2x^4+x^5+2x^6+x^7$	4	nsBCH
22	63	26	A→T	B_0	$3+x+2x^3+2x^4+x^5+2x^6+x^7$	4	nsBCH
22	63	33	G→C	C_0	$3+x^3+2x^4+3x^5+2x^6+x^7$	4	nsBCH
23	63	29	A→T	B_2	$3+2x+3x^2+2x^3+2x^4+3x^6+x^7$	4	nsBCH
24	63	28	C→G	A_3	$3+x+2x^3+2x^4+x^5+2x^6+x^7$	4	nsBCH
25	63	39	G→C	C_0	$3+2x+x^2+2x^3+3x^4+x^7$	4	nsBCH
26	63	35	G→T	A_0	$3+2x+x^2+2x^3+3x^4+x^7$	4	nsBCH
27	63	58	C→T	C_1	$3+2x+3x^2+2x^3+2x^4+3x^6+x^7$	4	nsBCH
27	63	12	T→G	C_2	$3+x+2x^3+2x^4+x^5+2x^6+x^7$	4	nsBCH
28	63	54	G→T	B_2	$3+2x+3x^2+3x^3+3x^5+x^6+x^7$	4	nsBCH
29	63	16	C→A	B_0	$3+x^3+2x^4+3x^5+2x^6+x^7$	4	nsBCH
29	63	24	G→T	B_2	$3+2x+x^2+2x^3+3x^4+x^7$	4	nsBCH
29	63	28	G→A	C_0	$3+2x+3x^2+3x^3+3x^5+x^6+x^7$	4	nsBCH
30	63	50	T→G	B_2	$3+2x+x^2+2x^3+3x^4+x^7$	4	nsBCH
30	63	62	C→G	C_1	$3+x+2x^3+2x^4+x^5+2x^6+x^7$	4	nsBCH
30	63	18	A→C	B_0	$3+3x+x^2+x^4+x^5+2x^6+x^7$	4	nsBCH
31	63	2	C→T	B_0	$3+3x+x^2+x^4+x^5+2x^6+x^7$	4	nsBCH
32	63	48	T→C	B_2	$3+x^3+2x^4+3x^5+2x^6+x^7$	4	nsBCH
33	63	10	G→T	A_2	$3+2x+3x^2+2x^3+2x^4+3x^6+x^7$	4	nsBCH
34	63	6	G→T	C_1	$3+2x+3x^2+2x^3+2x^4+3x^6+x^7$	4	nsBCH
35	63	24	A→C	C_2	$3+2x+3x^2+2x^3+2x^4+3x^6+x^7$	4	nsBCH
36	63	46	A→T	A_0	$3+x^3+2x^4+3x^5+2x^6+x^7$	4	nsBCH
36	63	21	G→C	C_1	$3+3x+x^2+x^4+x^5+2x^6+x^7$	4	nsBCH
36	63	58	T→G	C_2	$3+2x+3x^2+3x^3+3x^5+x^6+x^7$	4	nsBCH

Continua na seguinte página

Tabela 17 – *Continuação da página anterior*

Index	**N**	**Pos**	**Mut**	**Rotulo**	$g(x)$	δ	**Tipo**
37	63	6	G→A	B_0	$3+3x+x^2+x^4+x^5+2x^6+x^7$	4	nsBCH
38	63	53	C→A	A_0	$3+x^3+2x^4+3x^5+2x^6+x^7$	4	nsBCH
39	63	37	A→G	B_2	$3+2x+3x^2+2x^3+2x^4+3x^6+x^7$	4	nsBCH
39	63	13	C→A	B_3	$3+2x+3x^2+3x^3+3x^5+x^6+x^7$	4	nsBCH
40	63	52	T→C	A_0	$3+2x+x^2+2x^3+3x^4+x^7$	4	nsBCH
41	63	51	T→C	B_1	$3+2x+3x^2+3x^3+3x^5+x^6+x^7$	4	nsBCH
42	63	2	T→A	A_3	$3+2x+3x^2+3x^3+3x^5+x^6+x^7$	4	nsBCH
43	63	53	C→T	B_2	$3+x^3+2x^4+3x^5+2x^6+x^7$	4	nsBCH
44	63	13	T→A	A_2	$3+2x+x^2+2x^3+3x^4+x^7$	4	nsBCH
44	63	46	C→A	B_3	$3+2x+x^2+2x^3+3x^4+x^7$	4	nsBCH
44	63	52	T→G	B_3	$3+x^3+2x^4+3x^5+2x^6+x^7$	4	nsBCH
45	63	18	C→A	B_0	$3+2x+3x^2+3x^3+3x^5+x^6+x^7$	4	nsBCH
46	63	8	C→T	B_2	$3+3x+x^2+x^4+x^5+2x^6+x^7$	4	nsBCH
47	63	53	C→G	B_2	$3+x^3+2x^4+3x^5+2x^6+x^7$	4	nsBCH
48	63	50	A→C	A_3	$3+2x+x^2+2x^3+3x^4+x^7$	4	nsBCH
49	63	0	G→T	B_2	$3+2x+3x^2+3x^3+3x^5+x^6+x^7$	4	nsBCH
50	93	46	C→G	C_0	$3+2x+x^2+x^3+x^4+x^5+x^6+x^8+x^{11}$	4	nsBCH
51	93	89	T→G	B_0	$3+3x+x^2+x^3+2x^5+3x^7+x^8+3x^9+2x^{10}+x^{11}$	4	nsBCH
52	93	867	C→T	C_2	$3+2x+3x^2+3x^3+x^5+3x^6+2x^7+2x^8+x^{11}$	4	nsBCH

4.5.2 Sequências mRNA identificadas sobre $\mathbb{F}_4$

A Tabela 19 mostra todas as sequências que foram associadas com algum código BCH sobre o alfabeto $\mathbb{F}_4$. Na Tabela 20, os resultados obtidos (tipo de código BCH, polinômio gerador, distância de projeto, rótulo e a mutação com sua posição) ao aplicar o algoritmo são armazenados. Estas tabelas estão relacionadas através da coluna *Index*. Assim, por exemplo, a sequência identificada pelo índice 0: "Homo sapiens mRNA for T cell receptor beta chain V-D-J junctional region (TCRBV6BJ2S3)" foi identificada como palavra-código de um código BCH como se mostra na seguinte tabela:

Tabela 18 – Análise sobre $\mathbb{F}_4$ da sequência: "Homo sapiens mRNA for T cell receptor beta chain V-D-J junctional region (TCRBV6BJ2S3)"

Código nsBCH (51,42,4) sobre $\mathbb{F}_4$ e Rótulo A_3

Polinômio Gerador: $g(x) = 1 + \beta x^2 + \alpha x^4 + x^9$

Rotulamento: $(A, C, G, U/T) \to (\beta, \alpha, 1, 0)$

Oaa:	C	A	S	E	R	D	P	S	W	G	T	D	T	Q	Y
Ont:	TGT	GCC	AGT	GAA	AGG	GAC	CCG	TCC	TGG	GGC	ACA	GAT	ACG	CAG	TAT
Olb:	010	$1\alpha\alpha$	$\beta 10$	$1\beta\beta$	$\beta 11$	$1\beta\alpha$	$\alpha\alpha 1$	$0\alpha\alpha$	011	11α	$\beta\alpha\beta$	$1\beta 0$	$\beta\alpha 1$	$\alpha\beta 1$	$0\beta 0$
Glb:	010	$1\alpha\alpha$	$\beta 10$	$\beta\beta\beta$	$\beta 11$	$1\beta\alpha$	$\alpha\alpha 1$	$0\alpha\alpha$	011	11α	$\beta\alpha\beta$	$1\beta 0$	$\beta\alpha 1$	$\alpha\beta 1$	$0\beta 0$
Gnt:	TGT	GCC	AGT	AAA	AGG	GAC	CCG	TCC	TGG	GGC	ACA	GAT	ACG	CAG	TAT
Gaa:	C	A	S	K	R	D	P	S	W	G	T	D	T	Q	Y

Oaa:	F	G
Ont:	TTT	GGC
Olb:	000	11α
Glb:	000	11α
Gnt:	TTT	GGC
Gaa:	F	G

Quando aplicado o algoritmo proposto neste capítulo sobre o alfabeto $\mathbb{F}_4$, observou-se que este foi capaz de associar sequências mRNA com códigos BCH, diferentes dos códigos nsBCH. Estas associações são uma novidade e permitem identificar uma estrutura matemática para um conjunto maior de sequências. Entre as sequências mRNA identificadas se encontram aquelas que pertencem aos códigos revBCH (reversíveis) tais que quando lidas da direita para a esquerda pertencem ao mesmo código BCH que quando lidas da esquerda para a direita.

Uma sequência muito interessante para trabalhos futuros é a sequência: "Arabidopsis thaliana malate dehydrogenase 1". Em (BRANDÃO *et al.*, 2015), estudou-se a possibilidade que a mutação obtida (ver Tabela 17, *index*: 52) sobre o alfabeto $\mathbb{Z}_4$ sugere uma correlação entre a sequência detetada e uma forma ancestral dessa mesma proteína. No presente trabalho identificou-se que a sequência "Arabidopsis thaliana malate dehydrogenase 1" também está associada a um código BCH sobre $\mathbb{F}_4$ (ver Tabela 20, *index*: 52), possivelmente a mutação sugerida pelo código BCH sobre $\mathbb{F}_4$ também está relacionada com alguma outra sequência ancestral da proteína.

Por último, observa-se que as sequências mRNA identificadas como palavras-código sobre $\mathbb{F}_4$ estão distribuídas entre as diferentes classes de códigos BCH (nsBCH, revBCH e BCH), fenômeno diferente ao observado quando analisadas sobre $\mathbb{Z}_4$ (todas as sequência foram identificadas como palavras-código de códigos nsBCH).

Tabela 19 – Sequências mRNA identificadas como palavras-código de códigos BCH sobre $\mathbb{F}_4$.

Index	**Accession.Version**	**Detalhes da sequência**
		Sequência mRNA
0	AJ224238.1	Homo sapiens mRNA for T cell receptor beta chain V-D-J junctional region (TCRBV6BJ2S3)
	TGTGCCAGTGAAAGGGACCCGTCCTGGGGCACAGATACGCAGTATTTTGGC	
1	Z69459.1	H.sapiens mRNA for T cell receptor beta chain junctional region (clone K60)
	TGTGCCAGCACCTTCCGGGGACAGGGCTTAAATCAGCCCCAGCATTTTGGT	
2	Z69419.1	H.sapiens mRNA for T cell receptor beta chain junctional region (clone K4)
	TGTGCCAGCAGTTTAGGGTCAGGGGGAAATCATCAGCCCCAGCATTTTGGT	
3	Z69474.1	H.sapiens mRNA for T cell receptor beta chain junctional region (clone K3-4)
	TGTGCCAGCAGCCTTGGGACAGGGGGGAGCAATCAGCCCCAGCATTTTGGT	
4	AJ717925.1	Nicotiana tabacum cDNA-AFLP-fragment BSTT14-85, cultivar Bright Yellow 2
	CAAGGTTTTGGTCTTTCTTTTTTGGAGATTGGTTGTGCTATCTTAGCTCCA	
5	X58757.1	Human mRNA for T cell receptor J alpha gene segment IGRJa10
	ATAACAATGACATGCGCTTTGGAGCAGGGACCAGACTGACAGTAAAACCAA	
6	NM_001126020.2	Arabidopsis thaliana conserved peptide upstream open reading frame 38 mRNA, complete cds
	ATGTGTATTGCCGTATACCGTAAAGTTTTGAGCTTGAATCTGTATTGCCGTGTGATACTGTAG	
7	NM_001124073.2	Arabidopsis thaliana conserved peptide upstream open reading frame 23 mRNA, complete cds
	ATGGATCGGTGTGACACCTCAGGATTTGTAAGCGGCGGTGGAGATGCTTATAAGGTCTTCTAA	
8	AB642228.1	Solanum tuberosum mRNA, cDNA AFLP fragment TDF41, complete sequence, cultivar: Sarpo Mira
	TAACTCCATATCCTTCAAACTAAGAACCCAATTATAAGGATCACGACCCATATTTGCGAATTG	
9	AB642198.1	Solanum tuberosum mRNA, cDNA AFLP fragment TDF11, complete sequence, cultivar: Bintje
	AATTCAGAGACAGTTCAAGTGATGAAGATTCATCTCCAAAGAAACCAAGAGAAGAACACATTA	
10	GU785016.1	Arachis diogoi clone ADAF43 unknown mRNA
	TTACATTTCCATTGCACGTGCATCGACAAATGGCTGCTTCTAAATGCCACCTGTCCTCTCTGC	
11	M14399.1	ECOALKP E.coli alkaline phosphatase signal mRNA, 5' end
	GTGAAACAAAGCACTATTGCACTGGCTGTCTTACCGTTACTGTTTACCCCTGTGACAAAAGCC	
12	BT006508.1	Arabidopsis thaliana At2g34650 gene, complete cds
	ATGAAATCGGCGACGTTTAGTGGTAGAAGTAGTAACAAACCAGCGGCGTTCGATTACTTTTGA	
13	AF254835.1	Pyrococcus abyssi box C/D small nucleolar RNA sR20
	TGTTGATGATGAACTCGCTTGTTGCTGAAGGGTGATGAAGGGCATACTGGCTGATTTGGAGGT	
14	AJ617222.1	Nicotiana plumbaginifolia cDNA-AFLP fragment, clone Np464
	TGCTGCAACTATTGCATGGCACATGGACTTCGTGCCATCTGGACGGTATATTTTTGAGATGGG	
15	AJ617187.1	Nicotiana plumbaginifolia cDNA-AFLP fragment, clone Np415
	CTGCGTACCTAATGCCACGAGTTCCCCAGGTCATCCTCTTCCCATTCCTACCATTACTATCCA	
16	AJ617034.1	Nicotiana plumbaginifolia cDNA-AFLP fragment, clone Np197

Continua na seguinte página

Tabela 19 – *Continuação da página anterior*

Index	Accession.Version	Detalhes da sequência
		Sequência mRNA
	ATGTCACAAAGCTCAAGATTCATAGCCCAATCAGGTTCATCCAGGGTTTCAGCTGTGGCTTCC	
17	AK219759.1	Mus musculus cDNA, clone:Y2G0149I10, strand:plus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000055322, based on BLAT search
	CGTAGCGCCCGCCAAGTCTACCTGAGTGACCCCTGCCCTGGCCTCTACCTGGCTGGCCCTGCG	
18	AK219739.1	Mus musculus cDNA, clone:Y2G0149H05, strand:plus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000067989, based on BLAT search
	CGGAGGAGCACCTCATCCTAAAGACGACATTTGAGGACCTTATCCAGCGCTGCCTCTCCTCCG	
19	AK218250.1	Mus musculus cDNA, clone:Y2G0144I12, strand:plus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000026743, based on BLAT search
	CGTCAGGAGGCTGTCGCGCACAGACTTGACTGACTACCTCAACAGACATTACAAAGCCCCCCG	
20	AK211528.1	Mus musculus cDNA, clone:Y2G0122B02, strand:plus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000059290, based on BLAT search
	CGGATCACCAGCCTGCTGCTGCAGGTCACCAGTCAGGCAGAAGTCCCATCAGCCATGGCTGCG	
21	AK211210.1	Mus musculus cDNA, clone:Y2G0121A03, strand:minus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000049108, based on BLAT search
	CGGAGCAGAAGACAGAGCTAGTGTGCGAACTGCAGAGGCTTCAGTACTGTGTGGGCATGTGCG	
22	AK209415.1	Mus musculus cDNA, clone:Y2G0115A17, strand:minus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000039480, based on BLAT search
	AGCAGAAAAAAGAACTGCTCCAGAAAGGTGCTACTGGTGTCACAAGCACTGAAGGTTGGGTGG	
23	AK206820.1	Mus musculus cDNA, clone:Y2G0106A16, strand:minus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000009705, based on BLAT search
	CGGCTGGTTGTGGGCCAAGTCCTGCATATTCACTAATAAATCAGACATGAAACAAAACAGTCG	
24	AK202522.1	Mus musculus cDNA, clone:Y1G0141M22, strand:minus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000030637, based on BLAT search
	CGGCCAGGGCGTGGCCTCCCCAAGGCTGTGGTGCCCCTTCTGGCTCCCCCAGGTCAGGTCCGC	
25	AK185726.1	Mus musculus cDNA, clone:Y0G0135A12, strand:plus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000020522, based on BLAT search
	CTGAGTAGCAAGATGGGTATGGAGGCCGTGATGGCGCTGCTAGAGGCCACGCCTGACACGCCG	
26	AK184887.1	Mus musculus cDNA, clone:Y0G0131N15, strand:plus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000065254, based on BLAT search
	CGGTGGGACTCTCGCCACCGCCACCGCCGCCTCGGAGAGCAGGCCGCTACCGCGCGTACACCG	
27	AK182298.1	Mus musculus cDNA, clone:Y0G0121I03, strand:minus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000030724, based on BLAT search
	GTCAGCCAGCCTCGTCCTCTCCCATCCCCAGGTTCATGTGAACTTGCTGCTCCTTGAAGCCCG	
28	AK181982.1	Mus musculus cDNA, clone:Y0G0120F14, strand:plus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000066769, based on BLAT search
	CGCCATCTTGTCTCCCTCTCCTTACCTGCGTCCTCGGGGGCGTGCTCACCACCACCCCTCCCG	
29	AK179359.1	Mus musculus cDNA, clone:Y0G0110B13, strand:plus, reference:ENSEMBL:Mouse-Transcript-ENST:ENSMUST00000025684, based on BLAT search
	CGGAGCCCACCACCGACTCCTTCATCGCGGTCATGCACGGCCCCACCGAGGGCGTGGTGCCCG	
30	AK178647.1	Mus musculus cDNA, clone:Y0G0107H07, strand:unspecified
	CGGCCTCTCTCGGATGTCAGAGGATTCGCCGGGCCGCGGCGGCCTCGGAAGATGGGAGCTGCG	
31	D10490.1	HUMNF1IS Homo sapiens mRNA for insertion sequence in neurofibromatosis type 1 gene transcript, coding for GTPase-activating protein related domain
	GCAACTTGCCACTCCCTACTGAATAAAGCTACAGTAAAAGAAAAAAAGGAAAACAAAAAATCA	
32	AJ224229.1	Homo sapiens mRNA for T cell receptor beta chain V-D-J junctional region (TCRBV6BJ2S1)
	TGTGCCAGCAGCTTAGCGACCGGGACTAGCGGGGGGTGGTTGCCAAGCGAGCAGTTCTTCGGG	
33	AM232658.1	Catharanthus roseus cDNA-AFLP fragment, clone CRG434

Continua na seguinte página

Tabela 19 – *Continuação da página anterior*

Index	Accession.Version	Detalhes da sequência
		Sequência mRNA
		TTTTATTATATCTGTTCATTTCCTTTCAGATTTCTTGAATTTTAGTTTACTCAGGACTCATCA
34	AM232605.1	Catharanthus roseus cDNA-AFLP fragment, clone CRG9
		GTAGTGCAGGCATAATCCACAAGTGTTTCTTGGGAAGAAGAGTTGTTTACTCAGGACTCATCA
35	AM232521.1	Catharanthus roseus cDNA-AFLP fragment, clone CRG263
		GATTTGCTCTGGTTCAGCCTTCTTCATGGCATTTTTTCAAGGTGATGTTACTCAGGACTCATC
36	AM232508.1	Catharanthus roseus cDNA-AFLP fragment, clone CRG194
		TATGCTGCCAGCTACGTACGAGGACTTCAAGTAGCGTTGGAAATAGACTTACTCAGGACTCAT
37	AM232418.1	Catharanthus roseus cDNA-AFLP fragment, clone CRG426
		CATGGAATTGGCTGCAAAACAGGCCTATGCGCATGATTTCATCGCTTCCCTTCCAATGGTTAT
38	DQ460189.1	Nicotiana tabacum cDNA-AFLP fragment H-N_BT3M14-117 sequence
		GCAGTATCCCCATTTACAGACACAGGCTGATATAGTTATTTATGTATTTACTCAGGACTCATC
39	AJ717897.1	Nicotiana tabacum cDNA-AFLP-fragment BSTT14-100D, cultivar Bright Yellow 2
		GGCGGAATGTAACTGCAAGCGGAGCTCGCCCAAATCCCCAAGGCTCTTTCAGATATGGCTCAA
40	AJ717870.1	Nicotiana tabacum cDNA-AFLP-fragment BSTT12-4-215, cultivar Bright Yellow 2
		ACCTTGGAGAGGGTAGGAGGGCATGTCTTTATACATGTGCTTGTTTGCTATTCTGACAATGGA
41	AJ717848.1	Nicotiana tabacum cDNA-AFLP-fragment BSTT12-2-100, cultivar Bright Yellow 2
		ATACGAAGGTTCAGTGCTAGTAGCTGAACCCCGTTGCTTGGGAATTGATAGTTTGGGTGACAG
42	V00663.1	Hamster mitochondrion 3S E RNA, possible 5S rRNA equivalent
		GGAGAATGTATGCAAGAGCTGCTAACTCCTGCTACCATGTATAATAACATGGCTTTCTTACCA
43	X04549.1	Beef heart mitochondrial 3S transfer RNA-Ser (GCU)
		GAAAAAGTATGCAAGAACTGCTAATTCTATGCTCCCATATCTAATAGTATGGCTTTTTCGCCA
44	D28445.1	HUMCN104 Homo sapiens mRNA for 2',3'-cyclic nucleotide 3'-phosphodiesterase, 5'UTR region
		AACAGAGGCTTCTCCCGAAAAAGCCACACATTCCTGCCCAAGATCTTCTTCCGCAAGATGTCA
45	AY431119.1	Aedes aegypti ASAP ID: 37269 unknown mRNA sequence
		AGTCAAAATCGTGTTCGTGTTTTAACAAACAGATTCAAATCTAATAACTATTAAAGAAATAAA
46	X86712.1	P.troglodytes mRNA for TcR gamma chain, clone 24
		TCATACTGTGCTGCGTGGGATTGGAAAAAACTCTTTGGCAGTGGAACAACACTTGTTGTCACA
47	Z49946.1	H.sapiens mRNA for T cell receptor alpha chain region (TCRAV4S1AJ13S2)
		TGCTGTGTACTACTGCATCCTGAGAGACAAGACGATAACTATGGTCAGAATTTTGTCTTTGGT
48	X60144.1	Human J-alpha segment J-alpha FR8 mRNA for J-alpha region of T-cell receptor
		AAATCGGTGAATAGGCAAACAACCTCTTCTTTGGGACTGGAACGAGACTCACCGTTATTCCCT
49	U75780.1	REU75780 Raja eglanteria T cell antigen receptor beta mRNA, junctional region
		TATTTTTGCGCTGCTAAAGAAGCGCGGGCAGGAAACAATGCAGAAGCCTATTTCGGAAAGGGA
50	X72130.1	H.sapiens (patient HoP, clone 1) mRNA for T-cell receptor delta chain V-J region

Continua na seguinte página

Tabela 19 – *Continuação da página anterior*

Index	Accession.Version	Detalhes da sequência
	Sequência mRNA	
	GCTCTTGGGGATCCACACGCCAAATCTTTGCGAATATACCCTGGGGGATACTCCGATAAACTC	
51	Z27187.1	H.sapiens rearranged mRNA for TCR delta chain (VJ)
	GCTCTTGGGGACCCCTCCAATCTTCCTACGATAAGGGGGATACCGCCGTACACCGATAAACTC	
52	X74018.1	H.sapiens (1) Vdelta1-Jdelta1 mRNA for TCR delta
	GCTCTTGGGGAACTCGGCCTGCCTCGTCCTCCTTACTGGGGGATCCCTCCGACCGATAAACTC	
53	X14937.1	Mouse mRNA for T-cell receptor beta-chain V(beta)14-J(beta)2.2
	TGTGCCTGGAGTCTAGCGGGGGAGCAGCTCTACTTTGGTGAAGGCTCAAAGCTGACAGTGCTG	
54	X14936.1	Mouse mRNA for T-cell receptor beta-chain V(beta)14-J(beta)1.1
	TGTGCCTGGAGTCTCAGACAGAACACAGAAGTCTTCTTTGGTAAAGGAACCAGACTCGTTGTA	
55	X02975.1	Mouse mRNA fragment for T-cell receptor alpha chain J-C (TA 20)
	CTCAGCAGCCTCTTCTTTGGTGATGGGACGCAGCTGGTGGTGAAGCCCAACATCCAGAACCCA	
56	X74012.1	H.sapiens (10) Vdelta1-Jdelta1 mRNA for TCR delta
	GCTCTTGGGGATACCTCGAGTAGCCATTACTGGATCGGCTGGCCTTCATCCACCGATAAACTC	
57	L06870.1	HUMTCRAJE Human rearranged T-cell receptor alpha-chain joining region (TCRA) mRNA
	GTGAGAGGGGAGAACTTCAACAAATTTTACTTTGGATCTGTGACCAAACTCAATGTAAAACCA	
58	AK255393.1	Gryllus bimaculatus mRNA, GBcontig02254
	CGCGCCGGCGGCTTGTCCATGCTGATGTGGTAGCCCAGCGCCAGGATGGTTTTGAGCGTCTGCACGGCCAGCTGCGAGTCGTAGCGCTTCTCC	
59	NM_104202.2	Arabidopsis thaliana malate dehydrogenase 1
	ATGTTCAGATCTATGCTCGTCCGATCTTCTGCCT...GAACTCAAGTCCTCCATAGAAAAGGGAGTCAAGTTTGCCAACCAG	

Tabela 20 – Propriedades dos códigos BCH sobre $\mathbb{F}_4$ que identificam as sequências mRNA da Tabela 19 como palavras-código. A coluna *Index* relaciona cada uma das sequências mRNA com os códigos obtidos, **N** é comprimento da sequência, **Pos** indica a posição da mutação e **Mut** especifica a mutação.

Index	**N**	**Pos**	**Mut**	**Rotulo**	$g(x)$	δ	**Tipo**
0	51	9	G→A	A_3	$1+\beta x^2+\alpha x^4+x^9$	4	BCH
1	51	41	G→C	A_2	$1+x+\alpha x^2+x^4+\alpha x^6+\alpha x^8+x^{10}+\alpha x^{11}+x^{12}$	5	BCH
2	51	41	G→C	A_1	$1+x^5+x^7+x^9$	4	nsBCH
3	51	38	C→T	A_3	$1+\alpha x^2+x^3+x^4+x^5+x^6+\alpha x^7+x^9$	4	revBCH
4	51	41	C→T	A_2	$1+x^5+x^7+x^9$	4	nsBCH
5	51	9	A→G	A_3	$1+\beta x+x^4+x^5+\beta x^8+x^9$	4	revBCH
6	63	58	T→C	A_3	$1+x+x^2+\beta x^3+\beta x^4+x^5+x^6+x^7$	4	revBCH
7	63	0	A→A	A_0	$1+x+x^2+x^4+x^5+x^7$	4	nsBCH
8	63	51	A→C	A_0	$\beta+x+\alpha x^2+x^3+\alpha x^4+\alpha x^5+\beta x^6+x^7$	4	BCH
9	63	29	T→G	A_2	$\alpha+\beta x^2+\alpha x^3+\beta x^5+\alpha x^6+x^7$	4	BCH
9	63	8	G→C	A_0	$1+\alpha x+x^2+x^3+x^4+x^5+\alpha x^6+x^7$	4	revBCH
10	63	31	G→A	A_2	$1+x^3+x^5+x^7$	4	nsBCH
11	63	6	C→G	A_1	$\alpha+\beta x^2+\alpha x^3+\beta x^5+\alpha x^6+x^7$	4	BCH
11	63	60	G→C	A_1	$1+\beta x+x^2+x^3+x^4+x^5+\beta x^6+x^7$	4	revBCH
12	63	48	T→C	A_3	$1+\beta x+x^2+x^3+x^4+x^5+\beta x^6+x^7$	4	revBCH
13	63	35	T→C	A_1	$\alpha+x+\beta x^2+x^4+\beta x^5+x^7$	4	BCH
14	63	47	A→T	A_3	$1+x+x^2+x^4+x^5+x^7$	4	nsBCH
15	63	27	A→C	A_2	$1+\alpha x+x^2+x^3+x^4+x^5+\alpha x^6+x^7$	4	revBCH
16	63	20	C→T	A_2	$1+x+x^2+x^4+x^5+x^7$	4	nsBCH
17	63	10	G→T	A_2	$1+x+x^2+x^4+x^5+x^7$	4	nsBCH
18	63	40	T→G	A_2	$1+\beta x+\beta x^2+x^5+\alpha x^6+x^9$	4	BCH
18	63	61	C→T	A_1	$\alpha+\beta x^2+x^4+x^7$	4	BCH
19	63	21	A→G	A_2	$\beta+\alpha x+\alpha x^2+\beta x^3+x^4+\alpha x^5+x^6+x^7$	4	BCH
19	63	16	C→G	A_3	$\alpha+\beta x^2+\alpha x^6+x^7$	4	BCH
20	63	57	G→T	A_1	$1+x+x^2+x^4+x^5+x^7$	4	nsBCH
20	63	12	C→A	A_1	$\beta+\beta x+x^2+x^5+\alpha x^6+x^7$	4	BCH
21	63	26	G→T	A_1	$\alpha+\alpha x^3+\beta x^5+x^7$	4	BCH
21	63	28	A→G	A_2	$1+\beta x+\beta x^2+\beta x^5+\beta x^6+x^7$	4	revBCH
21	63	47	T→A	A_3	$\beta+\beta x+x^2+x^5+\alpha x^6+x^7$	4	BCH
22	63	13	A→T	A_1	$1+x+\beta x^3+\alpha x^4+x^5+x^6+\alpha x^8+x^9$	4	BCH
23	63	0	C→G	A_3	$\alpha+\alpha x+\beta x^2+\alpha x^3+x^4+\beta x^5+\beta x^6+x^7$	4	BCH
23	63	15	C→A	A_1	$1+x^2+x^4+x^7$	4	nsBCH
24	63	0	C→G	A_2	$1+x+x^5+x^7$	4	nsBCH
25	63	42	G→C	A_3	$1+x+x^5+x^7$	4	nsBCH
26	63	49	C→A	A_1	$1+x+x^2+\beta x^3+\beta x^4+x^5+x^6+x^7$	4	revBCH
27	63	3	A→T	A_0	$\beta+\beta x+\alpha x^2+\beta x^3+x^4+\alpha x^5+\alpha x^6+x^7$	4	BCH
27	63	31	G→T	A_2	$\alpha+\beta x^2+\alpha x^6+x^7$	4	BCH
28	63	6	C→A	A_3	$\alpha+\beta x^2+x^4+x^7$	4	BCH
29	63	25	C→A	A_2	$1+\beta x+\beta x^2+\beta x^5+\beta x^6+x^7$	4	revBCH
30	63	43	C→G	A_2	$1+x+x^5+x^7$	4	nsBCH
30	63	22	G→T	A_0	$\beta+\alpha x^2+\beta x^3+\alpha x^5+\beta x^6+x^7$	4	BCH
31	63	0	G→C	A_1	$1+x+x^2+\beta x^3+\beta x^4+x^5+x^6+x^7$	4	revBCH
32	63	22	G→T	A_0	$1+x+\alpha x^2+\alpha x^3+x^4+\alpha x^6+x^7+\alpha x^8+x^9$	5	BCH
32	63	18	A→G	A_3	$\beta+x+\alpha x^5+x^7$	4	BCH
33	63	1	T→A	A_2	$\beta+\alpha x+\beta x^2+\beta x^5+x^6+x^7$	4	BCH
33	63	3	T→G	A_0	$\alpha+\beta x+\beta x^2+\alpha x^3+x^4+\beta x^5+x^6+x^7$	4	BCH
33	63	6	T→G	A_0	$1+x+x^5+x^7$	4	nsBCH
34	63	24	G→T	A_0	$1+\beta x+\beta x^2+\beta x^5+\beta x^6+x^7$	4	revBCH
35	63	25	A→C	A_0	$\beta+\alpha x+\beta x^2+\beta x^5+x^6+x^7$	4	BCH
36	63	41	A→G	A_0	$\beta+x+\alpha x^2+x^4+\alpha x^5+x^7$	4	BCH
37	63	7	T→C	A_3	$\alpha+\beta x^2+\alpha x^6+x^7$	4	BCH

Continua na seguinte página

Tabela 20 – *Continuação da página anterior*

Index	**N**	**Pos**	**Mut**	**Rotulo**	$g(x)$	δ	**Tipo**
38	63	53	A→C	A_2	$1+\alpha x+\alpha x^2+\alpha x^5+\alpha x^6+x^7$	4	revBCH
39	63	10	A→G	A_2	$1+x+x^2+x^4+x^5+x^7$	4	nsBCH
39	63	53	T→C	A_2	$1+\alpha x+\alpha x^2+x^3+\beta x^6+\beta x^7+\beta x^8+x^9$	4	BCH
40	63	55	A→G	A_2	$1+\alpha x+x^2+x^3+x^4+x^5+\alpha x^6+x^7$	4	revBCH
41	63	44	T→C	A_2	$1+x^3+x^5+x^7$	4	nsBCH
42	63	29	T→C	A_2	$1+\beta x+\beta x^2+\beta x^5+\beta x^6+x^7$	4	revBCH
43	63	0	G→G	A_1	$1+\alpha x^2+\alpha x^3+x^4+\alpha x^7+x^8+x^9$	4	BCH
43	63	18	T→C	A_3	$1+x+x^2+\beta x^3+\beta x^4+x^5+x^6+x^7$	4	revBCH
44	63	9	T→G	A_3	$1+x^2+x^6+x^7$	4	nsBCH
45	63	0	A→C	A_0	$1+\alpha x+x^2+\beta x^3+x^5+x^6+\beta x^7+\alpha x^8+x^9+x^{10}$	6	nsBCH
45	63	20	T→C	A_3	$1+\alpha x+x^3+x^4+\alpha x^5+\beta x^6+x^8+x^9$	4	BCH
46	63	46	C→T	A_2	$1+\beta x^3+x^4+\alpha x^7+\alpha x^8+x^9$	4	BCH
47	63	40	A→C	A_1	$1+x^2+x^3+x^5+x^6+x^7$	4	nsBCH
47	63	59	T→A	A_3	$1+x^3+x^5+x^7$	4	nsBCH
48	63	43	G→A	A_0	$1+x+x^2+x^4+x^5+x^7$	4	nsBCH
49	63	43	A→G	A_2	$1+x^2+x^6+x^7$	4	nsBCH
50	63	25	C→T	A_1	$1+x^2+x^3+x^5+x^6+x^7$	4	nsBCH
51	63	57	A→T	A_1	$1+x^2+x^3+x^5+x^6+x^7$	4	nsBCH
52	63	34	A→G	A_0	$1+x^3+x^5+x^7$	4	nsBCH
53	63	46	C→G	A_3	$1+x^2+x^4+x^7$	4	nsBCH
53	63	44	C→G	A_3	$\beta+x+\alpha x^2+\alpha x^3+\beta x^4+\alpha x^5+\beta x^6+x^7$	4	BCH
54	63	3	G→T	A_0	$1+\alpha x+x^2+x^3+x^4+x^5+\alpha x^6+x^7$	4	revBCH
55	63	50	C→T	A_2	$1+x+x^2+x^4+x^5+x^7$	4	nsBCH
56	63	61	T→A	A_0	$\beta+x+\alpha x^5+x^7$	4	BCH
57	63	49	T→C	A_1	$\alpha+\beta x^2+\alpha x^3+\beta x^5+\alpha x^6+x^7$	4	BCH
58	93	29	G→A	A_0	$1+x^3+x^5+x^6+x^7+x^8+x^9+x^{11}$	4	nsBCH
59	1023	1006	T→G	A_2	$\alpha+\beta x+x^2+\beta x^7+x^8+x^{11}$	4	BCH

5 Códigos Cíclicos sobre Anéis Finitos, Comutativos e com Unidade

Como será visto no Capítulo 6, o projeto de códigos corretores de erros sobre $\mathbb{Z}_{20}$ e $\mathbb{F}_4 \times \mathbb{Z}_5$ é uma tarefa primordial para a modelagem do processo da síntese de proteínas como um único codificador genético concatenado e para evidenciar que a estrutura organizacional das proteínas é similar a estrutura de um código corretor de erros, isto é, algumas proteínas são identificadas como palavras-código de um código corretor de erros.

O procedimento padrão para o projeto de códigos corretores de erros sobre anéis comutativos com identidade, A, como é o caso específico dos anéis $\mathbb{Z}_{20}$ e $\mathbb{F}_4 \times \mathbb{Z}_5$, está baseado na descomposição do alfabeto em um produto direto de anéis locais. Assim, de maneira resumida, um código sobre A é construído pela justaposição de códigos corretores de erros, com igual comprimento, sobre esses anéis locais (BLAKE, 1972; SPIEGEL, 1978; DOUGHERTY *et al.*, 1999; DOUGHERTY; SHIROMOTO, 2000; GUENDA; GULLIVER, 2012). Como em (FARIA *et al.*, 2012; ROCHA *et al.*, 2010; FARIA *et al.*, 2010; BRANDÃO *et al.*, 2015), somente códigos BCH têm sido utilizados para identificar sequências DNA e mRNA e estes códigos BCH são bem restritos ao comprimento, assim a identificação de proteínas como palavras-código de códigos cíclicos sobre $\mathbb{Z}_{20}$ e $\mathbb{F}_4 \times \mathbb{Z}_5$, formados a partir de códigos BCH, somente é possível para um conjunto restrito de comprimentos, isto é, um número limitado de proteínas podem ser consideradas.

Neste capítulo, foi proposto um procedimento para o projeto de códigos corretores de erros lineares e cíclicos sobre anéis que generaliza a metologia padrão de projeto. Esta metologia surge da necessidade de flexibilizar a restrição de comprimento para assim poder analisar um número maior de proteínas, dado que as proteínas podem ter comprimentos muito variados (ZHANG, 2000; GEER *et al.*, 2010). O procedimento que é introduzido neste capítulo flexibiliza a restrição do comprimento e satisfaz as seguintes propriedades: construção simples, procedimentos de codificação e decodificação simples e estrutura matemática bem definida.

Antes de continuar, estabelece-se a notação requerida para o entendimento deste capítulo. A denota um anel finito comutativo, $\mathcal{C}$ um código de bloco corretor de erros sobre o alfabeto A^n, onde $\mathcal{C} \subset A^n$ e n é o comprimento do código. Se $\mathcal{C}$ é um submódulo de A^n, então diz-se que $\mathcal{C}$ é *linear*. Em particular, se $\mathcal{C}$ é um submódulo livre de A^n, então diz-se que $\mathcal{C}$ é *livre*.

Este capítulo está organizado da seguinte forma. Na Seção 5.1, revisa-se de maneira sucinta a metodologia tradicional para a construção de códigos produto sobre A

(CRT), segundo as propostas introduzidas em (BLAKE, 1972; DOUGHERTY *et al.*, 1999; GUENDA; GULLIVER, 2012), e indica-se as principais propriedades destes códigos. Na Seção 5.2, define-se o *código produto estendido* (ECRT), deduzem-se os parâmetros do código (comprimento, cardinalidade e distância mínima de Hamming), estabelecem-se as condições para que este código seja linear e/ou submódulo livre e obtém-se um conjunto gerador. Na Seção 5.3, verificam-se as condições para que o código ECRT seja cíclico e encontra-se um conjunto de polinômios que o geram como ideal. Na Seção 5.4, descrevem-se dois algoritmos: o primeiro para verificar quando uma sequência em A^n pertence ao código ECRT e o segundo para corrigir erros randômicos e do tipo "*Burst*" cíclicos. Finalmente, na Seção 5.5, realizam-se os comentários finais do capítulo.

5.1 Construção Tradicional de Códigos sobre Anéis Finitos

O procedimento proposto em (BLAKE, 1972) para a construção de códigos de bloco sobre $\mathbb{Z}_m$ foi estudada e estendida em diferentes trabalhos desde 1972 (SPIEGEL, 1978; DOUGHERTY *et al.*, 1999; DOUGHERTY; SHIROMOTO, 2000; GUENDA; GULLIVER, 2012) para a construção de códigos de bloco sobre A. A metodologia introduzida em tais trabalhos utiliza o Teorema Chinês do Resto e o isomorfismo entre A e a soma direta de anéis locais, isto é, $A \cong A_1 \oplus \cdots \oplus A_s$ (Teorema 3.14 in (BINI *et al.*, 2002)), onde A_i denota um anel local que pode ser $\mathbb{Z}_{p_i{}^{r_i}}$ ou um corpo, entre outros. A metodologia de construção consiste em: 1) construir códigos de bloco lineares, $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_s$, de comprimento n sobre $A_1, A_2, \ldots, A_s$, respectivamente; e 2) usar o Teorema Chinês do Resto (CRT) para obter um código de bloco linear $\mathcal{C}$ dado por:

$$\begin{aligned} \mathcal{C} :=& \mathrm{CRT}(\mathcal{C}_1, \ldots, \mathcal{C}_s) \\ :=& \{\Psi^{-1}(v_1, \ldots, v_s) \mid v_i \in \mathcal{C}_i\}, \end{aligned} \tag{5.1}$$

onde o mapa Ψ é definido como (GUENDA; GULLIVER, 2012):

$$\Psi : A^n \to \prod_{i=1}^{s} (R/\mathfrak{m}_i^{t_i})^n,$$

onde $\mathfrak{m}_i$ é o ideal maximal de A_i e t_i é o correspondente índice de estabilidade (GUENDA; GULLIVER, 2012). O mapa Ψ^{-1} usa o CRT e justapõe, componente-a-componente, todas as palavras-código dos códigos $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_s$. Diferentes metodologias para a construção de códigos de bloco lineares sobre anéis locais e/ou corpos podem ser encontradas nos trabalhos: (NORTON; SALAGEAN, 2000; DOUGHERTY *et al.*, 2011; SHANKAR, 1979; KANWAR; LÓPEZ-PERMOUTH, 1997), e referências citadas nesses trabalhos.

Note que o código $\mathcal{C}$ pode ser construído, passo a passo, pela justaposição de dois códigos, como se mostra a seguir:

$$\mathcal{C} = \mathrm{CRT}(\mathcal{C}_1,\ \mathrm{CRT}(\mathcal{C}_2,\ \mathrm{CRT}(\mathcal{C}_3,\ \ldots))). \tag{5.2}$$

Portanto, é suficiente encontrar as propriedades da combinação de dois códigos para deduzir as propriedades de $\mathcal{C}$.

Seja $\mathcal{C}_{12}$ a justaposição de $\mathcal{C}_1$ e $\mathcal{C}_2$, isto é, $\mathcal{C}_{12} = \mathrm{CRT}(\mathcal{C}_1, \mathcal{C}_2)$ sobre $A_{12} \cong A_1 \oplus A_2$. $\mathcal{C}_1$ e $\mathcal{C}_2$ são códigos sobre os anéis locais A_1 e A_2, respectivamente; e assim $\mathcal{C}_{12}$ pode ser equivalentemente expresso através do seguinte operador binário:

Definição 5.1. *Sejam c_i e c_j dados por $c_i = (c_{i_0}, c_{i_1}, \ldots, c_{i_{n-1}})$ e $c_j = (c_{j_0}, c_{j_1}, \ldots, c_{j_{n-1}})$, onde $c_{i_k} \in A_1$ e $c_{j_k} \in A_2$, respectivamente, com $0 \leq k \leq n-1$. Define-se,*

$$c_i \bullet c_j = \Big((c_{i_0}, c_{j_0}), \ldots, (c_{i_{n-1}}, c_{j_{n-1}})\Big) \in (A_1 \times A_2)^n.$$

Equivalentemente, $\mathcal{C}_{12}$ pode também ser definido como:

$$\mathcal{C}_{12} := \{c_i \bullet c_j \mid \forall c_i \in \mathcal{C}_1,\ \forall c_j \in \mathcal{C}_2\}. \tag{5.3}$$

Segundo (BLAKE, 1972) e (GUENDA; GULLIVER, 2012), se $\mathcal{C}_1$ e $\mathcal{C}_2$ são ambos códigos lineares e cíclicos com parâmetros (n, k_1, d_1) e (n, k_2, d_2), respectivamente, então o código de bloco resultante, $\mathcal{C}_{12}$, de comprimento n e cardinalidade $|C_1| \cdot |C_2|$, é linear e cíclico e tem distância mínima de Hamming: $d = \min\{d_1, d_2\}$.

Em (BLAKE, 1972; SPIEGEL, 1978; DOUGHERTY *et al.*, 1999; DOUGHERTY; SHIROMOTO, 2000; GUENDA; GULLIVER, 2012), um conjunto mínimo de geradores para o submódulo C_{12} é dado e a propriedade do **posto** é usada para mostrar que: $\mathrm{Rank}(\mathcal{C}_{12}) = \max\{\mathrm{Rank}(\mathcal{C}_1), \mathrm{Rank}(\mathcal{C}_2)\}$ é MDR (*distância máxima com respeito ao posto*) (GUENDA; GULLIVER, 2012)), se $\mathcal{C}_1$ e $\mathcal{C}_2$ são ambos MDR. Além disso, os códigos CRT são generalizados para polinômios, os quais permitem a caracterização quando $\mathcal{C}_{12}$ é cíclico. Nesta direção, um conjunto de geradores para o ideal $\mathcal{C}_{12}$ é dado e a generalização para a construção de códigos BCH e Reed-Solomon sobre $A = \mathbb{Z}_m$ é introduzida. Finalmente, note que as propriedades anteriores são facilmente generalizadas para o código C, dada a Equação 5.2.

5.2 Códigos Lineares Produto Estendido

A construção dos códigos CRT sobre A de comprimento n necessita que todos os códigos $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_s$ tenham o mesmo comprimento n. Logo, a limitação desta metodologia está relacionada ao fato que, para alguns valores de n, estes códigos não são conhecidos. Para superar esta limitação, neste capítulo, se propõe um procedimento para a construção

de códigos sobre $A_1 \times A_2$ com comprimento $n_{12} = \text{lcm}(n_1, n_2)$, onde *lcm* representa a operação do *mínimo múltiplo comum*, pela justaposição de dois códigos diferentes sobre A_1 e A_2 de comprimento n_1 e n_2 (não necessariamente iguais), isto é, $C_{12} := ECRT(C_1, C_2)$ é projetado sobre $A_1 \times A_2$ para um conjunto maior de comprimentos ao usar códigos $\mathcal{C}_1$ e $\mathcal{C}_2$ sobre A_1 e A_2 conhecidos. Usa-se um procedimento similar ao mostrado na Equação 5.2 para a construção de códigos sobre A; como indica a Equação 5.4.

$$\mathcal{C} := \text{ECRT}(\mathcal{C}_1,\ \text{ECRT}(\mathcal{C}_2,\ \text{ECRT}(\mathcal{C}_3,\ \ldots))). \tag{5.4}$$

Sejam $\mathcal{C}_1$ e $\mathcal{C}_2$ códigos de bloco lineares sobre A_1 e A_2, anéis finitos, locais e comutativos. Os correspondentes comprimentos e distâncias mínimas de $\mathcal{C}_1$ e $\mathcal{C}_2$ são: n_1 e d_1 e n_2 e d_2. Logo, existem números inteiros: n, g, α e β, tais que as seguintes relações são satisfeitas:

$$\begin{aligned} n =& \text{lcm}(n_1, n_2) \\ g =& \gcd(n_1, n_2) \\ n =& \alpha n_1 = \beta n_2 = (n_1 \cdot n_2)/g, \end{aligned} \tag{5.5}$$

onde o operador $\gcd(n_1, n_2)$ representa o máximo divisor comum entre n_1 e n_2.

De maneira similar à definição do operador $\bullet$ (veja a Definição 5.1), o operador $\star$ para a combinação de duas sequências pode ser definido como na Definição 5.2.

Definição 5.2. *Sejam* $c_r = (c_{r_0}, c_{r_1}, \ldots, c_{r_{n_1-1}}) \in {A_1}^{n_1}$ *e* $c_s = (c_{s_0}, c_{s_1} \ldots, c_{s_{n_2-1}}) \in {A_2}^{n_2}$, *onde* A_1 *e* A_2 *são anéis locais, finitos e comutativos. O operador binário* $\star$ *é definido como:*

$$\begin{aligned} \star\ :\ {A_1}^{n_1} \times {A_2}^{n_2} &\to (A_1 \times A_2)^n \\ (c_r, c_s) &\mapsto c_r \star c_s := (p_0, \ldots, p_l, \ldots, p_{n-1}), \end{aligned}$$

onde $p_l = (c_r, c_s) \in A_1 \times A_2$, *com* $r = (l)_{n_1}$ *e* $s = (l)_{n_2}$, *e* $(x)_y$ *denota* x *módulo* y.

Exemplo 5.1. *Dado* $c_r = (1, 0, 1, 1, 0, 1) \in (\mathbb{Z}_2)^6$ *e* $c_s = (2, 1, 0, 1) \in (\mathbb{Z}_3)^4$, *segue que* $n = 2 \cdot 6 = 3 \cdot 4 = 12$ *e:*

$$\begin{aligned} c_r \star c_s =& (1, 0, 1, 1, 0, 1) \star (2, 1, 0, 1) \\ c_r \star c_s =& ([1, 2], [0, 1], [1, 0], [1, 1], [0, 2], [1, 1], \\ & [1, 0], [0, 1], [1, 2], [1, 1], [0, 0], [1, 1]) \\ c_r \star c_s =& (5, 4, 3, 1, 2, 1, 3, 4, 5, 1, 0, 1), \end{aligned}$$

onde se utiliza o isomorfismo de anéis entre $\mathbb{Z}_2 \times \mathbb{Z}_3$ *e* $\mathbb{Z}_6$. *A seguir detalha-se tal isomorfismo:*

$$\begin{aligned} &[0, 0] \to 0, \quad [1, 1] \to 1, \quad [0, 2] \to 2, \\ &[1, 0] \to 3, \quad [0, 1] \to 4, \quad [1, 2] \to 5. \end{aligned}$$

Segundo a Definição 5.2, seja $l = m(i,j)$, o qual denota o índice do vetor $c_r \star c_s$, onde i e j se referem aos índices dos vetores c_r e c_s, respectivamente. Assim, a seguinte relação pode ser estabelecida:

$$i = (l)_{n_1} = (m(i,j))_{n_1} \tag{5.6}$$

$$j = (l)_{n_2} = (m(i,j))_{n_2}. \tag{5.7}$$

Note que i e j podem ser obtidos a partir de l ao aplicar a operação módulo n_1 e módulo n_2 a l, respectivamente. Em algumas aplicações pode ser relevante encontrar o valor de l a partir de i e j; para isso, as Equações 5.6 e 5.7 devem ser resolvidas. A partir da Equação 5.6, obtém-se $l = q_1 n_1 + i$, para algum inteiro q_1, e ao substitui-la na Equação 5.7, chega-se em:

$$q_1 n_1 + i \equiv j \pmod{n_2} \tag{5.8}$$

$$q_1 n_1 \equiv j - i \pmod{n_2}$$

$$q_1 \left(\frac{n_1}{g}\right) \equiv \frac{j-i}{g} \pmod{n_2/g}. \tag{5.9}$$

Dado que $\gcd(n_1/g, n_2/g) = 1$, sabe-se que existe $t, u \in \mathbb{Z}$ tais que $t(n_1/g) + u(n_2/g) = 1$. Nesta direção $t(n_1/g) \equiv 1 \pmod{n_2/g}$, ao substitui-la nas Equações 5.9 e 5.6, obtém-se, para algum inteiro q_2:

$$q_1 = t\,\frac{j-i}{g} + \; q_2 \frac{n_2}{g}$$

$$l = \left(t\frac{(j-i)}{g} + q_2\frac{n_2}{g}\right) n_1 + i$$

$$m(i,j) \equiv t n_1 \frac{(j-i)}{g} + i \pmod{n}. \tag{5.10}$$

A Equação 5.10 é válida somente para i e j que satisfazem as Equações 5.6 e 5.7 para algum $l \in \{0, \ldots, n-1\}$.

Observe que quando $n_1 = n_2$, obtém-se: $n = n_1$, $g = n$ e $l = m(i,j) = i = j$. Este é o caso especial quando o operador binário $\bullet$, na Definição 5.1, é utilizado ao invés do operador $\star$.

Quando n_1 é coprimo com n_2, obtém-se: $n = n_1 \cdot n_2$ e $g = 1$. Este caso resulta na mesma definição introduzida em (BURTON; WELDON, 1965). Originalmente, este trabalho se baseou nesta definição.

Definição 5.3. *Seja $\mathcal{C}$ o **Código Produto Estendido** sobre $A_1 \times A_2$, definido da seguinte maneira:*

$$\mathcal{C} := ECRT(\mathcal{C}_1, \mathcal{C}_2) = \{c_i \star c_j | \forall c_i \in \mathcal{C}_1, \forall c_j \in \mathcal{C}_2\}. \tag{5.11}$$

O próximo teorema estabelece as condições nas quais o *código produto estendido* é linear; isto é, as condições para as quais $\mathcal{C}$ é um submódulo de $(A_1 \times A_2)^n$.

Teorema 5.1. *$\mathcal{C}$ é um código de bloco linear (um submódulo) sobre $A_1 \times A_2$, se $\mathcal{C}_1$ e $\mathcal{C}_2$ são códigos de bloco lineares sobre A_1 e A_2, respectivamente.*

Demonstração: Sejam $\gamma = (\gamma_1, \gamma_2) \in A_1 \times A_2$ e $v, w \in C$. Assim, $\exists v_1, w_1 \in \mathcal{C}_1$ e $\exists v_2, w_2 \in \mathcal{C}_2$ tais que: $v = v_1 \star v_2$ e $w = w_1 \star w_2$. Portanto, os seguintes fatos devem ser provados:

- $v + w = (v_1 + w_1) \star (v_2 + w_2)$.
- $\gamma v = (\gamma_1 v_1 \star \gamma_2 v_2)$.

Dado que $\mathcal{C}_1$ e $\mathcal{C}_2$ são A_1-módulo e A_2-módulo, respectivamente, segue que $v + w \in \mathcal{C}$ e $\gamma v \in \mathcal{C}$. Consequentemente, $\mathcal{C}$ é um $A_1 \times A_2$-modulo. A seguir, provam-se os fatos acima mencionados:
Para valores arbitrários de $v = (p_0, \ldots, p_{n-1})$ e $w = (q_0, \ldots, q_{n-1})$, onde:

- $v_1 = (a_0, \ldots, a_{n_1-1}) \in \mathcal{C}_1$
- $w_1 = (c_0, \ldots, c_{n_1-1}) \in \mathcal{C}_1$
- $v_2 = (b_0, \ldots, b_{n_2-1}) \in \mathcal{C}_2$
- $w_2 = (d_0, \ldots, d_{n_2-1}) \in \mathcal{C}_2$.

segue que

$$\begin{aligned} v + w =& ((p_0 + q_0), (p_1 + q_1), \ldots, (p_{n-1} + q_{n-1})) \\ =& (\ldots, ((a_{(l)_{n_1}}, b_{(l)_{n_2}}) + (c_{(l)_{n_1}}, d_{(l)_{n_2}})), \ldots) \\ =& (\ldots, (a_{(l)_{n_1}} + c_{(l)_{n_1}}, b_{(l)_{n_2}} + d_{(l)_{n_2}}), \ldots) \\ =& (v_1 + w_1) \star (v_2 + w_2) \end{aligned}$$

e

$$\begin{aligned} \gamma v =& (\gamma p_0, \gamma p_1, \ldots, \gamma p_{n-1}) \\ =& (\ldots, (\gamma_1, \gamma_2)(a_{(l)_{n_1}}, b_{(l)_{n_2}}), \ldots) \\ =& (\ldots, (\gamma_1 a_{(l)_{n_1}}, \gamma_2 b_{(l)_{n_2}}), \ldots) \\ =& (\gamma_1 v_1) \star (\gamma_2 v_2). \end{aligned}$$

■

Para obter os parâmetros do código, precisa-se provar que o operador binário $\star$ é um mapa injetivo.

Lema 5.1. *O operador binário:*

$$\star \ : \ {A_1}^{n_1} \times {A_2}^{n_2} \to (A_1 \times A_2)^n$$

é injetivo.

Demonstração: Considere $v = (p_0, \ldots, p_{n-1}) = v_1 \star v_2$ e $w = (q_0, \ldots, q_{n-1}) = w_1 \star w_2$, onde $v_1 = (a_0, \ldots, a_{n_1-1})$, $w_1 = (c_0, \ldots, c_{n_1-1})$, $v_2 = (b_0, \ldots, b_{n_2-1})$, e $w_2 = (d_0, \ldots, d_{n_2-1})$. Se $v = w$, então $p_l = (a_{(l)_{n_1}}, b_{(l)_{n_2}}) = (c_{(l)_{n_1}}, d_{(l)_{n_2}}) = q_l$, para todo $l \in \{0, \ldots, n-1\}$. Em particular, se $a_i = c_i$, para $i \in \{0, \ldots, n_1 - 1\}$, e $b_j = d_j$, para $j \in \{0, \ldots, n_2 - 1\}$, então $v_1 = w_1$ e $v_2 = w_2$. ∎

O Lema 5.1 é muito importante e estabelece que a cardinalidade do novo código $\mathcal{C}$ é dada por: $|C| = |C_1|.|C_2|$. O seguinte passo é encontrar a distância mínima do *código produto estendido.*

Lema 5.2. *Sejam $d(C_1) = d_1$ e $d(C_2) = d_2$ as distâncias mínimas de Hamming dos códigos $\mathcal{C}_1$ e $\mathcal{C}_2$, respectivamente. A distância mínima de $\mathcal{C}$ é dada por:*

$$d(C) = \min\{\alpha d_1, \beta d_2\}, \tag{5.12}$$

onde α e β são números inteiros positivos.

Demonstração: Considere $w_1 \in \mathcal{C}_1$ e $w_2 \in \mathcal{C}_2$ como palavras-código com peso de Hamming d_1 e d_2, respectivamente. Por inspeção, veja a Definição 5.2, observa-se que a operação $w_1 \star w_2$ pode ser realizada pelos seguintes passos: 1) repetir α vezes a palavra-código w_1 para formar um vetor em $(A_1)^n$; 2) repetir β vezes a palavra-código w_2 para formar um vetor em $(A_2)^n$; e 3) justapor componente-a-componente esses vetores para formar uma palavra-código de $\mathcal{C}$ em $(A_1 \times A_2)^n$. Assim, pela propriedade injectiva e ao notar que o *zero* em $A_1 \times A_2$ é $(0, 0)$, o qual é a justaposição do 0 em A_1 com o 0 em A_2, obtém-se:

$$\begin{aligned} \mathrm{d}(\mathcal{C}) &= \min\{\mathrm{peso}(w_1 \star \underline{0}), \mathrm{peso}(\underline{0} \star w_2)\} \\ \mathrm{d}(\mathcal{C}) &= \min\{\alpha d_1, \beta d_2\}. \end{aligned}$$

∎

Dado que $(A_1)^n$ e $(A_2)^n$ são módulos finitos, segue que $\mathcal{C}_1$ e $\mathcal{C}_2$ são submódulos finitamente gerados. O seguinte teorema estabelece um conjunto de geradores para o **código produto estendido** $\mathcal{C}$.

Lema 5.3. *Se $\mathcal{C}_1$ e $\mathcal{C}_2$ são submódulos finitamente gerados por $\Gamma_1 = \{x_1, \ldots, x_{s_1}\}$ e $\Gamma_2 = \{y_1, \ldots, y_{s_2}\}$ (assumindo $s_1 \geq s_2$ sem perda de generalidade), então $\mathcal{C}$ é um submódulo finitamente gerado por:*

$$\Psi = \{(x_1 \star y_1), \ldots, (x_{s_2} \star y_{s_2}), (x_{s_2+1} \star \underline{0}), \ldots, (x_{s_1} \star \underline{0})\},$$

onde $\underline{0}$ denota o vetor zero.

Demonstração: Seja $\mathcal{C}$ tal que

$$\mathcal{C} \subseteq \{\cdots + \gamma_i(x_i \star y_i) + \ldots \mid \gamma_i \in (A_1 \times A_2)\}.$$

Seja $v \in \mathcal{C}$, então existem: $v_1 \in C_1$ e $v_2 \in C_2$ tal que $v = v_1 \star v_2$. Dado que, $\mathcal{C}_1$ e $\mathcal{C}_2$ são submódulos finitamente gerados, segue que há combinações lineares para v_1 e v_2 tais que:

$$\begin{aligned} v =& (a_1x_1 + \cdots + a_{s_1}x_{s_1}) \star (b_1y_1 + \cdots + b_{s_2}y_{s_2}) \\ =& \cdots + (a_{s_2}x_{s_2} \star b_{s_2}y_{s_2}) + \cdots + (a_{s_1}x_{s_1} \star \underline{0}) \\ =& \cdots + (a_{s_2}, b_{s_2})(x_{s_2} \star y_{s_2}) + \cdots + (a_{s_1}, b_{s_1})(x_{s_1} \star \underline{0}) \\ =& \cdots + \gamma_{s_2}(x_{s_2} \star y_{s_2}) + \cdots + \gamma_{s_1}(x_{s_1} \star \underline{0}). \end{aligned}$$

onde $b_{s_2+1}, \ldots, b_{s_1}$ é algum elemento em A_2 e $(a_l, b_l) = \gamma_l \in (A_1 \times A_2)$.

Note que a outra direção da demonstração consiste em provar que:

$$\{\cdots + \gamma_i(x_i \star y_i) + \ldots \mid \gamma_i \in (A_1 \times A_2)\} \subseteq C,$$

o qual fica evidente ao fazer a leitura da prova anterior na outra direção. ∎

O Lema 5.3 mostra que Ψ é um conjunto gerador de $\mathcal{C}$. Os seguintes lemas proveem as condições para que Ψ seja *mínimo* e para que $\mathcal{C}$ seja um submódulo finito livre de **posto**: $\text{Rank}(C) = |\Psi|$. Diz-se que Ψ é um *conjunto gerador mínimo* se não há outro conjunto gerador de $\mathcal{C}$ (Υ) tal que $|\Psi| > |\Upsilon|$.

Lema 5.4. *Usando a mesma notação usada no Lema 5.3, se Γ_1 e Γ_2 são conjuntos geradores mínimos de $\mathcal{C}_1$ e $\mathcal{C}_2$, respectivamente, então Ψ é um conjunto gerador mínimo de $\mathcal{C}$.*

Demonstração: A prova é por contradição. Se Ψ não é minimal, então existe um conjunto gerador $\Upsilon = \{z_1, \ldots, z_t\}$ para $\mathcal{C}$ tal que $|\Upsilon| < |\Psi|$ $(t < s_1)$. Dado que todo z_j pertence a $\mathcal{C}$, segue que existe $p_j \in C_1$ e $q_j \in C_2$ tais que $z_j = p_j \star q_j$. Como Υ é um conjunto gerador de $\mathcal{C}$, segue que todo elemento em Ψ pode ser expresso como uma combinação linear de elementos em Υ, isto é, para $1 \leq i \leq s_1$, tem-se:

$$\begin{aligned} x_i \star y_i =& (a_{i1}, b_{i1})z_1 + \cdots + (a_{it}, b_{it})z_t \\ x_i \star y_i =& (a_{i1}p_1 \star b_{i1}q_1) + \cdots + (a_{it}p_t \star b_{it}q_t) \\ x_i \star y_i =& (a_{i1}p_1 + \cdots + a_{it}p_t) \star (b_{i1}q_1 + \cdots + b_{it}q_t). \end{aligned} \tag{5.13}$$

Considerando a Equação 5.13 e o fato que o operador $\star$ é injetivo, obtém-se que x_i pode ser representado como:

$$x_i = a_{i1}p_1 + \cdots + a_{it}p_t.$$

Portanto, o conjunto $\{p_1, \ldots, p_t\}$ é um conjunto gerador para $\mathcal{C}_1$ menor que Γ_1, o qual é uma contradição. ∎

A partir da prova do Lema 5.3, verifica-se o fato de $\mathcal{C}_1$ e $\mathcal{C}_2$ serem submódulo livres, não garante que $\mathcal{C}$ seja um submódulo livre. Porém, pode-se demonstrar o seguinte resultado.

Lema 5.5. *Se $\mathcal{C}_1$ e $\mathcal{C}_2$ são submódulos livres tais que Rank(C_1) = Rank$(C_2) = s$, então $\mathcal{C}$ é um submódulo livre em $(A_1 \times A_2)^n$ com Rank$(C) = s$.*

Demonstração: Sejam C_1 e C_2 submódulos livres e $\Gamma_1 = \{x_1, \ldots, x_s\}$ e $\Gamma_2 = \{y_1, \ldots, y_s\}$ suas bases correspondentes. Sabe-se que $\Psi = \{(x_1 \star y_1), \ldots, (x_s \star y_s)\}$ é um conjunto minimal de geradores de $\mathcal{C}$ pelos Lemas 5.3 e 5.4. Assim, para provar que Ψ é uma base, precisa-se mostrar que Ψ é um conjunto linearmente independente, isto é,

$$\begin{aligned} \gamma_1(x_1 \star y_1) + \cdots + \gamma_s(x_s \star y_s) &= 0 \\ (a_1, b_1)(x_1 \star y_1) + \cdots + (a_s, b_s)(x_s \star y_s) &= 0 \\ (a_1x_1 + \cdots + a_sx_s) \star (b_1y_1 + \cdots + b_sy_s) &= 0. \end{aligned}$$

Dado que o operador "$\star$" é injetivo, segue que:

$$\begin{aligned} a_1x_1 + \cdots + a_sx_s &= 0 \\ b_1y_1 + \cdots + b_sy_s &= 0. \end{aligned}$$

Assim, a única solução é: $a_l = b_l = 0$ e $\gamma_l = (0, 0)$, para $l \in \{1, \ldots, s\}$, dado que Γ_1 e Γ_2 são bases. Portanto, provou-se que: Ψ é uma base do submódulo livre $\mathcal{C}$, Rank$(\mathcal{C}) = s$ e $C \simeq (A_1 \times A_2)^s$. ∎

Como esperado, o Lema 5.4 implica o Lema 2.1 em (DOUGHERTY; SHIROMOTO, 2000), devido que o operador binário "$\bullet$" é um caso especial do operador "$\star$" quando os comprimentos de $\mathcal{C}_1$ e $\mathcal{C}_2$ são iguais.

Exemplo 5.2. *Sejam $\mathcal{C}_1$ e $\mathcal{C}_2$ códigos lineares sobre $\mathbb{Z}_3$ e $\mathbb{Z}_4$, com bases $\Gamma_1 = \{(1, 1)\}$ e $\Gamma_2 = \{(1, 0, 1, 0), (0, 2, 0, 2)\}$, respectivamente:*

- *$A_1 = \mathbb{Z}_3$, $\mathcal{C}_1 \subset (\mathbb{Z}_3)^2$ e $|\mathcal{C}_1| = 3$;*
- *$A_2 = \mathbb{Z}_4$, $\mathcal{C}_2 \subset (\mathbb{Z}_4)^4$ e $|\mathcal{C}_2| = 8$.*

*O **código produto estendido** $\mathcal{C}$ sobre $\mathbb{Z}_3 \times \mathbb{Z}_4 \simeq \mathbb{Z}_{12}$ tem comprimento $n = 4$ ($\alpha = 2$ e $\beta = 1$), 24 palavras-código ($|\mathcal{C}| = 24$) e distância mínima 2 ($d(C) = 2$).*

Segundo o Lema 5.3, segue que

$$\begin{aligned}
\Psi =& \{(1,1)\star(1,0,1,0),(0,0)\star(0,2,0,2)\}\\
\Psi =& \{([1,1],[1,0],[1,1],[1,0]),([0,0],[0,2],[0,0],[0,2])\}\\
\Psi =& \left\{\left(\begin{bmatrix}1\\1\end{bmatrix},\begin{bmatrix}1\\0\end{bmatrix},\begin{bmatrix}1\\1\end{bmatrix},\begin{bmatrix}1\\0\end{bmatrix}\right),\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}0\\2\end{bmatrix},\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}0\\2\end{bmatrix}\right)\right\}\\
\Psi =& \{(1,4,1,4),(0,6,0,6)\},
\end{aligned}$$

onde o isomorfismo entre $\mathbb{Z}_3 \times \mathbb{Z}_4$ e $\mathbb{Z}_{12}$ é utilizado.

5.3 Códigos Cíclicos Produto Estendido

Nesta seção, prova-se que se $\mathcal{C}_1$ e $\mathcal{C}_2$ são cíclicos, então o *código produto estendido* $\mathcal{C}$, obtido pelo uso de: $C = ECRT(C_1, C_2)$, é cíclico. Além disso, representa-se $\mathcal{C}$ como um ideal em $(A_1 \times A_2)[x]/\langle x^n - 1\rangle$ pelo uso da representação de $\mathcal{C}_1$ e $\mathcal{C}_2$ como ideais em $A_1[x]/\langle x^{n_1} - 1\rangle$ e $A_2[x]/\langle x^{n_2} - 1\rangle$, respectivamente. Primeiro, estabelecem-se as condições para que $\mathcal{C}$ seja cíclico, logo um operador equivalente $\star$ para a justaposição de dois polinômios em $A_1[x]/\langle x^{n_1} - 1\rangle$ e $A_2[x]/\langle x^{n_2} - 1\rangle$ é definido, e finalmente, quando $\mathcal{C}$ é cíclico, obtém-se um conjunto de geradores do ideal $\mathcal{C}$ pela justaposição dos geradores dos ideais $\mathcal{C}_1$ e $\mathcal{C}_2$.

Teorema 5.2. *Se $\mathcal{C}_1$ e $\mathcal{C}_2$ são códigos de bloco cíclicos sobre A_1 e A_2, respectivamente, então $\mathcal{C}$ é um código de bloco cíclico sobre $A_1 \times A_2$.*

Demonstração: De maneira similar à prova do *Teorema I* em (BURTON; WELDON, 1965), é suficiente mostrar que o deslocamento cíclico de uma palavra-código é igual à justaposição de palavras-código deslocadas, pelo uso do operador "$\star$", isto é,

$$\begin{aligned}
m(i+1, j+1) &\equiv rn_1\frac{(j-i)}{g} + i + 1 \pmod{n}\\
&\equiv m(i,j) + 1 \pmod{n}.
\end{aligned}$$

■

Para definir o operador $\star$ para polinômios se utiliza a bem conhecida bijeção, denotada por T, entre $A[x]/\langle x^n - 1\rangle$ e A^n:

$$\begin{aligned}
T : a_0 + a_1x + \cdots + a_{n-1}x^{n-1} &\mapsto (a_0, \ldots, a_{n-1})\\
g(x) \star f(x) &:= T^{-1}\{T(g(x)) \star T(f(x))\}.
\end{aligned}$$

Observa-se que a linearidade obtida a partir do Teorema 5.1 pode ser facilmente estendida para polinômios. Assim, dado:

$$g(x) = (g_1(x) \star g_2(x)) \quad \text{e} \quad f(x) = (f_1(x) \star f_2(x)),$$

e

$$\gamma = (\gamma_1, \gamma_2) \in A_1 \times A_2,$$

para todo $g_1(x), f_1(x) \in A_1[x]/\langle x^{n_1} - 1\rangle$ e $g_2(x), f_2(x) \in A_2[x]/\langle x^{n_2} - 1\rangle$, obtém.se:

$$g(x) + f(x) = ((g_1(x) + f_1(x)) \star (g_2(x) + f_2(x))) \tag{5.14}$$

$$\gamma g(x) = (\gamma_1 g_1(x)) \star (\gamma_2 g_2(x)). \tag{5.15}$$

Usando estes fatos, representam-se os ideais de $\mathcal{C}$ pela combinação dos ideais $\mathcal{C}_1$ e $\mathcal{C}_2$.

Lema 5.6. *Sejam $\mathcal{C}_1$ e $\mathcal{C}_2$ ideais em $A_1[x]/\langle x^{n_1}-1\rangle$ e $A_2[x]/\langle x^{n_2}-1\rangle$, finitamente gerados por um conjunto de geradores $\Theta_1 = \{f_1, \ldots, f_s\}$ e $\Theta_2 = \{h_1, \ldots, h_t\}$, respectivamente. O* código produto estendido *$\mathcal{C}$ é um ideal em $(A_1 \times A_2)[x]/\langle x^n - 1\rangle$ finitamente gerado pelo conjunto: $\Theta = \{g_1, \ldots, g_t\}$ (para $t \geq s$), onde:*

$$g_i = \begin{cases} f_i \star h_i & 0 \leq i \leq s \\ 0 \star h_i & s+1 \leq i \leq t \end{cases} \tag{5.16}$$

Demonstração: Sabe-se que $\mathcal{C}_1 = \langle f_1 \ldots, f_s\rangle$ e $\mathcal{C}_2 = \langle h_1, \ldots, h_t\rangle$. Assim, deve-se provar que $\mathcal{C} = \langle g_1, \ldots, g_t\rangle$.

- $C \subseteq \langle g_1, \ldots, g_t\rangle$. Se $c \in \mathcal{C}$, então existe $c_1 = a_1 f_1 + \cdots + a_s f_s$ e $c_2 = b_1 h_1 + \cdots + b_t h_t$ tais que:

$$\begin{aligned} c =& c_1 \star c_2 \\ c =& (a_1 f_1 + \cdots + a_s f_s) \star (b_1 h_1 + \cdots + b_t h_t) \\ c =& (a_1 f_1 \star b_1 h_1) + \cdots + (a_i f_i \star b_i h_i) + \ldots \\ & (a_s f_s \star b_s h_s) + \cdots + (0 \star b_t h_t) \qquad (5.17) \\ c =& \sigma_1 g_1 + \cdots + \sigma_t g_t. \qquad (5.18) \end{aligned}$$

  onde a Equação 5.17 é obtida ao aplicar muitas vezes o fato expressado na Equação 5.14, e a Equação 5.18 é explicada a seguir; onde $a_i = a_{i0} + \cdots + a_{i(n_1-1)} x^{n_1-1}$, $b_i = b_{i0} + \cdots + b_{i(n_2-1)} x^{n_2-1}$ e $g^{(j)}$ denota o j-ésimo deslocamento cíclico no polinômio g:

$$\begin{aligned} a_i f_i \star b_i h_i =& (a_{i0} f_i + \cdots + a_{i(n_1-1)} {f_i}^{(n_1-1)}) \star \\ & (b_{i0} h_i + \cdots + b_{i(n_2-1)} {h_i}^{(n_2-1)}) \\ a_i f_i \star b_i h_i =& \cdots + (a_{ij} f_i^{(j)} \star b_{ij} h_i^{(j)}) + \ldots \qquad (5.19) \\ a_i f_i \star b_i h_i =& \cdots + (a_{ij}, b_{ij}) (f_i \star h_i)^{(j)} + \ldots \qquad (5.20) \\ a_i f_i \star b_i h_i =& \cdots + \sigma_{ij} {g_i}^{(j)} + \ldots \qquad (5.21) \\ a_i f_i \star b_i h_i =& \sigma_i g_i. \qquad (5.22) \end{aligned}$$

onde as Equações 5.19 e 5.20 são obtidas por aplicar repetidas vezes os fatos expressos nas Equações 5.14 e 5.15, respectivamente. Na Equação 5.21 se considera $\sigma_{ij} = (a_{ij}, b_{ij})$ e $a_{ij} = 0$ para $j > n_1 - 1$ ($n_1 \leq n_2$ sem perda de generalidade); e na Equação 5.22,

$$\sigma_i = \sigma_{i0} + \cdots + \sigma_{i(n_2-1)} x^{n_2-1}.$$

- $\langle g_1, \ldots, g_t \rangle \subseteq C$: Considere $c = \sigma_1 g_1 + \cdots + \sigma_t g_t$. A seguir, prova-se que cada $\sigma_i g_i \in C$. Portanto, $c \in C$.

$$\begin{aligned} \sigma_i g_i &= \cdots + \sigma_{ij} x^j g_i + \ldots \\ &= \cdots + (a_{ij}, b_{ij}) g_i^{(j)} + \ldots \\ &= \cdots + (a_{ij} f_i^{(j)} \star b_{ij} h_i^{(j)}) + \ldots \end{aligned}$$

Dado que $f_i \in \Theta_1$, $h_i \in \Theta_2$ e C_1 e C_2 são códigos cíclicos, segue que $(a_{ij} f_i^{(j)} \star b_{ij} h_i^{(j)}) \in C \ \forall j$. Portanto, $c \in C$.

■

Exemplo 5.3. *Considere*

- $A_1 = \mathbb{Z}_2$, $n_1 = 3$, $\mathcal{C}_1 \subset \mathbb{Z}_2[x]/\langle x^3 - 1 \rangle$, $\Theta_1 = \{1 + x + x^2\}$, $|\mathcal{C}_1| = 2$ *e* $d(\mathcal{C}_1) = 3$.
- $A_2 = \mathbb{Z}_3$, $n_2 = 2$, $\mathcal{C}_2 \subset \mathbb{Z}_3[x]/\langle x^2 - 1 \rangle$, $\Theta_2 = \{2 + x\}$, $|\mathcal{C}_2| = 3$ *e* $d(\mathcal{C}_2) = 2$.

O ***código produto estendido*** $\mathcal{C}$ *sobre* $\mathbb{Z}_2 \times \mathbb{Z}_3 \simeq \mathbb{Z}_6$, *tem comprimento* $n = 6$ *(*$\alpha = 2$ *e* $\beta = 3$*),* 6 *palavras-código (*$|\mathcal{C}| = 6$*), e distância mínima 6 (*$d(\mathcal{C}) = 6$*). Segundo o Lema 5.6:*

$$\Theta = \{(1 + x + x^2) \star (2 + x)\} = \{(5 + x + 5x^2 + x^3 + 5x^4 + x^5)\}.$$

5.4 Detecção e Correção de Erros

Dado que o código linear produto estendido, $\mathcal{C}$, é completamente descrito pelos código lineares $\mathcal{C}_1$ e $\mathcal{C}_2$, segue que ambos algoritmos para a detecção e correção de erros de $\mathcal{C}$ podem ser derivados dados os algoritmos, respectivos, para a detecção e correção de erros dos códigos $\mathcal{C}_1$ e $\mathcal{C}_2$.

5.4.1 Algoritmo de detecção de erros - $\mathcal{A}_d$

Sejam $\mathcal{A}_1$ e $\mathcal{A}_2$ dois algoritmos, tais que $\mathcal{A}_1(c_1)$ = "Verdadeiro" e $\mathcal{A}_2(c_2)$ = "Verdadeiro" se, e somente se, $c_1 \in \mathcal{C}_1$ e $c_2 \in \mathcal{C}_2$; caso contrário suas saídas são "Falso".

Teorema 5.3. *Usando a notação anterior, sejam* $x = (\dots, (a_i, b_i), \dots) \in (A_1 \times A_2)^n$ *sequência e* $\mathcal{A}_d$ *um algoritmo de detecção definido como:*

1: ***procedure*** $\mathcal{A}_d(x)$
2: ***if*** $(a_j == a_{(j)_{n_1}}$ *e* $b_j == b_{(j)_{n_2}})$, $\forall j$ ***then***
3: ***if*** $\mathcal{A}_1((a_0, \dots, a_{n_1-1})) ==$ *"Verdadeiro"* ***then***
4: ***if*** $\mathcal{A}_2((b_0, \dots, b_{n_2-1})) ==$ *"Verdadeiro"* ***then***
5: ***return*** *Verdadeiro*
return *Falso*

Assim, o algoritmo $\mathcal{A}_d(c)$ *retorna "Verdadeiro" se, e somente se,* $c \in \mathcal{C}$.

Demonstração:

- Primeiro, se $c \in \mathcal{C}$, então $\mathcal{A}_d(c) =$ "Verdadeiro". Dado que $c \in \mathcal{C}$, segue que existem $c_1 \in \mathcal{C}_1$ e $c_2 \in \mathcal{C}_2$ tais que: $c = c_1 \star c_2$, $\mathcal{A}_1(c_1) =$ "Verdadeiro" e $\mathcal{A}_2(c_2) =$ "Verdadeiro". Obtém-se $\mathcal{A}_d(c) =$ "Verdadeiro" ao aplicar a Definição 5.2.

- Como $\mathcal{A}_d(c) =$ "Verdadeiro", obtém-se as seguintes afirmações:

$$\mathcal{A}_1((a_0, \dots, a_{n_1-1})) = \text{"Verdadeiro"} \tag{5.23}$$

$$\mathcal{A}_2((b_0, \dots, b_{n_2-1})) = \text{"Verdadeiro"} \tag{5.24}$$

$$(a_j = a_{(j)_{n_1}} \text{ e } b_j = b_{(j)_{n_2}}), \quad \forall j. \tag{5.25}$$

  Sejam $c_1 = (a_0, \dots, a_{n_1-1})$ e $c_2 = (b_0, \dots, b_{n_2-1})$. Prova-se que $c_1 \in \mathcal{C}_1$ e $c_2 \in \mathcal{C}_2$ pelas Equações 5.23 e 5.24. Como a Equação 5.25 é equivalente à Definição 5.2, segue que $c = c_1 \star c_2$ e $c \in \mathcal{C}$.

■

Exemplo 5.4. *Sejam* $\mathcal{C}_1$, $\mathcal{C}_2$ *e* $\mathcal{C}$ *os códigos introduzidos no Exemplo 5.2. Propõe-se verificar se a sequência* $c = (7, 10, 7, 10) \in \mathbb{Z}_{12}^4$ *pertence a* $\mathcal{C}$. *Suponha que os algoritmos* $\mathcal{A}_1$ *e* $\mathcal{A}_2$ *são conhecidos. O seguinte passo é aplicar o algoritmo* $\mathcal{A}_d$.

$$\begin{aligned} c &= (7, 10, 7, 10) \\ &= ([1, 3], [1, 2], [1, 3], [1, 2]) \in (\mathbb{Z}_3 \times \mathbb{Z}_4)^4. \end{aligned}$$

- *Dado que* $n_1 = 2$ *e* $n_2 = 4$, *verifica-se que* $a_0 = 1 = a_2$, $a_1 = 1 = a_3$, $b_0 = 3$, $b_1 = 2$, $b_2 = 3$ *e* $b_3 = 2$.

- *Certamente,* $(1, 1) = c_1 \in \mathcal{C}_1$ *e* $(3, 2, 3, 2) = c_2 \in \mathcal{C}_2$. *Portanto,* $\mathcal{A}_1(c_1) =$ *"Verdadeiro"* *e* $\mathcal{A}_2(c_2) =$ *"Verdadeiro".*

Os passos anteriores mostraram que $c = (7, 10, 7, 10) \in \mathcal{C}$. *Além disso, usando o conjunto gerador:* $\Psi = \{(1, 4, 1, 4), (0, 6, 0, 6)\}$; *verifica-se que* $c = 7(1, 4, 1, 4) + 1(0, 6, 0, 6)$.

5.4.2 Algoritmo de correção de erros - $\mathcal{A}_c$

Sejam $\mathcal{A}_1$ e $\mathcal{A}_2$ dois algoritmos, tais que $\mathcal{A}_1(w_1) = c_1$ e $\mathcal{A}_2(w_2) = c_2$, onde w_1 e w_2 sequência com no máximo t_1 e t_2 erros com respeito a $c_1 \in \mathcal{C}_1$ e $c_2 \in \mathcal{C}_2$. Isto significa que $\mathcal{A}_1$ e $\mathcal{A}_2$ são dois algoritmos com a capacidade de corrigir t_1 e t_2 erros para os códigos $\mathcal{C}_1$ e $\mathcal{C}_2$, respectivamente.

Teorema 5.4. *Usando a notação anterior, sejam $x = (\dots, (a_i, b_i), \dots) \in (A_1 \times A_2)^n$ sequência e $\mathcal{A}_c$ um algoritmo de correção de erros definido como:*

1: ***procedure*** $\mathcal{A}_c(x)$
2: ***for*** $i \in \{0, \dots, n_1 - 1\}$ ***do***
3: $a'_i \leftarrow MaioriaDe\{a_i, a_{i+n_1}, \dots, a_{i+(\alpha-1)n_1}\}$
4: $w_1 \leftarrow (a'_0, \dots a'_{n_1-1})$
5: $c_1 \leftarrow \mathcal{A}_1(w_1)$
6: ***for*** $i \in \{0, \dots, n_2 - 1\}$ ***do***
7: $b'_i \leftarrow MaioriaDe\{b_i, b_{i+n_2}, \dots, b_{i+(\beta-1)n_2}\}$
8: $w_2 \leftarrow (b'_0, \dots b'_{n_2-1})$
9: $c_2 \leftarrow \mathcal{A}_2(w_2)$
10: ***return*** $c_1 \star c_2$

Assim, o algoritmo $\mathcal{A}_c$ corrige no máximo t erros, isto é, $c \leftarrow \mathcal{A}_c(x)$ retorna a palavra-código, $c \in \mathcal{C}$, mais próxima com respeito a uma sequência x, se x e c diferem no máximo t posições, onde:

$$t = \min\left\{ \left\lceil \frac{\alpha}{2} \right\rceil (t_1 + 1) - 1 \ , \left\lceil \frac{\beta}{2} \right\rceil (t_2 + 1) - 1 \right\}.$$

Demonstração: Considere $x = (\dots, (d_i, e_i), \dots) \in A^n$ como a sequência recebida, dado que $c = c_1 \star c_2$ foi transmitida, onde $c_1 = (\dots, a_i, \dots)$ e $c_2 = (\dots, b_i, \dots)$. O algoritmo $\mathcal{A}_c$ utiliza os algoritmos $\mathcal{A}_1$ e $\mathcal{A}_2$ para estimar c_1 e c_2, e assim, para poder corrigir corretamente as sequências $w_1 = (a'_0, \dots a'_{n_1-1})$ e $w_2 = (b'_0, \dots b'_{n_1-1})$, w_1 e w_2 devem diferir no máximo em t_1 e t_2 posições, quando comparadas com c_1 e c_2, respectivamente. Além disso, a_i é igual a a'_i se, e somente se, a maioria dos elementos em $\{d_i, d_{i+n_1}, \dots, d_{i+(\alpha-1)n_1}\}$ é igual a a_i. Quando nenhum erro ocorreu (i.e., $x = c$), obtém-se $a_i = d_i = \cdots = d_{i+(\alpha-1)n_1}$ pela Definição 5.2. Uma única diferença entre c_1 e w_1 implica que o teste da lógica majoritária falhou e que há uma quantidade maior ou igual a $\left\lceil \frac{\alpha}{2} \right\rceil$ de erros em x. Dado que o algoritmo $\mathcal{A}_1$ é capaz de corrigir no máximo t_1 erros, segue que uma correção não é bem-sucedida (i.e. $c_1 \neq w_1$), utilizando o algoritmo $\mathcal{A}_1$, quando há mais que $\left\lceil \frac{\alpha}{2} \right\rceil (t_1+1)$ diferenças entre c e w. Finalmente, conclui-se que se há uma quantidade menor ou igual a $\left\lceil \frac{\alpha}{2} \right\rceil (t_1+1)-1$ de diferenças entre c e w, então $c_1 = w_1$. Um procedimento lógico similar pode ser aplicado sobre os b_i's, b'_i's, e_i's, t_2, c_2 e w_2, para obter que se há uma quantidade menor ou igual

a $\left\lceil \frac{\alpha}{2} \right\rceil (t_2 + 1) - 1$ de diferenças entre c e w, então $c_2 = w_2$. Finalmente, o algoritmo $\mathcal{A}_c$ corrige erros corretamente quando o mínimo desses valores é assumido. ∎

Até aqui, somente erros randômicos e com distribuição uniforme têm sido considerados. Porém, dado que a Definição 5.2 foi inspirada na definição de códigos produto (BURTON; WELDON, 1965) e esses códigos estão equipados com a capacidade de corrigir erros do tipo "*Burst*", espera-se alguma capacidade de correção de erros do tipo "*Burst*" para os códigos $ECRT$. A seguir demonstra-se que o algoritmo introduzido no Teorema 5.4 é capaz de corrigir erros do tipo "*Burst*" cíclicos.

Analogamente ao caso anterior, sejam $\mathcal{A}_1$ e $\mathcal{A}_2$ dois algoritmos, tais que $\mathcal{A}_1(w_1) = c_1$ e $\mathcal{A}_2(w_2) = c_2$, onde w_1 e w_2 são sequências com erros do tipo "*Burst*" cíclicos com comprimento máximo B_1 e B_2, com respeito a $c_1 \in \mathcal{C}_1$ e $c_2 \in \mathcal{C}_2$. Isto significa que os algoritmos $\mathcal{A}_1$ e $\mathcal{A}_2$ são dois algoritmos para os códigos $\mathcal{C}_1$ e $\mathcal{C}_2$, respectivamente, capazes de corrigir erros do tipo "*Burst*" cíclicos com comprimentos B_1 e B_2, ou menores.

Teorema 5.5. *Usando a notação anterior, sejam:*

1. $x = (\dots, (a_i, b_i), \dots) \in (A_1 \times A_2)^n$*; e*
2. *o algoritmo* $\mathcal{A}_c$ *introduzido no Teorema 5.4.*

Assim, o algoritmo $\mathcal{A}_c$ *corrige erros do tipo "*Burst*" cíclicos com comprimento máximo:*

$$B = \min \left\{ n_1 \left\lfloor \frac{\alpha - 1}{2} \right\rfloor + B_1 \ , n_2 \left\lfloor \frac{\beta - 1}{2} \right\rfloor + B_2 \right\}.$$

Demonstração: Considere $x = (\dots, (d_i, e_i), \dots) \in A^n$ como a sequência recebida, dado que $c = c_1 \star c_2$ foi transmitida, onde $c_1 = (\dots, a_i, \dots)$ e $c_2 = (\dots, b_i, \dots)$. A seguir, analisa-se o caso limite tal que $\mathcal{A}_c(x) = c$, isto é, o máximo comprimento do erro tipo "*Burst*" que o algoritmo garante uma correção bem-sucedida. O algoritmo $\mathcal{A}_c$ utiliza os algoritmos $\mathcal{A}_1$ e $\mathcal{A}_2$ para estimar c_1 e c_2, e assim, precisa-se mostrar que $w_1 = (a'_0, \dots a'_{n_1-1})$ e $w_2 = (b'_0, \dots b'_{n_2-1})$ possuem no máximo erros do tipo "*Burst*" de comprimento B_1 e B_2, quando comparadas com c_1 e c_2, respectivamente. Primeiro, estuda-se a capacidade de correção de erros do tipo "*Burst*" para $(\dots, d_i, \dots) \in {A_1}^n$. Sejam $M_1, \dots, M_{n_1}$ os conjuntos nos quais a lógica majoritária será aplicada, onde $M_i = \{d_i, d_{i+n_1}, \dots, d_{i+(\alpha-1)n_1}\}$, para $i = 1, \dots, n_1$. Se um erro do tipo "*Burst*" de comprimento n_1 ocorre, então somente um único elemento no conjunto M_i é corrompido. Portanto, o teste da lógica majoritária é suficiente para corrigir erros do tipo "*Burst*" de comprimento máximo igual a: $n_1\lfloor(\alpha-1)/2\rfloor$. Dado que o algoritmo $\mathcal{A}_1$ está projetado para coriigir erros do tipo "*Burst*" de comprimento B_1, segue que, na notação vetorial $(\dots, d_i, \dots) \in {A_1}^n$, o algoritmo corrige erros do tipo "*Burst*" de comprimento:

$$n_1 \left\lfloor \frac{\alpha - 1}{2} \right\rfloor + B_1.$$

O mesmo procedimento lógico pode ser aplicado para estudar a capacidade de corrigir erros do tipo "*Burst*" para $(\dots, e_i, \dots) \in {A_2}^n$, obtendo que o algoritmo é capaz de corrigir erros do tipo "*Burst*" de comprimento:

$$n_2 \left\lfloor \frac{\beta - 1}{2} \right\rfloor + B_2.$$

Finalmente, a capacidade de correção de erros do tipo "*Burst*" é obtida ao tomar o mínimo dos valores obtidos anteriormente. ∎

Exemplo 5.5. *Considere os seguintes códigos BCH não-primitivos sobre* $\mathbb{Z}_4$ *e* $\mathbb{Z}_5$*, para construir um código cíclico sobre* $\mathbb{Z}_4 \times \mathbb{Z}_5 \cong \mathbb{Z}_{20}$ *de comprimento* $n = 84$*:*

- $\mathcal{C}_1$ *sobre* $\mathbb{Z}_4$ *com parâmetros: (21,14,3)*

$$\begin{aligned} g_1(x) =& (3+x)(1+x+3x^2+3x^4+2x^5+x^6) \\ =& 3+2x^2+3x^3+x^4+x^5+x^6+x^7. \end{aligned}$$

  O código $\mathcal{C}_1$ *corrige no máximo* $t_1 = 1$ *erros ao utilizar o algoritmo de Berlekam-Massey modificado,* $\mathcal{A}_1$*, para anéis com identidade (INTERLANDO* et al.*, 1997) e corrige erros do tipo "*Burst*" com comprimento* $B_1 = 1$*.*

- $\mathcal{C}_2$ *sobre* $\mathbb{Z}_5$ *com parâmetros: (28,14,4)*

$$\begin{aligned} g_2(x) =& (4+x)(4+3x+x^2+2x^3+4x^4+3x^5+x^6) \\ & (3+x)(1+4x+x^2+4x^3+1x^4+4x^5+x^6) \\ g_2(x) =& 3+x+1x^2+3x^3+x^5+x^6+2x^7+3x^8 \\ & +4x^9+3x^{11}+3x^{12}+4x^{13}+x^{14}. \end{aligned}$$

  O código $\mathcal{C}_2$ *corrige no máximo* $t_2 = 1$ *erros ao utilizar o algoritmo de Berlekam-Massey modificado,* $\mathcal{A}_2$*, e corrige erros do tipo "*Burst*" com comprimento* $B_2 = 1$*.*

Usando as propriedades e parâmetros de $\mathcal{C}_1$ *e* $\mathcal{C}_2$*, encontram-se os parâmetros de* $\mathcal{C} = ECTR(\mathcal{C}_1, \mathcal{C}_2)$*:* $(84, 14, 12)$*. Estes valores são obtidos como se mostra a seguir:*

- $n = lcm(21, 28) = 84$, $\gcd(21, 28) = 7$ *e*
  $(\alpha)(n_1) = n = (\beta)(n_2) \leftrightarrow (4)(21) = 84 = (28)(3)$.

- *Dado que* $\mathcal{C}_1$ *e* $\mathcal{C}_2$ *são módulos livres de posto* 14*, segue a partir do Lema 5.5 que* $\mathcal{C} = ECTR(\mathcal{C}_1, \mathcal{C}_2)$ *é um submódulo livre de posto* 14*. Assim,* $|\mathcal{C}| = |\mathcal{C}_1||\mathcal{C}_2| = (4^{14})(5^{14}) = 20^{14}$*.*

- $d(\mathcal{C}) = \min\{(4)(3), (3)(4)\} = 12$

Além disso, usando a notação polinomial e segundo o Lema 5.6, $\mathcal{C}$ é um ideal principal gerado por:
$g(x) = g_1(x) \star g_2(x)$ *($C = \langle g(x) \rangle$), e assim:*

$$\begin{aligned} g(x) =& 3 + 16x + 6x^2 + 3x^3 + 5x^4 + x^5 + x^6 + 17x^7 + 8x^8 \\ &+ 4x^9 + 8x^{11} + 8x^{12} + 4x^{13} + 16x^{14} + 15x^{21} \\ &+ 10x^{23} + 15x^{24} + 5x^{25} + 5x^{26} + 5x^{27} + 13x^{28} \\ &+ 16x^{29} + 16x^{30} + 8x^{31} + 16x^{33} + 16x^{34} + 12x^{35} \\ &+ 8x^{36} + 4x^{37} + 8x^{39} + 8x^{40} + 4x^{41} + 11x^{42} \\ &+ 10x^{44} + 15x^{45} + 5x^{46} + 5x^{47} + 5x^{48} + 5x^{49} \\ &+ 8x^{56} + 16x^{57} + 16x^{58} + 8x^{59} + 16x^{61} + 16x^{62} \\ &+ 7x^{63} + 8x^{64} + 14x^{65} + 15x^{66} + 13x^{67} + 13x^{68} \\ &+ 9x^{69} + x^{70}. \end{aligned}$$

*Usando os algoritmos $\mathcal{A}_1$ e $\mathcal{A}_2$ (para $\mathcal{C}_1$ e $\mathcal{C}_2$, respectivamente) e o algoritmo definido no Teorema 5.4, obtém-se que o algoritmo resultante pode corrigir t erros randômicos com distribuição uniforme e erros do tipo "*Burst*" cíclicos com comprimento B ou menores, onde:*

- $t = \min\left\{\left\lceil \frac{4}{2} \right\rceil (2) - 1 \ , \left\lceil \frac{3}{2} \right\rceil (2) - 1\right\} = 3$
- $B = \min\left\{21 \left\lfloor \frac{4-1}{2} \right\rfloor + 1 \ , 28 \left\lfloor \frac{3-1}{2} \right\rfloor + 1\right\} = 22$-

5.5 Considerações Finais

Neste capítulo, propõe-se um procedimento para a construção de códigos lineares e cíclicos sobre anéis finitos, comutativos e com identidade multiplicativa. A construção do código $\mathcal{C}$ sobre A está baseado na decomposição de A numa soma direta de anéis locais $A \simeq A_1 \times \cdots \times A_s$ e na justaposição de s códigos cíclicos $\mathcal{C}_1, \ldots, \mathcal{C}_s$ sobre $A_1, \ldots, A_s$, respectivamente, com não necessariamente iguais comprimentos. Os parâmetros $\mathcal{C}$ foram derivados dos parâmetros dos códigos $\mathcal{C}_1, \ldots, \mathcal{C}_s$, onde o comprimento do código é dado por $n = \text{lcm}(n_1, \ldots, n_s)$, a cardinalidade é dada por $|\mathcal{C}| = |\mathcal{C}_1| \cdot \ldots \cdot |\mathcal{C}_s|$ e a distância mínima de Hamming é dada por $d = \min\{\alpha_1 d_1, \ldots, \alpha_s d_s\}$, onde n_i é o comprimento e d_i é a distância mínima de Hamming do código $\mathcal{C}_i$, e α_i satisfaz: $n = \alpha_i n_i$. Um conjunto mínimo de geradores para $\mathcal{C}$ foi obtido e as condições para que $\mathcal{C}$ seja um submódulo livre foram estabelecidas. Dois algoritmos foram introduzidos: o primeiro para verificar se uma sequência sobre A pertence a $\mathcal{C}$ e o segundo para corrigir erros randômicos e do tipo "*Burst*" cíclicos.

Os parâmetros *taxa de informação de* $\mathcal{C}$, (R), e *distância mínima relativa de* $\mathcal{C}$, δ, são valores entre 0 e 1, e $\mathcal{C}$ é considerado um "Bom" código se ambos parâmetros R e δ são próximos a 1. Sejam R_i e δ_i os correspondentes parâmetros do código $\mathcal{C}_i$, $n = \alpha_1 n_1 = \cdots = \alpha_s n_s$ o comprimento de $\mathcal{C}$ e l o índice do código $\mathcal{C}_l$ tal que $d = \alpha_l d_l = \min\{\alpha_1 d_1, \ldots, \alpha_s d_s\}$. Então R e δ são obtidas das Equações 5.26 e 5.27,

$$\begin{aligned} R =& \frac{|C|}{|A|^n} = \left(\frac{|C_1|}{|A_1|^{(\alpha_1 n_1)}}\right) \cdots \left(\frac{|C_s|}{|A_s|^{(\alpha_s n_s)}}\right) \\ R =& \left(\frac{R_1}{|A_1|^{(\alpha_1 - 1)}}\right) \cdots \left(\frac{R_s}{|A_s|^{(\alpha_s - 1)}}\right), \end{aligned} \tag{5.26}$$

$$\delta = \frac{d}{n} = \frac{d_l}{n_l} = \delta_l. \tag{5.27}$$

A partir de R e δ, conclui-se que $\mathcal{C}$ não é melhor que algum dos códigos que o compõem $(\mathcal{C}_i)$. Porém, $\mathcal{C}$ é um código apropriado para a identificação de proteínas como palavras-código, devido ao fato que em geral o processo de identificação de proteínas utiliza códigos com baixa capacidade de correção. Adicionalmente, $\mathcal{C}$ é de fácil construção, os algoritmos de detecção e correção são simples e a estrutura matemática é bem conhecida.

6 Modelo para a Sínteses de Proteínas: Codificador Genético Concatenado

Em teoria de codificação, os códigos concatenados surgiram pela necessidade de uma ferramenta de comunicação confiável e eficiente, em termos de energia e largura de banda, com uma boa capacidade de correção de erros e baixa complexidade (ZYABLOV *et al.*, 1999). Segundo a aproximação de Zyablov (ZYABLOV *et al.*, 1999), a unificação do codificador de canal com o modulador num sistema de comunicação digital pode ser considerado como um codificador concatenado, e assim, de maneira análoga, o codificador genético e o ribossomo, segundo o modelo usado no Capítulo 4, podem ser unificados num único bloco, denotado como: *Codificador Genético Concatenado*.

Neste capítulo, estabelece-se pela primeira vez a noção de *Codificador Genético Concatenado* e se verifica, através do uso de códigos cíclicos sobre alfabetos $\mathbb{Z}_{20}$ e $\mathbb{F}_4 \times \mathbb{Z}_5$, a aplicabilidade deste conceito. Este capítulo está organizado da seguinte forma; na Seção 6.1, introduz-se o modelo do codificador genético concatenado para a síntese de proteínas; na Seção 6.2, detalha-se uma metodologia para a verificação do modelo pelo uso de códigos BCH e para a identificação de sequências de aminoácidos (proteínas) como palavras-código de um código corretor de erros; na Seção 6.3, mostra-se algumas proteínas identificadas como palavras-código de um codificador genético concatenado; na Seção 6.4, estuda-se a classe de proteínas: *cytochrome b6-f complex subunit 6-OS* de diferentes organismos, e comparam-se as propriedades biológicas conhecidas com as possíveis inferências obtidas através dos códigos corretores de erros; e na Seção 6.5, apresentam-se algumas considerações sobre os resultados obtidos durante o desenvolvimento deste capítulo.

6.1 Modelo: Codificador Genético Concatenado

Os códigos concatenados foram introduzidos pela primeira vez em 1966 (FORNEY, 1966) e é uma decomposição natural de um sistema de comunicação em duas partes: *Inner encoder* e *Outer encoder*. Existem duas interpretações equivalentes da codificação concatenada, a primeira pertence a D. Forney, na qual a codificação é considerada como um esquema de blocos consistindo do codificador externo seguido pelo codificador interno, canal e de decodificador interno e decodificador externo, os quais são considerados como um único super canal. A segunda interpretação pertence a V. Zyablov (ZYABLOV *et al.*, 1999), na qual o *Inner encoder* e o *Outer encoder* são agrupados num único codificador (Figura 23a). Assim, obtém-se um *Concatenated Encoder*, um *Channel* e um *Concatenated*

decoder. Esta abordagem permite o uso de um esquema de modulação como caso específico do *Inner encoder* (Figura 23b).

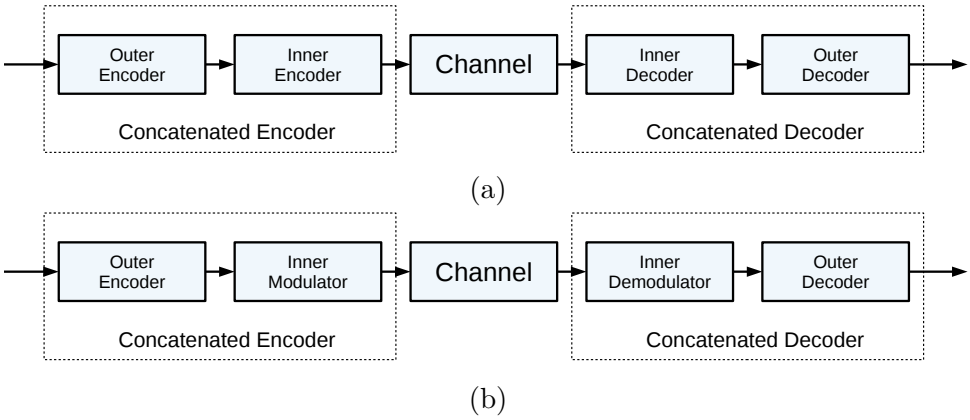


Figura 23 – Canal de comunicação com codificação concatenada: abordagem de Zyablov (ZYABLOV *et al.*, 1999).

Em (FARIA *et al.*, 2014), um modelo para o sistema de transmissão intra-celular de informação genética, similar ao modelo de um sistema de comunicação digital, é proposto (Figura 19). Este modelo é composto por um **codificador genético** que codifica a informação genética numa sequência de nucleotídeos (mRNA) casado ao código genético (visto como **modulador**), que representa o ribossomo e traduz as sequências mRNA em proteínas (sequência de aminoácidos). De acordo a V. Zyablov (ZYABLOV *et al.*, 1999), o codificador e o modulador podem ser agrupados em um único codificador, chamado de **Codificador Concatenado**. Portanto, neste capítulo, o conceito de **Codificador Genético Concatenado** é introduzido e agrupa o codificador genético e o ribossomo em um único codificador. Neste sentido, as proteínas podem ser estudadas como palavras-código de Códigos Corretores de Erros (Figura 24). De fato, a redundância é inerente às proteínas desde que, por exemplo, quando consideradas proteínas de tamanho 21, há $20^{21} = 2.097152 \times 10^{27}$ possíveis sequências de aminoácidos, porém, somente algumas dessas sequências são proteínas biologicamente funcionais. A existência de redundância justifica o uso de códigos corretores de erros para representar proteínas, para os quais, as seguintes relações entre os códigos corretores e as proteínas podem ser estabelecidas: 1) Sequências sobre um alfabeto de cardinalidade 20 **com** sequências de aminoácidos, 2) Palavras-código **com** proteínas biologicamente funcionais, 3) Um código corretor de erros **com** o conjunto de proteínas funcionais e 4) Sequências corregíveis para a palavra-código c **com** proteínas similares a uma proteína funcional específica.

Tendo em conta o exposto, neste capítulo, a teoria de informação e o modelo do *Codificador Genético Concatenado* são aplicados pela primeira vez na analise de sequências de aminoácidos para verificar seu uso potencial na detecção de relações biológicas entre proteínas. Para este objetivo, uma metodologia para representar ou identificar proteínas através de códigos cíclicos ($\mathcal{C} = ECRT(\mathcal{C}_4, \mathcal{C}_5)$) (ver Capítulo 5) sobre os alfabetos

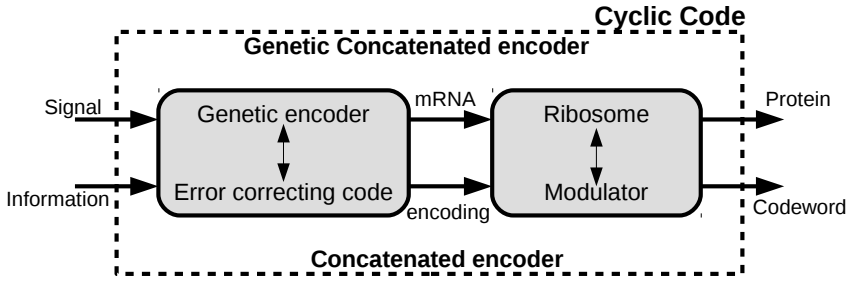


Figura 24 – O *Codificador Genético Concatenado* e a analogia entre o transmissor em um sistema de comunicação digital e a síntese de proteínas.

$\mathbb{Z}_{20}$ e $\mathbb{F}_4 \times \mathbb{Z}_5$ (ambos com 20 elementos) é proposta e desenvolvida. Os códigos BCH são considerados para $\mathcal{C}_4$ e $\mathcal{C}_5$ porque sua estrutura é bem conhecida, são relativamente fáceis de projetar e porque têm sido previamente usados no contexto biológico como **Codificador Genético** (BRANDÃO *et al.*, 2015; FARIA *et al.*, 2012; ROCHA *et al.*, 2010; FARIA *et al.*, 2010). Como caso de estudo, consideram-se as proteínas *Cytochrome b6-f complex subunit 6-OS* de diferentes organismos e são analisados através da metodologia proposta. Os resultados serão contrastados com as análises filogenética e taxonômica para descobrir relações entre essas sequências. Observa-se que, usando códigos BCH somente algumas sequências são identificadas, todas diferindo em um único aminoácido da sequência original.

6.2 Representação de Proteínas Através de Códigos Corretores de Erros

Para que os Códigos Corretores de Erros sejam capazes de detetar e/ou corrigir erros, estes devem estar associados a uma estrutura algébrica bem conhecida; sendo que as proteínas são sequências de aminoácidos e que 20 são os aminoácidos conhecidos que compõem as proteínas, então, os códigos corretores de erros que identificam proteínas como palavras-código devem ser construídos sobre alfabetos com 20 elementos. Os alfabetos com estruturas algébricas bem conhecidas são: o anel dos inteiros módulo 20 ($\mathbb{Z}_{20}$) e o produto cartesiano do corpo com 4 elementos e o anel dos inteiros módulo 5 ($\mathbb{F}_4 \times \mathbb{Z}_5$); $\mathbb{Z}_5$ também representa o corpo de 5 elementos ($\mathbb{F}_5$). Portanto, códigos corretores de erros ECRT sobre alfabetos $\mathbb{Z}_{20}$ e $\mathbb{F}_4 \times \mathbb{Z}_5$ que identifiquem proteínas como palavras-código serão construídos de acordo com os resultados apresentados no Capítulo 5.

De maneira similar ao realizado no Capítulo 4, o precedimento para a identificação de proteínas inicia ao mapear a sequência de aminoácidos numa sequência sobre $\mathbb{Z}_{20}$ e numa outra sobre $\mathbb{F}_4 \times \mathbb{Z}_5$. Porém, neste caso há 20 aminoácidos e, portanto, há $20! \approx 24,33 \times 10^{17}$ possíveis rotulamentos ou mapeamentos, os quais correspondem às $20!$ possíveis permutações. Devido à grande quantidade de rotulamentos, testar cada um

dos possíveis rotulamentos se torna numa tarefa pouco viável. Para superar esta limitação, neste trabalho, é proposto o uso dos rotulamentos derivados das representações da *Dayhoff's mutation odds matrix* (DAYHOFF; FOUNDATION, 1979) sobre o círculo, descritas por Taylor (TAYLOR, 1986) e Swanson (SWANSON, 1984). Nestas representações, os aminoácidos são localizados num círculo de acordo com a frequência de troca em mutações e as características químicas dos aminoácidos: hidrofobicidade e tamanho (Figura 25). A representação da matriz de Dayhoff no círculo lembra à representação matemática do anel $\mathbb{Z}_{20}$, e portanto, ao usar o rotulamento induzido pela matriz de Dayhoff no anel $\mathbb{Z}_{20}$, reduz-se o número de possíveis rotulamentos a 40, os quais são as simetrias do polígono de 20 lados. Cada uma das simetrias preserva os vizinhos correspondentes a cada aminoácido e formam o subgrupo diedral D_{20} (20 rotações e uma reflexão).

Como mencionado anteriormente, nos trabalhos de Taylor (TAYLOR, 1986) e Swanson (SWANSON, 1984), os aminoácidos foram alocados no círculo e a única diferença entre essas configurações é que os aminoácidos R e H têm sua posição trocada, como indicado pela seta na Figura 25. Dado que não existe algum conhecimento prévio sobre qual das duas configurações é a mais conveniente, segue que as duas configurações são testadas. Assim, 80 rotulamentos diferentes devem ser estudados; 40 para a configuração de Taylor e 40 para a configuração de Swanson.

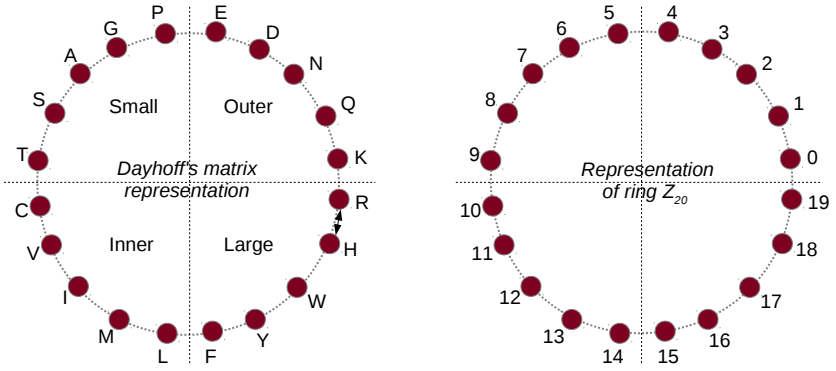


Figura 25 – *Dayhoff's mutation odds matrix* restrita ao círculo e a representação matemática do anel $\mathbb{Z}_{20}$.

O rotulamento obtido através da Figura 25 é usado para representar os aminoácidos através do anel $\mathbb{Z}_{20}$, porém esta não é a única estrutura algébrica com 20 elementos. O anel $\mathbb{F}_4 \times \mathbb{Z}_5$ também tem 20 elementos, e portanto, pode ser usado para representar proteínas. Novamente, o primeiro passo é mapear os aminoácidos em $\mathbb{F}_4 \times \mathbb{Z}_5$, para o qual se usa o esquema da Figura 25. Sabe-se que cada aminoácido pode ser representado como um elemento de $\mathbb{Z}_{20} \simeq (\mathbb{Z}_4 \times \mathbb{Z}_5)$ (uma dupla ordenada (a, b), onde $a \in \mathbb{Z}_4$ e $b \in \mathbb{Z}_5$) e que $\mathbb{F}_4$ é um $\mathbb{F}_2-$espaço vetorial de dimensão 2 (os seus elementos são duplas ordenadas (c, d) onde $c, d \in \mathbb{F}_2$, portanto, utiliza-se o mapa da $\mathbb{Z}_4-$linearidade (Equação 6.1) para identificar cada elemento de $(\mathbb{Z}_4 \times \mathbb{Z}_5)$ em $(\mathbb{F}_4 \times \mathbb{Z}_5)$, e assim, identificar cada aminoácido

Tabela 21 – Mapeamento entre aminoácidos e $\mathbb{Z}_{20}$ ou $\mathbb{F}_4 \times \mathbb{Z}_5$. Aqui $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2 = \beta\}$.

Amino acid	**Taylor**			**Swanson**		
	$\mathbb{Z}_{20}$	$\mathbb{Z}_4 \times \mathbb{Z}_5$	$\mathbb{F}_4 \times \mathbb{Z}_5$	$\mathbb{Z}_{20}$	$\mathbb{Z}_4 \times \mathbb{Z}_5$	$\mathbb{F}_4 \times \mathbb{Z}_5$
K	0	$(0,0)$	$(0,0)$	0	$(0,0)$	$(0,0)$
Q	1	$(1,1)$	$(1,1)$	1	$(1,1)$	$(1,1)$
N	2	$(2,2)$	$(\beta,2)$	2	$(2,2)$	$(\beta,2)$
D	3	$(3,3)$	$(\alpha,3)$	3	$(3,3)$	$(\alpha,3)$
E	4	$(0,4)$	$(0,4)$	4	$(0,4)$	$(0,4)$
P	5	$(1,0)$	$(1,0)$	5	$(1,0)$	$(1,0)$
G	6	$(2,1)$	$(\beta,1)$	6	$(2,1)$	$(\beta,1)$
A	7	$(3,2)$	$(\alpha,2)$	7	$(3,2)$	$(\alpha,2)$
S	8	$(0,3)$	$(0,3)$	8	$(0,3)$	$(0,3)$
T	9	$(1,4)$	$(1,4)$	9	$(1,4)$	$(1,4)$
C	10	$(2,0)$	$(\beta,0)$	10	$(2,0)$	$(\beta,0)$
V	11	$(3,1)$	$(\alpha,1)$	11	$(3,1)$	$(\alpha,1)$
I	12	$(0,2)$	$(0,2)$	12	$(0,2)$	$(0,2)$
M	13	$(1,3)$	$(1,3)$	13	$(1,3)$	$(1,3)$
L	14	$(2,4)$	$(\beta,4)$	14	$(2,4)$	$(\beta,4)$
F	15	$(3,0)$	$(\alpha,0)$	15	$(3,0)$	$(\alpha,0)$
Y	16	$(0,1)$	$(0,1)$	16	$(0,1)$	$(0,1)$
W	17	$(1,2)$	$(1,2)$	17	$(1,2)$	$(1,2)$
H	18	$(2,3)$	$(\beta,3)$	19	$(3,4)$	$(\alpha,4)$
R	19	$(3,4)$	$(\alpha,4)$	18	$(2,3)$	$(\beta,3)$

em $(\mathbb{F}_4 \times \mathbb{Z}_5)$ respeitando a distância, ou seja,

$$\mathbb{Z}_4 - \text{linearidade} \ : \ \{0 \mapsto 00, 1 \mapsto 10, 2 \mapsto 11, 3 \mapsto 01\}. \tag{6.1}$$

Na Tabela 21, mostra-se os rotulamentos canônicos (mapeamentos obtidos através da Figura 25 sem aplicar nenhuma rotação ou reflexão) entre os aminoácidos e $\mathbb{Z}_{20}$ e $\mathbb{F}_4 \times \mathbb{Z}_5$.

Considerando as informações expostas anteriormente, quatro possíveis **casos de estudo** são definidos para fazer a posterior análise (encontrar códigos corretores de erros cíclicos tais que uma determinada proteína é uma palavra-código):

- CCEs sobre o alfabeto $\mathbb{F}_4 \times \mathbb{Z}_5$ usando o rotulamento derivado por Taylor.
- CCEs sobre o alfabeto $\mathbb{F}_4 \times \mathbb{Z}_5$ usando o rotulamento derivado por Swanson.
- CCEs sobre o alfabeto $\mathbb{Z}_{20}$ usando o rotulamento derivado por Taylor.
- CCEs sobre o alfabeto $\mathbb{Z}_{20}$ usando o rotulamento derivado por Swanson.

No Capítulo 5, um procedimento para projetar código lineares e cíclicos sobre anéis comutativos com identidade foi introduzido com o propósito de usa-los na identificação

de proteínas como palavras-código. Lembre que o procedimento de projeto de códigos introduzido neste trabalho generaliza o procedimento padrão apresentado em (BLAKE, 1972) e permite a construção de códigos lineares e cíclicos sobre $\mathbb{Z}_4 \times \mathbb{Z}_5$ e sobre $\mathbb{F}_4 \times \mathbb{Z}_5$ pela justaposição de códigos cíclicos, não necessariamente, com comprimentos iguais. Como consequência, os códigos obtidos podem ser empregados para representar proteínas como palavras-código de códigos cíclicos.

6.2.1 Propriedades dos rotulamentos

O primeiro passo para estabelecer o algoritmo de identificação é mapear uma sequência dada de aminoácidos numa sequência sobre $\mathbb{Z}_{20}$ ou sobre $\mathbb{F}_4 \times \mathbb{Z}_5$, dependendo do caso de estudo. Ainda assim, como dito na seção anterior, para cada um dos 4 casos de estudo, todas as simetrias do polígono de 20 lados devem ser analisadas (subgrupo diedral D_{20}). De maneira similar que o descoberto para o algoritmo de identificação de sequências de nucleotídeos no Capítulo 4, os rotulamentos em D_{20} estão relacionados. A Afirmação 6.1 expõe as relações pertinentes à identificação de proteínas como palavras-código de códigos cíclicos sobre $\mathbb{Z}_{20}$.

Tabela 22 – Ação da rotação sobre o mapeamento da Figura 25.

Canônico:	→	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Canônico:	→	(0,0)	(1,1)	(2,2)	(3,3)	(0,4)	(1,0)	(2,1)	(3,2)	(0,3)	(1,4)	(2,0)	(3,1)	(0,2)	(1,3)	(2,4)	(3,0)	(0,1)	(1,2)	(2,3)	(3,4)
										⇓											
Rotação:	→	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	0
Rotação:	→	(1,1)	(2,2)	(3,3)	(0,4)	(1,0)	(2,1)	(3,2)	(0,3)	(1,4)	(2,0)	(3,1)	(0,2)	(1,3)	(2,4)	(3,0)	(0,1)	(1,2)	(2,3)	(3,4)	(0,0)

Afirmação 6.1. *Sem considerar o polinômio $x-1$ (verificação de paridade); se um código cíclico ($\mathcal{C}$) identifica como palavra-código uma sequência sobre $\mathbb{Z}_{20}$, obtida ao aplicar o mapeamento canônico numa sequência de aminoácidos de acordo com a Figura 25, então, o código $\mathcal{C}$ identifica como palavra-código todas as sequências sobre $\mathbb{Z}_{20}$ geradas pelos mapeamentos correspondentes às simetrias da Figura 25.*

Demonstração. O grupo de simetrias do polígono de 20 lados é gerado por somente duas ações: 1) Rotação e 2) Reflexão. Portanto, para mostrar que $\mathcal{C}$ gera todas as sequências obtidas através dos rotulamentos rotacionados e refletidos basta provar que $\mathcal{C}$ é invariante às ações de rotação e reflexão. Sejam $prot_e$ a sequência sobre $\mathbb{Z}_{20}$ obtida ao mapear uma dada sequência de aminoácidos através do rotulamento canônico, representado na Figura 25, $prot_{rot}$ a sequência sobre $\mathbb{Z}_{20}$ obtida ao mapear a sequência de aminoácidos através do rotulamento canônico rotacionado numa posição e $prot_{ref}$ a sequência sobre $\mathbb{Z}_{20}$ obtida ao mapear a sequência de aminoácidos através do rotulamento canônico refletido com respeito ao eixo formado pelos pontos 0 e 10. Primeiro demonstra-se a invariância com respeito a rotação. Observe que a ação de rotação (Tabela 22) corresponde a somar 1 em cada das posições do vetor, e assim, a sequência $prot_{rot}$ é obtida ao somar 1 em cada uma das posições da sequência $prot_e$. O código $\mathcal{C}$ é formado pela justaposição de dois códigos,

um sobre $\mathbb{Z}_4$ ($\mathcal{C}_4$) e outro sobre $\mathbb{Z}_5$ ($\mathcal{C}_5$). Logo se observa que a ação de rotação sobre cada uma das componentes $\mathbb{Z}_4$ e $\mathbb{Z}_5$ age da seguinte maneira:

$$\text{Rotação sobre } \mathbb{Z}_4 : \{0 \mapsto 1, 1 \mapsto 2, 2 \mapsto 3, 3 \mapsto 0\}$$
$$\text{Rotação sobre } \mathbb{Z}_5 : \{0 \mapsto 1, 1 \mapsto 2, 2 \mapsto 3, 3 \mapsto 4, 4 \mapsto 0\}.$$

o que significa que se deve somar 1 em cada uma das posições das sequências correspondentes. Dado que os códigos $\mathcal{C}_4$ e $\mathcal{C}_5$ são cíclicos, de acordo com a construção mostrada no Capítulo 5, segue que o polinômio $1 + x + x^2 + \cdots + x^{n-1}$ é uma palavra-código e, assim, $prot_{rot}$ é palavra-código de $\mathcal{C}$, somente se $prot_e$ é uma palavra-código de $\mathcal{C}$. Em seguida demonstra-se a invariância com respeito a reflexão. Observe que a ação de reflexão (Tabela 23) corresponde a multiplicar por $-1 \equiv 19 \pmod{20}$ cada uma das posições do vetor, e assim, a sequência $prot_{ref}$ é obtida ao multiplicar por -1 cada uma das posições da sequência $prot_e$. Dado que $\mathcal{C}$ é linear, de acordo com a construção mostrada no Capítulo 5, segue que se $prot_e$ é uma palavra-código de $\mathcal{C}$, então $prot_{ref} = 19 \cdot prot_e$ é uma palavra-código de $\mathcal{C}$.

Tabela 23 – Ação da reflexão com respeito ao eixo formado pelos elementos 0 e 10 sobre o mapeamento da Figura 25.

Canônico:	→	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Canônico:	→	(0,0)	(1,1)	(2,2)	(3,3)	(0,4)	(1,0)	(2,1)	(3,2)	(0,3)	(1,4)	(2,0)	(3,1)	(0,2)	(1,3)	(2,4)	(3,0)	(0,1)	(1,2)	(2,3)	(3,4)
										⇓											
Reflexão:	→	0	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Reflexão:	→	(0,0)	(3,4)	(2,3)	(1,2)	(0,1)	(3,0)	(2,4)	(1,3)	(0,2)	(3,1)	(2,0)	(1,4)	(0,3)	(3,2)	(2,1)	(1,0)	(0,4)	(3,3)	(2,2)	(1,1)

□

A Afirmação 6.1 é muito importante para o projeto do algoritmo que encontra um código cíclico tal que uma dada proteína é identificada como palavra-código, pois basta encontrar o rotulamento no grupo diedral no qual a paridade é zero e, somente para o mapeamento canônico, encontrar os polinômios minimais que dividem a sequência de aminoácidos; para assim analisar completamente uma proteína dada em todos os rotulamentos.

No caso da identificação de proteínas como palavras-código de códigos cíclicos sobre o alfabeto $\mathbb{F}_4 \times \mathbb{Z}_5$, o primeiro passo é representar a sequência de aminoácidos como uma sequência de elementos do alfabeto $\mathbb{F}_4 \times \mathbb{Z}_5$; para o qual se emprega a $\mathbb{Z}_4$-linearidade (Tabela 21). De igual maneira que o realizado para o alfabeto $\mathbb{Z}_{20}$, todas as simetrias do polígono de 20 lados devem ser analisadas (subgrupo diedral D_{20}). Porém, neste caso, o seguinte procedimento para considerar as simetrias é seguido: 1) Aplique a rotação e/ou reflexão sobre $\mathbb{Z}_{20} \simeq \mathbb{Z}_4 \times \mathbb{Z}_5$ e 2) Aplique a $\mathbb{Z}_4$-linearidade sobre a componente $\mathbb{Z}_4$ e considere o vetor $(\mathbb{Z}_2)^2$ como elemento de $\mathbb{F}_4$ para obter uma sequência sobre $\mathbb{F}_4 \times \mathbb{Z}_5$.

Utilizando o procedimento apresentado, a Afirmação 6.2 expõe as relações pertinentes à identificação de proteínas como palavras-código de códigos cíclicos sobre $\mathbb{F}_4 \times \mathbb{Z}_5$

Afirmação 6.2. *Sem considerar o polinômio $x-1$ (verificação de paridade); se um código cíclico ($\mathcal{C} = ECRT(\mathcal{C}_4, \mathcal{C}_5)$) com distância mínima d_{min} identifica como palavra-código uma sequência sobre $\mathbb{F}_4 \times \mathbb{Z}_5$, obtida ao aplicar o mapeamento canônico e a $\mathbb{Z}_4$-linearidade numa sequência de aminoácidos de acordo com a Figura 25 e Equação 6.1, então todas as sequências sobre $\mathbb{F}_4 \times \mathbb{Z}_5$, geradas pelos mapeamentos correspondentes às simetrias da Figura 25, são identificadas como palavras-código de algum dos códigos cíclicos $\mathcal{C}$ ou $\mathcal{C}' = ECRT(\overline{\mathcal{C}_4}, \mathcal{C}_5)$, ambos com distância mínima d_{min}; onde $\overline{\mathcal{C}_4}$ é o código conjugado de $\mathcal{C}_4$.*

Tabela 24 – Ação da rotação sobre o mapeamento da Figura 25 e a $\mathbb{Z}_4$-linearidade.

Canônico:	$\rightarrow$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Canônico:	$\rightarrow$	$(0,0)$	$(1,1)$	$(2,2)$	$(3,3)$	$(0,4)$	$(1,0)$	$(2,1)$	$(3,2)$	$(0,3)$	$(1,4)$	$(2,0)$	$(3,1)$	$(0,2)$	$(1,3)$	$(2,4)$	$(3,0)$	$(0,1)$	$(1,2)$	$(2,3)$	$(3,4)$
Canônico:	$\rightarrow$	$(0,0)$	$(1,1)$	$(\beta,2)$	$(\alpha,3)$	$(0,4)$	$(1,0)$	$(\beta,1)$	$(\alpha,2)$	$(0,3)$	$(1,4)$	$(\beta,0)$	$(\alpha,1)$	$(0,2)$	$(1,3)$	$(\beta,4)$	$(\alpha,0)$	$(0,1)$	$(1,2)$	$(\beta,3)$	$(\alpha,4)$
											$\Downarrow$										
Rotação:	$\rightarrow$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	0
Rotação:	$\rightarrow$	$(1,1)$	$(2,2)$	$(3,3)$	$(0,4)$	$(1,0)$	$(2,1)$	$(3,2)$	$(0,3)$	$(1,4)$	$(2,0)$	$(3,1)$	$(0,2)$	$(1,3)$	$(2,4)$	$(3,0)$	$(0,1)$	$(1,2)$	$(2,3)$	$(3,4)$	$(0,0)$
Rotação:	$\rightarrow$	$(1,1)$	$(\beta,2)$	$(\alpha,3)$	$(0,4)$	$(1,0)$	$(\beta,1)$	$(\alpha,2)$	$(0,3)$	$(1,4)$	$(\beta,0)$	$(\alpha,1)$	$(0,2)$	$(1,3)$	$(\beta,4)$	$(\alpha,0)$	$(0,1)$	$(1,2)$	$(\beta,3)$	$(\alpha,4)$	$(0,0)$

Demonstração. O grupo de simetrias do polígono de 20 lados é gerado por somente duas ações: 1) Rotação e 2) Reflexão. Portanto, deve-se provar que a identificação de proteínas é invariante às ações de rotação e reflexão. Sejam $prot_e$ a sequência sobre $\mathbb{F}_4 \times \mathbb{Z}_5$ obtida ao mapear uma dada sequência de aminoácidos através do rotulamento canônico e a $\mathbb{Z}_4$-linearidade, $prot_{rot}$ a sequência sobre $\mathbb{F}_4 \times \mathbb{Z}_5$ obtida ao mapear a sequência de aminoácidos através do rotulamento canônico rotacionado numa posição e a $\mathbb{Z}_4$-linearidade e $prot_{ref}$ a sequência sobre $\mathbb{F}_4 \times \mathbb{Z}_5$ obtida ao mapear a sequência de aminoácidos através do rotulamento canônico refletido com respeito ao eixo formado pelos pontos 0 e 10 e a $\mathbb{Z}_4$-linearidade. Primeiro demonstra-se a invariância com respeito a rotação onde a ação da rotação é detalhada na Tabela 24. O código $\mathcal{C}$ é formado pela justaposição de dois códigos, um sobre $\mathbb{F}_4$ ($\mathcal{C}_4$) e outro sobre $\mathbb{Z}_5$ ($\mathcal{C}_5$). Logo, observa-se que a ação de rotação sobre cada uma das componentes $\mathbb{F}_4$ e $\mathbb{Z}_5$ age da seguinte maneira:

$$\text{Rotação sobre } \mathbb{F}_4 : \{0 \mapsto 1, 1 \mapsto \beta, 2 \mapsto \alpha, 3 \mapsto 0\}$$
$$\text{Rotação sobre } \mathbb{Z}_5 : \{0 \mapsto 1, 1 \mapsto 2, 2 \mapsto 3, 3 \mapsto 4, 4 \mapsto 0\}.$$

Portanto, para $\mathbb{Z}_5$ deve-se somar 1 em cada uma das posições da sequência e como $\mathcal{C}_5$ é cíclico, segue que o polinômio $1 + x + x^2 + \cdots + x^{n-1}$ é uma palavra-código e, portanto, a componente $\mathbb{Z}_5$ da sequência $prot_{rot}$ é uma palavra-código de $\mathcal{C}_5$. Para $\mathbb{F}_4$, a transformação de rotação que deve ser aplicada a cada uma das posições da sequência sobre $\mathbb{F}_4$ é não linear e é descrita como $y = T(x) = \alpha \cdot \overline{x} + 1$. Note que as operações soma de 1 e multiplicação por α são invariantes porque $\mathcal{C}_4$ é cíclico e linear, respectivamente. A operação de conjugação não preserva o mesmo código $\mathcal{C}_4$, porém, como demonstrado na Afirmação 4.11, preserva os parâmetros do código (igual distância mínima). Logo, o código $\mathcal{C}' = ECRT(\overline{\mathcal{C}_4}, \mathcal{C}_5)$ identifica a sequência $prot_{rot}$ e $\mathcal{C}$ identifica a sequência de aminoácidos quando se aplicam

duas rotações. Em seguida, demonstra-se a invariância com respeito a reflexão onde a ação da reflexão é detalhada na Tabela 25. Observe que a ação de reflexão sobre cada uma das componentes $\mathbb{F}_4$ e $\mathbb{Z}_5$ age da seguinte maneira:

$$\text{Reflexão sobre } \mathbb{F}_4 : \{0 \mapsto 1, 1 \mapsto \beta, 2 \mapsto \alpha, 3 \mapsto 0\}$$
$$\text{Reflexão sobre } \mathbb{Z}_5 : \{0 \mapsto 0, 1 \mapsto 4, 2 \mapsto 3, 3 \mapsto 2, 4 \mapsto 1\}.$$

Portanto, para $\mathbb{Z}_5$ se deve multiplicar por $-1 \equiv 4 \pmod 5$ em cada uma das posições da sequência e como $\mathcal{C}_5$ é linear, segue que a componente $\mathbb{Z}_5$ da sequência $prot_{ref}$ é uma palavra-código de $\mathcal{C}_5$. Para $\mathbb{F}_4$, a transformação de reflexão que deve ser aplicada a cada uma das posições da sequência sobre $\mathbb{F}_4$ é não linear e é descrita como $y = T(x) = \alpha \cdot \overline{x}$. Como visto anteriormente, o código $\mathcal{C}' = ECRT(\overline{\mathcal{C}_4}, \mathcal{C}_5)$ identifica a sequência $prot_{ref}$ com os mesmos parâmetros que o código $\mathcal{C}$. □

Tabela 25 – Ação da reflexão com respeito ao eixo formado pelos elementos 0 e 10 sobre o mapeamento da Figura 25 e a $\mathbb{Z}_4$-linearidade.

Canônico:	→	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Canônico:	→	(0,0)	(1,1)	(2,2)	(3,3)	(0,4)	(1,0)	(2,1)	(3,2)	(0,3)	(1,4)	(2,0)	(3,1)	(0,2)	(1,3)	(2,4)	(3,0)	(0,1)	(1,2)	(2,3)	(3,4)
Canônico:	→	(0,0)	(1,1)	$(\beta,2)$	$(\alpha,3)$	(0,4)	(1,0)	$(\beta,1)$	$(\alpha,2)$	(0,3)	(1,4)	$(\beta,0)$	$(\alpha,1)$	(0,2)	(1,3)	$(\beta,4)$	$(\alpha,0)$	(0,1)	(1,2)	$(\beta,3)$	$(\alpha,4)$
										⇓											
Reflexão:	→	0	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Reflexão:	→	(0,0)	(3,4)	(2,3)	(1,2)	(0,1)	(3,0)	(2,4)	(1,3)	(0,2)	(3,1)	(2,0)	(1,4)	(0,3)	(3,2)	(2,1)	(1,0)	(0,4)	(3,3)	(2,2)	(1,1)
Reflexão:	→	(0,0)	$(\alpha,4)$	$(\beta,3)$	(1,2)	(0,1)	$(\alpha,0)$	$(\beta,4)$	(1,3)	(0,2)	$(\alpha,1)$	$(\beta,0)$	(1,4)	(0,3)	$(\alpha,2)$	$(\beta,1)$	(1,0)	(0,4)	$(\alpha,3)$	$(\beta,2)$	(1,1)

Novamente, a Afirmação 6.2 é muito importante para o projeto do algoritmo que encontra um código cíclico tal que uma dada proteína é identificada como palavra-código, pois basta encontrar o rotulamento no grupo diedral no qual a paridade é zero e, somente para o mapeamento canônico, encontrar os polinômios minimais que dividem a sequência de aminoácidos; para assim analisar completamente uma proteína dada em todos os rotulamentos.

6.2.2 Algoritmo para a identificação de proteínas

De maneira similar ao algoritmo apresentado na Seção 4.4, a ideia nesta seção é propor um algoritmo que identifique sequências de aminoácidos (proteínas) como palavras-código de códigos cíclicos formados a partir de códigos BCH, os quais modelam o processo de síntese de proteínas como um único *codificador genético concatenado*, o qual verifica a hipótese do uso de informação redundante na síntese de proteínas. O diagrama de blocos dos algoritmos $BCH\_OneProt\_\mathbb{Z}_{20}$ e $BCH\_OneProt\_\mathbb{F}_4 \times \mathbb{Z}_5$ usados para identificar sequências de aminoácidos para os quatro casos de estudo são introduzidos nas Figuras 26 (Taylor e Swanson sobre $\mathbb{Z}_{20}$) e 27 (Taylor e Swanson sobre $\mathbb{F}_4 \times \mathbb{Z}_5$), respectivamente.

Os dois algoritmos são similares, a menos da adição do bloco $\mathbb{Z}_4$-linearidade e que o algoritmo *BCH_One_Seq* procurará códigos sobre $\mathbb{F}_4$ ao invés de códigos sobre $\mathbb{Z}_4$. Como demonstrado na Seção 6.2.1, os algoritmos somente devem ser executados para o

rotulamento canônico, pois os rotulamentos simétricos levam a resultados similares ou iguais. A única diferença notável entre esses resultados é a paridade da sequência, pois para um único rotulamento no subgrupo das rotações a soma dos elementos da sequência é 0, i.e., o polinômio $x-1$ divide a sequência para um único rotulamento no subgrupo das rotações. O polinômio não pode reduzir a distância de projeto, porém, pode aumenta-la, e assim a paridade é calculada ao finalizar o algoritmo.

A seguir explicam-se cada um dos blocos que compõem os algoritmos:

- **Aplicação do rotulamento canônico:** Este bloco recebe uma sequência de aminoácidos (proteína) de tamanho n. No bloco, a sequência de aminoácidos é mapeada numa sequência no alfabeto $\mathbb{Z}_{20}$ de acordo com a Figura 25 e o caso que se deseja estudar (Taylor ou Swanson). A saída é uma sequência sobre o alfabeto $\mathbb{Z}_{20}$.

- **Divisor de sequências em componentes:** Este bloco recebe uma sequência sobre $\mathbb{Z}_{20}$ de tamanho n e a separa em duas sequências de tamanho n, uma sobre $\mathbb{Z}_4$ e a outra sobre $\mathbb{Z}_5$. O método de separação utiliza o Teorema Chinês do Resto, onde $(\mathbb{Z}_{20})^n \simeq {\mathbb{Z}_4}^n \times {\mathbb{Z}_5}^n$. A saída são as sequências sobre $\mathbb{Z}_4$ e sobre $\mathbb{Z}_5$.

- $\mathbb{Z}_4$**-Linearidade:** Neste bloco, aplica-se o mapeamento da $\mathbb{Z}_4$-linearidade (Equação 6.1). Assim, recebe-se uma sequência sobre $\mathbb{Z}_4$ de tamanho n e aplica-se a $\mathbb{Z}_4$-Linearidade em cada um dos elementos da sequência para obter uma outra sequência $(\mathbb{Z}_2 \times \mathbb{Z}_2)^n$. A saída deste bloco é uma sequência sobre $\mathbb{F}_4$ onde:

$$\{(0,0) \mapsto 0, (1,0) \mapsto 1, (0,1) \mapsto \alpha, (1,1) \mapsto \beta\}.$$

- *BCH_One_Seq***:** Revise o funcionamento deste algoritmo no Capítulo 3. Este bloco recebe uma sequência sobre $\mathbb{Z}_4$ ou $\mathbb{F}_4$ ou $\mathbb{Z}_5$. Este bloco encontra o polinômio gerador do maior código BCH ($\mathcal{C}$) de comprimento n e com a maior distância de projeto, tal que a sequência de entrada pertence a $\mathcal{C}$. A saída são os parâmetros do código BCH, o polinômio gerador ($g(x)$) e os polinômios minimais que fatoram $g(x)$.

- **Armazena resultado:** Neste bloco, o resultado é armazenado, somente se os código obtidos tiverem uma distância de projeto maior ou igual que 3.

- **Constrói o código ECRT:** Neste bloco, os códigos $\mathcal{C}_4$ e $\mathcal{C}_5$ são justapostos segundo a teoria vista no Capítulo 5 para formar o código $\mathcal{C} = \text{ECRT}(\mathcal{C}_4, \mathcal{C}_5)$. Os códigos $\mathcal{C}_4$ e $\mathcal{C}_5$ foram identificados pela aplicação do algoritmo *BCH_One_Seq* nas sequências respectivas (sobre os alfabetos $\mathbb{Z}_4$ ou $\mathbb{F}_4$ e $\mathbb{Z}_5$). Dado que ambos códigos são cíclicos e ideais principais, segue que $\mathcal{C} = \langle g(x) \rangle$, onde $g(x) = g_4(x) \star g_5(x)$.

- **Toda possível mutação foi estudada?** Este bloco controla o fluxo do algoritmo e verifica que cada uma das dezenove possíveis mutações em cada uma das possíveis

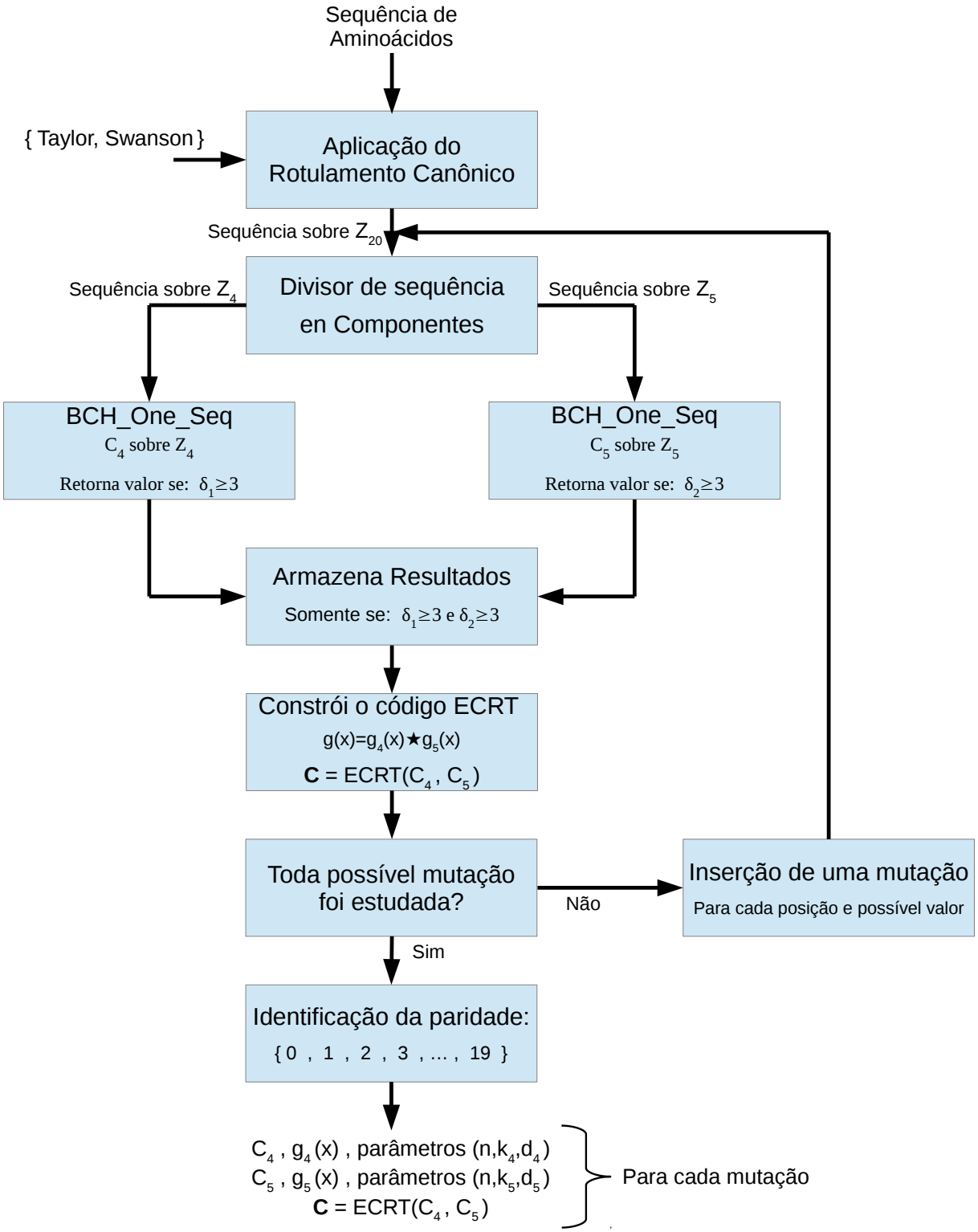


Figura 26 – $BCH\_OneProt\_\mathbb{Z}_{20}(prot)$

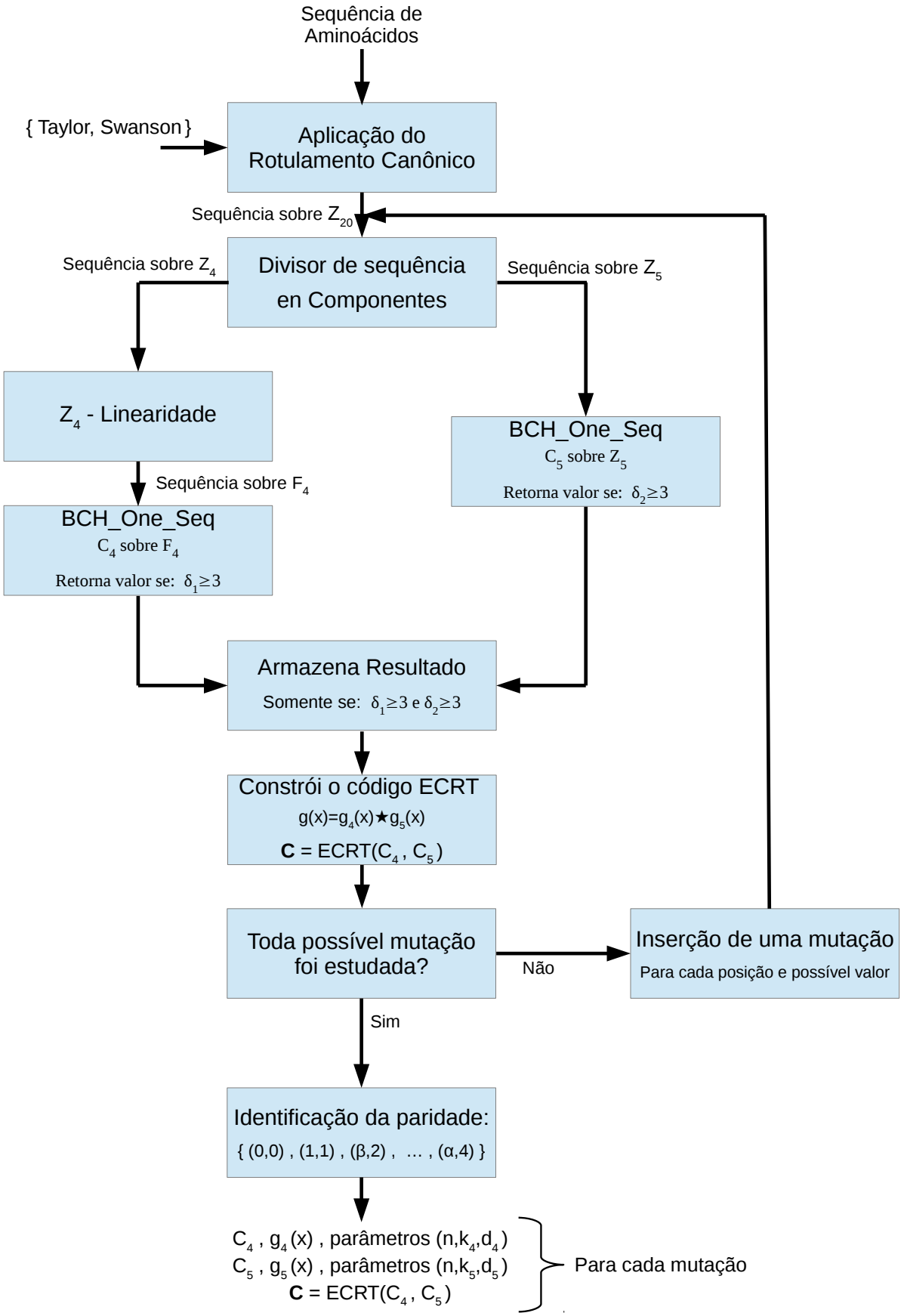


Figura 27 – $BCH\_OneProt\_\mathbb{F}_4 \times \mathbb{Z}_5(prot)$

posições sejam consideradas. Ele retorna "Não" quando alguma mutação ainda não foi considerada e retorna "Sim" quando todas as possíveis mutações de um único aminoácido já foram estudadas.

- **Inserção de uma mutação:** Este bloco insere uma única mutação na sequência de aminoácidos com respeito à sequência original. Dado que existem dezenove possíveis mutações para cada uma das posições, segue que o algoritmo *BCH_One_Seq* é executado: $2\cdot((n\cdot 19)+1)$ vezes. Esta é a justificativa para se procurar um algoritmo rápido e de poupar rotulamentos.

- **Identificação da paridade:** Este bloco calcula a paridade da sequência de aminoácidos sem mutação depois de ter sido mapeada sobre $\mathbb{Z}_{20}$ ou $\mathbb{F}_4 \times \mathbb{Z}_5$. Determina-se para qual das simetrias do rotulamento canônico o polinômio $x-1$ é um divisor ao verificar para qual dos rotulamentos a paridade é zero.

Note que nos algoritmos anteriores escolheu-se uma distância de projeto maior ou igual a 3 como critério de decisão se os códigos deviam ser armazenados e considerados. Este número foi determinado experimentalmente, pois encontrou-se que somente códigos BCH com distâncias de projeto pequenas eram capazes de identificar sequências de aminoácidos. Por outro lado, códigos com distâncias de projeto menores que três não foram considerados porque são códigos muito pobres e não são uteis para mostrar que o sistema biológico tem capacidade de detecção ou correção de erros.

O fato que o polinômio $x-1$ parece não ter muita importância não quer dizer que este nunca é considerado, pois, como foi visto na Seção 4.5, existem sequências para as quais o polinômio $x-1$ é necessário para obter uma distância de projeto maior ou igual que 3. Por esta razão, o rotulamento para o qual a paridade é zero será armazenado.

6.3 Exemplos de Proteínas Identificadas

Nesta seção, apresenta-se alguns exemplos de sequências de aminoácidos (proteínas funcionais) identificadas como palavras-código de códigos cíclicos através dos algoritmos apresentados na Seção 6.2.2 (Figuras 26 e 27).

Exemplo 6.1. *Considere a proteína identificada na Base de Dados Protein Data Bank como: "pdb/1WAA/ E from* Homo sapiens *organism" que possui a seguinte representação sobre aminoácidos:*

Oaa: *GAMALIEVEKPLYGVEVFVGETAHFEIELSEPDVHGQWKLKGQPLAASPDCEIIEDGKKHILILHNCQLG MTGEVSFQAANTKSAANLKVKEL*

*Aplica-se o algoritmo BCH_OneProt_*$\mathbb{Z}_{20}$ *(Figura 26) na proteína.*

- **Aplicação do rotulamento canônico:** Aplicam-se os mapeamentos Taylor e Swanson:

Tay: 6 7 13 7 14 12 4 11 4 0 5 14 16 6 11 4 11 15 11 6 4 9 7 18 15 4 12 4 14 8 4 5 3 11 18 6 1 17 0 14 0 6 1 5 14 7
7 8 5 3 10 4 12 12 4 3 6 0 0 18 12 14 12 14 18 2 10 1 14 6 13 9 6 4 11 8 15 1 7 7 2 9 0 8 7 7 2 14 0 11 0 4 14
Swa: 6 7 13 7 14 12 4 11 4 0 5 14 16 6 11 4 11 15 11 6 4 9 7 19 15 4 12 4 14 8 4 5 3 11 19 6 1 17 0 14 0 6 1 5 14 7
7 8 5 3 10 4 12 12 4 3 6 0 0 19 12 14 12 14 19 2 10 1 14 6 13 9 6 4 11 8 15 1 7 7 2 9 0 8 7 7 2 14 0 11 0 4 14

- **Divisor de sequências em componentes:** Este bloco recebe a sequência Tay e Swa sobre $\mathbb{Z}_{20}$ e as separa em duas sequências, uma sobre $\mathbb{Z}_4$ e a outra sobre $\mathbb{Z}_5$.
Taylor:

$\mathbb{Z}_4$:23132003001202303332013230002001332211020211233013200003200202022221221120303133210033220300 2
$\mathbb{Z}_5$:12324241400411141011442304244340313112040110422303042243100324243201413414130122240322240104 4

Swanson:

$\mathbb{Z}_4$:23132003001202303332013330002001333211020211233013200003200302023221221120303133210033220300 2
$\mathbb{Z}_5$:12324241400411141011442404244340314112040110422303042243100424244201413414130122240322240104 4

- *BCH_One_Seq***:** Este bloco é aplicado duas vezes para cada um dos pares de sequências sobre $\mathbb{Z}_4$ e $\mathbb{Z}_5$, mostradas acima, e para nenhum desses pares de sequências se encontra um código **ECRT** cíclico tal que essas sequências sejam palavras-código. Assim, estudam-se todas as possíveis mutações em uma única posição sobre as sequências **Tay** e **Swa** para identificar se há uma sequência mutação como palavra-código. Este procedimento faz sentido porque os códigos que serão encontrados têm distância maior ou igual a três, e assim a sequência original estaria na nuvem de sequência corrigíveis. A mutação é inserida pelo bloco: **Inserção de uma mutação**.

  Ao analisar as mutações, chega-se que somente a sequência **Tay** com uma única diferença com respeito a original é identificada como palavra-código de um código **CRT** com distância 3:

$\widehat{\text{Tay}}$: 6 7 13 7 14 12 4 11 4 0 5 14 16 **7** 11 4 11 15 11 6 4 9 7 18 15 4 12 4 14 8 4 5 3 11 18 6 1 17 0 14 0 6 1 5 14 7
7 8 5 3 10 4 12 12 4 3 6 0 0 18 12 14 12 14 18 2 10 1 14 6 13 9 6 4 11 8 15 1 7 7 2 9 0 8 7 7 2 14 0 11 0 4 14

$\widehat{\mathbb{Z}_4}$:23132003001203033320133300020013332110202112330132000032003020232212211203031332100332203002
$\widehat{\mathbb{Z}_5}$:12324241400411410114424042443403141120401104223030422431004242442014134141301222403222401044

  As características do código $\mathcal{C} = CRT(\mathcal{C}_4, \mathcal{C}_5)$ são mostradas a seguir:

  - $\mathcal{C}_4$: $g_4(x) = 3 + x + x^3 + 3x^4 + x^5 + x^6 + x^7$ e parâmetros $(93, 86, 3)$.
  - $\mathcal{C}_5$: $g_5(x) = 4 + 3x + 2x^3 + 3x^4 + x^5$ e parâmetros $(93, 88, 3)$.

  Estas informações são armazenadas no bloco **Armazena resultado**.

- **Identificação da paridade:** A paridade da sequência com a mutação é: 19. Porém, neste exemplo, os códigos $\mathcal{C}_4$ e $\mathcal{C}_5$ não são códigos nsBCH e a raiz inserida pela paridade não modifica a distância de projeto. Nem todos os códigos são sempre não *narrow-sense*, por esta razão é importante indicar o rotulamento no qual a paridade foi zero. Neste caso, o rotulamento é a rotação 17.
  Como visto na Seção 4.4.1, para alguns códigos a paridade leva numa distância de projeto maior que três e/ou é necessária para atingir uma distância de projeto igual a três.

Resumindo, a seguinte sequência de aminoácidos com uma mutação é identificada como palavra-código de um código CRT (ver Tabela 26):

Gaa: GAMALIEVEKPLYAVEVFVGETAHFEIELSEPDVHGQWKLKGQPLAASPDCEIIEDGKKHILILHNCQLGMTG EVSFQAANTKSAANLKVKEL

onde Gaa é a sequência com a mutação sugerida pelo código CRT.

Tabela 26 – Análise sobre $\mathbb{Z}_{20}$ da proteína: "pdb|1WAA| E from *Homo sapiens* organism"

Caso Taylor $\mathbb{Z}_{20}$. Rotulamento rotação 17. Código $\mathcal{C} = CRT(\mathcal{C}_4, \mathcal{C}_5)$: $n = 93$ e $d(\mathcal{C}) \geq 3$.

$\mathcal{C}_4$ **BCH:** $g_4(x) = 3 + x + x^3 + 3x^4 + x^5 + x^6 + x^7$ e parâmetros $(93, 86, 3)$
$\mathcal{C}_5$ **BCH:** $g_5(x) = 4 + 3x + 2x^3 + 3x^4 + x^5$ e parâmetros $(93, 88, 3)$

Rotulamento: $(D, E, P, G, A, S, \ldots, Q, N) \to (0, 1, 2, 3, 4, 5, \ldots, 18, 19)$

Oaa:	G	A	M	A	L	I	E	V	E	K	P	L	Y	**G**	V	E	V	F	V	G	E	T	A	H	F	E	I	E
Oar:	3	4	10	4	11	9	1	8	1	17	2	11	13	**3**	8	1	8	12	8	3	1	6	4	15	12	1	9	1
Gar:	3	4	10	4	11	9	1	8	1	17	2	11	13	**4**	8	1	8	12	8	3	1	6	4	15	12	1	9	1
Gaa:	G	A	M	A	L	I	E	V	E	K	P	L	Y	**A**	V	E	V	F	V	G	E	T	A	H	F	E	I	E

Oaa:	L	S	E	P	D	V	H	G	Q	W	K	L	K	G	Q	P	L	A	A	S	P	D	C	E	I	I	E	D
Oar:	11	5	1	2	0	8	15	3	18	14	17	11	17	3	18	2	11	4	4	5	2	0	7	1	9	9	1	0
Gar:	11	5	1	2	0	8	15	3	18	14	17	11	17	3	18	2	11	4	4	5	2	0	7	1	9	9	1	0
Gaa:	L	S	E	P	D	V	H	G	Q	W	K	L	K	G	Q	P	L	A	A	S	P	D	C	E	I	I	E	D

Oaa:	G	K	K	H	I	L	I	L	H	N	C	Q	L	G	M	T	G	E	V	S	F	Q	A	A	N	T	K	S
Oar:	3	17	17	15	9	11	9	11	15	19	7	18	11	3	10	6	3	1	8	5	12	18	4	4	19	6	17	5
Gar:	3	17	17	15	9	11	9	11	15	19	7	18	11	3	10	6	3	1	8	5	12	18	4	4	19	6	17	5
Gaa:	G	K	K	H	I	L	I	L	H	N	C	Q	L	G	M	T	G	E	V	S	F	Q	A	A	N	T	K	S

Oaa:	A	A	N	L	K	V	K	E	L
Oar:	4	4	19	11	17	8	17	1	11
Gar:	4	4	19	11	17	8	17	1	11
Gaa:	A	A	N	L	K	V	K	E	L

Exemplo 6.2. *Considere a proteína identificada na Base de Dados Protein Data Bank como: "pdb|2M9W| from* Homo sapiens *organism" que possui a seguinte representação sobre aminoácidos:*

Oaa: *SHMSASRRVGLSCANCQTTTTTLWRRNAEGEPVCNACGLYMKLHGVPRPLAMRKEGIQTRKRK*

Aplica-se o algoritmo BCH_OneProt_$\mathbb{F}_4 \times \mathbb{Z}_5$ (Figura 27) na proteína.

- **Aplicação do rotulamento canônico:** Aplicam-se os mapeamentos Taylor e

Swanson:

Tay: 8 18 13 8 7 8 19 19 11 6 14 8 10 7 2 10 1 9 9 9 9 9 14 17 19 19 2 7 4 6 4 5 11 10 2 7 10 6 14 16 13 0 14 18 6 11 5 19 5 14 7 13 19 0 4 6 12 1 9 19 0 19 0

Swa: 8 19 13 8 7 8 18 18 11 6 14 8 10 7 2 10 1 9 9 9 9 9 14 17 18 18 2 7 4 6 4 5 11 10 2 7 10 6 14 16 13 0 14 19 6 11 5 18 5 14 7 13 18 0 4 6 12 1 9 18 0 18 0

- **Divisor de sequências em componentes:** Este bloco recebe a sequência Tay e Swa sobre $\mathbb{Z}_{20}$ e as separa em duas sequências, uma sobre $\mathbb{Z}_4$ e a outra sobre $\mathbb{Z}_5$.
  Taylor:

  $\mathbb{Z}_4$:021030333220232211111121332302013223222010222313123130020113030

  $\mathbb{Z}_5$:333323441143022014444442442241401022014130431104042340412144040

  Swanson:

  $\mathbb{Z}_4$:031030223220232211111121222302013223222010232312123120020112020

  $\mathbb{Z}_5$:343323331143022014444442332241401022014130441103042330412143030

- $\mathbb{Z}_4$**-Linearidade:** Aplica-se o mapeamento da $\mathbb{Z}_4$-Linearidade (Equação 6.1) e obtém-se uma uma sequência sobre $\mathbb{F}_4$.
  Taylor:

  $\mathbb{F}_4$:0β10α0$\alpha\alpha\alpha\beta\beta$0$\beta\alpha\beta\beta$111111$\beta$1$\alpha\alpha\beta\alpha$0$\beta$01$\alpha\beta\beta\alpha\beta\beta\beta$010$\beta\beta\beta\alpha$1$\alpha$1$\beta\alpha$1$\alpha$00$\beta$011$\alpha$0$\alpha$0

  $\mathbb{Z}_5$:333323441143022014444442442241401022014130431104042340412144040

  Swanson:

  $\mathbb{F}_4$:0α10α0$\beta\beta\alpha\beta\beta$0$\beta\alpha\beta\beta$111111$\beta$1$\beta\beta\beta\alpha$0$\beta$01$\alpha\beta\beta\alpha\beta\beta\beta$010$\beta\alpha\beta\alpha$1$\beta$1$\beta\alpha$1$\beta$00$\beta$011$\beta$0$\beta$0

  $\mathbb{Z}_5$:343323331143022014444442332241401022014130441103042330412143030

- *BCH_One_Seq***:** Este bloco é aplicado duas vezes para cada um dos pares de sequências sobre $\mathbb{F}_4$ e $\mathbb{Z}_5$, mostradas acima, e para nenhum desses pares de sequências se encontra um código **ECRT** cíclico tal que essas sequências sejam palavras-código. Assim, estudam-se todas as possíveis mutações em uma única posição sobre as sequências **Tay** e **Swa** para identificar se há uma sequência mutação como palavra-código. A mutação é inserida pelo bloco: **Inserção de uma mutação**.

  Ao analisar as mutações, chega-se que somente a sequência **Tay** com uma única diferença com respeito a original é identificada como palavra-código de um código **CRT** com distância 3:

  $\widehat{\text{Tay}}$: 8 18 13 8 7 8 19 19 11 6 14 8 10 7 2 10 1 9 9 9 9 9 14 17 19 19 2 7 4 6 4 5 11 10 2 7 10 6 14 16 13 0 14 18 6 11 5 19 5 14 7 13 **19** 4 6 12 1 9 19 0 19 0

  $\widehat{\mathbb{Z}_4}$:021030333220232211111121332302013223222010222313123133020113030

  $\widehat{\mathbb{F}_4}$:0β10α0$\alpha\alpha\alpha\beta\beta$0$\beta\alpha\beta\beta$111111$\beta$1$\alpha\alpha\beta\alpha$0$\beta$01$\alpha\beta\beta\alpha\beta\beta\beta$010$\beta\beta\beta\alpha$1$\alpha$1$\beta\alpha$1$\alpha\boldsymbol{\alpha}0\beta$011$\alpha$0$\alpha$0

  $\widehat{\mathbb{Z}_5}$:333323441143022014444442442241401022014130431104042344**1**2144040

  As características do código $\mathcal{C} = CRT(\mathcal{C}_4, \mathcal{C}_5)$ são mostradas a seguir:

– $\mathcal{C}_4$: $g_4(x) = \alpha + \beta x^3 + x^4$ e parâmetros $(63, 59, 3)$.

– $\mathcal{C}_5$: $g_5(x) = 4 + 4x^2 + 2x^3 + 3x^4 + x^5 + x^7$ e parâmetros $(63, 56, 4)$.

Estas informações são armazenadas no bloco **Armazena resultado**.

- **Identificação da paridade:** A paridade da sequência com a mutação é: 0. Neste exemplo, os códigos $\mathcal{C}_4$ e $\mathcal{C}_5$ são códigos nsBCH e revBCH, respectivamente, e portanto, a raiz inserida pela paridade modifica a distância de projeto.
  De fato, para $\mathcal{C}_4$ e para $\mathcal{C}_5$ a raiz inserida pela paridade é necessária para atingir uma distância de projeto maior ou igual que 3.

Resumindo, a seguinte sequência de aminoácidos com uma mutação é identificada como palavra-código de um código CRT (ver Tabela 27):

Gaa: SHMSASRRVGLSCANCQTTTTTLWRRNAEGEPVCNACGLYMKLHGVPRPLAMRREGIQTRKRKMTGEVSF QAANTKSAANLKVKEL

onde Gaa é a sequência com a mutação sugerida pelo código CRT.

Tabela 27 – Análise sobre $\mathbb{F}_4 \times \mathbb{Z}_5$ da proteína: "pdb|2M9W| from *Homo sapiens* organism"

Caso Taylor $\mathbb{F}_4 \times \mathbb{Z}_5$. Rotulamento rotação 0. Código $\mathcal{C} = CRT(\mathcal{C}_4, \mathcal{C}_5)$: $n = 63$ e $d(\mathcal{C}) \geq 3$.

$\mathcal{C}_4$ **nsBCH:** $g_4(x) = \alpha + \beta x^3 + x^4$ e parâmetros $(63, 59, 3)$
$\mathcal{C}_5$ **revBCH:** $g_5(x) = 4 + 4x^2 + 2x^3 + 3x^4 + x^5 + x^7$ e parâmetros $(63, 56, 4)$

Rotulamento: $(K, Q, N, D, E, P, \ldots, H, R) \to (0, 1, 2, 3, 4, 5, \ldots, 18, 19)$

Oaa:	S	H	M	S	A	S	R	R	V	G	L	S	C	A	N	C
Oar:	$(0,3)$	$(\beta,3)$	$(1,3)$	$(0,3)$	$(\alpha,2)$	$(0,3)$	$(\alpha,4)$	$(\alpha,4)$	$(\alpha,1)$	$(\beta,1)$	$(\beta,4)$	$(0,3)$	$(\beta,0)$	$(\alpha,2)$	$(\beta,2)$	$(\beta,0)$
Gar:	$(0,3)$	$(\beta,3)$	$(1,3)$	$(0,3)$	$(\alpha,2)$	$(0,3)$	$(\alpha,4)$	$(\alpha,4)$	$(\alpha,1)$	$(\beta,1)$	$(\beta,4)$	$(0,3)$	$(\beta,0)$	$(\alpha,2)$	$(\beta,2)$	$(\beta,0)$
Gaa:	S	H	M	S	A	S	R	R	V	G	L	S	C	A	N	C
Oaa:	Q	T	T	T	T	T	L	W	R	R	N	A	E	G	E	P
Oar:	$(1,1)$	$(1,4)$	$(1,4)$	$(1,4)$	$(1,4)$	$(1,4)$	$(\beta,4)$	$(1,2)$	$(\alpha,4)$	$(\alpha,4)$	$(\beta,2)$	$(\alpha,2)$	$(0,4)$	$(\beta,1)$	$(0,4)$	$(1,0)$
Gar:	$(1,1)$	$(1,4)$	$(1,4)$	$(1,4)$	$(1,4)$	$(1,4)$	$(\beta,4)$	$(1,2)$	$(\alpha,4)$	$(\alpha,4)$	$(\beta,2)$	$(\alpha,2)$	$(0,4)$	$(\beta,1)$	$(0,4)$	$(1,0)$
Gaa:	Q	T	T	T	T	T	L	W	R	R	N	A	E	G	E	P
Oaa:	V	C	N	A	C	G	L	Y	M	K	L	H	G	V	P	R
Oar:	$(\alpha,1)$	$(\beta,0)$	$(\beta,2)$	$(\alpha,2)$	$(\beta,0)$	$(\beta,1)$	$(\beta,4)$	$(0,1)$	$(1,3)$	$(0,0)$	$(\beta,4)$	$(\beta,3)$	$(\beta,1)$	$(\alpha,1)$	$(1,0)$	$(\alpha,4)$
Gar:	$(\alpha,1)$	$(\beta,0)$	$(\beta,2)$	$(\alpha,2)$	$(\beta,0)$	$(\beta,1)$	$(\beta,4)$	$(0,1)$	$(1,3)$	$(0,0)$	$(\beta,4)$	$(\beta,3)$	$(\beta,1)$	$(\alpha,1)$	$(1,0)$	$(\alpha,4)$
Gaa:	V	C	N	A	C	G	L	Y	M	K	L	H	G	V	P	R
Oaa:	P	L	A	M	R	**K**	E	G	I	Q	T	R	K	R	K	
Oar:	$(1,0)$	$(\beta,4)$	$(\alpha,2)$	$(1,3)$	$(\alpha,4)$	$\mathbf{(0,0)}$	$(0,4)$	$(\beta,1)$	$(0,2)$	$(1,1)$	$(1,4)$	$(\alpha,4)$	$(0,0)$	$(\alpha,4)$	$(0,0)$	
Gar:	$(1,0)$	$(\beta,4)$	$(\alpha,2)$	$(1,3)$	$(\alpha,4)$	$\mathbf{(\alpha,4)}$	$(0,4)$	$(\beta,1)$	$(0,2)$	$(1,1)$	$(1,4)$	$(\alpha,4)$	$(0,0)$	$(\alpha,4)$	$(0,0)$	
Gaa:	P	L	A	M	R	**R**	E	G	I	Q	T	R	K	R	K	

Exemplo 6.3. *Considere a proteína identificada na Base de Dados NCBI como: "gi|156339520| from* Nematostella vectensis *organism" que possui a seguinte representação sobre aminoácidos:*

Oaa: *LPSGLAELPSGLVELPSGLVELPSGLVELPSGLAELPSGLVELPSGLVELPSGLAELPSGLVELPSGLVEL PSGLAELPSGLVE*

Aplica-se o algoritmo BCH_OneProt_$\mathbb{Z}_{20}$ (Figura 26) na proteína.

- **Aplicação do rotulamento canônico:** Aplicam-se os mapeamentos Taylor e Swanson. Os quais levam aos mesmos resultados, pois a proteína não tem aminoácidos R nem H.

Tay: 14 5 8 6 14 7 4 14 5 8 6 14 11 4 14 5 8 6 14 11 4 14 5 8 6 14 11 4 14 5 8 6 14 7 4 14 5 8 6 14 11 4 14
5 8 6 14 11 4 14 5 8 6 14 7 4 14 5 8 6 14 11 4 14 5 8 6 14 11 4 14 5 8 6 14 7 4 14 5 8 6 14 11 4
Swa: 14 5 8 6 14 7 4 14 5 8 6 14 11 4 14 5 8 6 14 11 4 14 5 8 6 14 11 4 14 5 8 6 14 7 4 14 5 8 6 14 11 4 14
5 8 6 14 11 4 14 5 8 6 14 7 4 14 5 8 6 14 11 4 14 5 8 6 14 11 4 14 5 8 6 14 7 4 14 5 8 6 14 11 4

- **Divisor de sequências em componentes:** Este bloco recebe a sequência Tay e Swa sobre $\mathbb{Z}_{20}$ e as separa em duas sequências, uma sobre $\mathbb{Z}_4$ e a outra sobre $\mathbb{Z}_5$. **Taylor e Swanson:**

$\mathbb{Z}_4$:21022302102230210223021022302102230210223021022302102230210223021022302102230
$\mathbb{Z}_5$:40314244031414403141440314144031424403141440314144031424403141440314144031424403141414

- *BCH_One_Seq***:** Este bloco é aplicado duas vezes para cada um dos pares de sequências sobre $\mathbb{Z}_4$ e $\mathbb{Z}_5$, mostradas acima, e para nenhum desses pares de sequências se encontra um código **ECRT** cíclico tal que essas sequências sejam palavras-código. Assim, estudam-se todas as possíveis mutações em uma única posição sobre a sequência **Tay** ou **Swa** para identificar se há uma sequência mutação como palavra-código. Ao analisar as mutações, chega-se que somente a seguinte sequência com uma única diferença com respeito a original é identificada como palavra-código de um código **ECRT** com distância 3:

$\widehat{\text{Tay}}$: 14 5 8 6 14 7 4 14 5 8 6 14 11 4 14 5 8 6 14 11 4 14 5 8 6 14 11 4 14 5 8 6 14 7 4 14 5 8 6 14 11 4 14
5 8 6 14 11 4 14 5 8 6 14 7 4 14 5 8 6 14 **7** 4 14 5 8 6 14 11 4 14 5 8 6 14 7 4 14 5 8 6 14 11 4

$\widehat{\mathbb{Z}_4}$:21022302102230210223021022302102230210223021022302102230**3**0210223021022302102230
$\widehat{\mathbb{Z}_5}$:40314244031414403141440314144031424403141440314144031424403**2**44031414403142440314144

As características do código $\mathcal{C} = ECRT(\mathcal{C}_4, \mathcal{C}_5)$ são mostradas a seguir:

  - $\mathcal{C}_4$: $g_4(x) = 1 + 2x + 3x^2 + 3x^4 + x^5 + x^6$ e parâmetros $(21, 15, 4)$.
  - $\mathcal{C}_5$: $g_5(x) = 4 + 2x + x^2 + 3x^3 + 3x^4 + 4x^6 + x^7 + 4x^9 + 4x^{11} + x^{12}$ e parâmetros $(84, 72, 4)$.

  Estas informações são armazenadas no bloco **Armazena resultado**.

- **Identificação da paridade:** A paridade da sequência com a mutação é: 4. Neste exemplo, os códigos $\mathcal{C}_4$ e $\mathcal{C}_5$ são códigos nsBCH, e portanto a raiz inserida pela paridade modifica a distância de projeto de $\mathcal{C}_4$ e $\mathcal{C}_5$, as quais passam de 3 a 4 quando a paridade é considerada.

Resumindo, a seguinte sequência de aminoácidos com uma mutação é identificada como palavra-código de um código $ECRT$ (ver Tabela 28):

Gaa: LPSGLAELPSGLVELPSGLVELPSGLVELPSGLAELPSGLVELPSGLVELPSGLAELPSGL**A**ELPSGLVELPSG LAELPSGLVE

onde Gaa é a sequência com a mutação sugerida pelo código ECRT.

Tabela 28 – Análise sobre $\mathbb{Z}_{20}$ da proteína: "gi|156339520| from *Nematostella vectensis* organism"

Caso Taylor/Swanson $\mathbb{Z}_{20}$. Rotulamento rotação 4. Código $\mathcal{C} = ECRT(\mathcal{C}_4, \mathcal{C}_5)$: $n = 84$ e $d(\mathcal{C}) \geq 3$

$\mathcal{C}_4$ **BCH:** $g_4(x) = 1 + 2x + 3x^2 + 3x^4 + x^5 + x^6$ e parâmetros $(21, 15, 4)$
$\mathcal{C}_5$ **BCH:** $g_5(x) = 4 + 2x + x^2 + 3x^3 + 3x^4 + 4x^6 + x^7 + 4x^9 + 4x^{11} + x^{12}$ e parâmetros $(84, 72, 4)$

Rotulamento: $(Y, W, H, R, K, Q, \ldots, L, F) \to (0, 1, 2, 3, 4, 5, \ldots, 18, 19)$

Oaa:	L	P	S	G	L	A	E	L	P	S	G	L	V	E	L	P	S	G	L	V	E	L	P	S	G	L	V	E	L
Oar:	18	9	12	10	18	11	8	18	9	12	10	18	15	8	18	9	12	10	18	15	8	18	9	12	10	18	15	8	18
Gar:	18	9	12	10	18	11	8	18	9	12	10	18	15	8	18	9	12	10	18	15	8	18	9	12	10	18	15	8	18
Gaa:	L	P	S	G	L	A	E	L	P	S	G	L	V	E	L	P	S	G	L	V	E	L	P	S	G	L	V	E	L

Oaa:	P	S	G	L	A	E	L	P	S	G	L	V	E	L	P	S	G	L	V	E	L	P	S	G	L	A	E	L	P
Oar:	9	12	10	18	11	8	18	9	12	10	18	15	8	18	9	12	10	18	15	8	18	9	12	10	18	11	8	18	9
Gar:	9	12	10	18	11	8	18	9	12	10	18	15	8	18	9	12	10	18	15	8	18	9	12	10	18	11	8	18	9
Gaa:	P	S	G	L	A	E	L	P	S	G	L	V	E	L	P	S	G	L	V	E	L	P	S	G	L	A	E	L	P

Oaa:	S	G	L	**V**	E	L	P	S	G	L	V	E	L	P	S	G	L	A	E	L	P	S	G	L	V	E
Oar:	12	10	18	**15**	8	18	9	12	10	18	15	8	18	9	12	10	18	11	8	18	9	12	10	18	15	8
Gar:	12	10	18	**11**	8	18	9	12	10	18	15	8	18	9	12	10	18	11	8	18	9	12	10	18	15	8
Gaa:	S	G	L	**A**	E	L	P	S	G	L	V	E	L	P	S	G	L	A	E	L	P	S	G	L	V	E

A seguir apresentam-se algumas proteínas analisadas pelos algoritmos $BCH\_OneProt\_\mathbb{Z}_{20}$ e $BCH\_OneProt\_\mathbb{F}_4 \times \mathbb{Z}_5$ e que foram identificadas como palavras-código ou como sequências na nuvem de sequências corregíveis. As características e parâmetros de cada um desses códigos são resumidos na Tabela 29.

- pdb|2XA0| from *Mus musculus* organism - A
- sp|P86841| from *Clitoria ternatea* organism - B
- pdb|1WAA| E from *Homo sapiens* organism - C
- sp|Q2SS74| from *Mycoplasma capricolum* organism - D
- sp|Q05190| from *Hordeum vulgare* organism - E
- pdb|2M9W| from *Homo sapiens* organism - F
- sp|Q06J23| from *Bigelowiella natans* organism - G
- gi|573572947| from *Bacillus weihenstephanensis* organism - H
- gi|156339520| from *Nematostella vectensis* organism - I

Tabela 29 – Algumas proteínas identificadas como palavras-código de códigos cíclicos e as propriedades desses códigos.

ID	Mutação	Pos.	$d(C)$	n_C	Alfabeto	n_{C_4} n_{C_5}	$d(C_4)$ $d(C_5)$	$g_4(x)$ $g_5(x)$
A	Q→A	0	3	31	$\mathbb{Z}_{20}$	31 31	4 3	$1+2x+x^2+x^3+2x^4+x^6$ $1+4x+2x^2+2x^3+x^4$
B	I→Q	18	3	31	$\mathbb{Z}_{20}$	31 31	4 3	$1+2x^2+x^3+x^4+2x^5+x^6$ $1+2x^2+x^3+x^4$
C	G→A	13	3	93	$\mathbb{Z}_{20}$	93 93	3 3	$3+x+x^3+3x^4+x^5+x^6+x^7$ $4+3x+2x^3+3x^4+x^5$
D	L→T	6	3	93	$\mathbb{F}_4\times\mathbb{Z}_5$	93 93	3 3	$\alpha+x+\alpha x^2+\beta x^3+\beta x^4+\beta x^5+x^6$ $4+x^2+3x^3+2x^4+x^5$
E	R→C	42	3	93	$\mathbb{F}_4\times\mathbb{Z}_5$	93 93	3 3	$\alpha+\alpha x+\alpha x^3+\alpha x^4+x^5+x^6$ $4+2x^3+x^4+x^5$
F	K→R	53	3	63	$\mathbb{F}_4\times\mathbb{Z}_5$	63 63	4 3	$\alpha+\beta x^3+x^4$ $4+4x^2+2x^3+3x^4+x^5+x^7$
G	L→A	13	3	31	$\mathbb{F}_4\times\mathbb{Z}_5$	31 31	3 4	$1+x+x^2+x^6$ $1+x+4x^2+3x^3+x^4$
H	N/A	-	6	78	$\mathbb{Z}_{20}$	39 78	3 6	$1+x+3x^2+3x^3+2x^6+x^7+3x^8+x^9+3x^{10}+2x^{11}+x^{12}$ $1+4x+x^2+4x^4+2x^5+4x^6+3x^7+4x^8+4x^9+4x^{10}+$ $3x^{11}+3x^{12}+2x^{13}+4x^{14}+4x^{15}+x^{16}$
I	V→A	61	3	84	$\mathbb{Z}_{20}$	21 84	3 3	$1+2x+3x^2+3x^4+x^5+x^6$ $4+2x+x^2+3x^3+3x^4+4x^6+x^7+4x^9+4x^{11}+x^{12}$

6.4 Caso de estudo: *Cytochrome b6-f complex subunit 6-OS*

Os algoritmos $BCH\_OneProt\_\mathbb{Z}_{20}$ e $BCH\_OneProt\_\mathbb{F}_4\times\mathbb{Z}_5$ foram testados nos bancos de dados: UniprotKB, NCBI e PDB, e assim, estruturas matemáticas relacionadas com códigos corretores de erros foram identificadas para um conjunto de **544** proteínas. Isto verifica a existência de códigos corretores de erros no processo de síntese de proteínas que agem de acordo com os sistemas de comunicação digital tradicionais.

Com o objetivo de tentar explicar os resultados decorridos dos algoritmos propostos neste capítulo e verificar sua aplicabilidade como uma ferramenta útil para a biologia, estudou-se uma classe específica de proteínas as quais estão presente em diferentes organismos: ***Cytochrome b6-f complex subunit 6-OS***, e comparou-se os resultados dos algoritmos com fatos biológicos conhecidos: agrupamento taxonômico e reconstrução filogenética.

6.4.1 Sobre as sequências

O primeiro passo, antes de analisar as sequências com os algoritmos apresentados em seções anteriores, é verificar se o comprimento das sequências satisfaz o critério de comprimento BCH para $\mathbb{Z}_4$, $\mathbb{F}_4$ e $\mathbb{Z}_5$, e assim, poder construir códigos CTR que possam identificar essas proteínas. As proteínas *Cytochrome b6-f complex subunit 6-OS* são de comprimento 31, e o critério do comprimento é satisfeito.

As sequências de aminoácidos foram obtidas de banco de dados públicos: National Center for Biotechnology Information (NCBI) e Ensembl (FLICEK *et al.*, 2014). Através deste processo, e para evitar erros de alinhamento e problemas associados com sequências incompletas, foram consideradas somente espécies para as quais as informações: sequência de aminoácidos e nucleotídeos, estão disponíveis. Assim, suficientes alinhamentos foram gerados através da aplicação ClustalW sobre o programa BioEdit (HALL, 2001). As 128 sequências de aminoácidos escolhidas pertencem aos phylums tracheophyta, anthocerophyta, charophyta, magnoliophyta, haptophyta, marchantiophyta, ochrophyta, chriptophyta, heterokontophyta e streptophyta (Ver Tabela 30). Ressalta-se que estas sequências representam ao redor de 500 milhões de anos atrás, quando as primeiras plantas vasculares e não vasculares apareceram. Uma cianobactéria (*anabaena variabilis*) foi incluída como sequência fora do grupo e, com isso, discriminar melhor a árvore filogenética.

Tabela 30 – Sequências empregadas nesta Seção.

ID	Nome taxonômico	NCBI ID	Classificação Taxonômica				
AA	*Beta vulgaris*	sp|P46612	Plantae	Tracheophyta	Magnoliopsida	Caryophyllales	Amaranthaceae
AB	*Cucumis sativus*	sp|Q4VZJ9	Plantae	Tracheophyta	Magnoliopsida	Cucurbitales	Cucurbitaceae
AC	*Arabidopsis thaliana*	sp|P56776	Plantae	Tracheophyta	Magnoliopsida	Brassicales	Brassicaceae
AD	*Acorus calamus*	sp|Q3V517	Plantae	Tracheophyta	Magnoliopsida	Acorales	Acoraceae
AE	*Abies homolepis*	sp|Q5K3V1	Plantae	Tracheophyta	Pinopsida	Pinales	Pinaceae
AF	*Adiantum capillus-veneris*	sp|Q85FK4	Plantae	Tracheophyta	Polypodiopsida	Polypodiales	Pteridaceae
AG	*Amaranthus caudatus*	sp|Q5K3T3	Plantae	Tracheophyta	Magnoliopsida	Caryophyllales	Amaranthaceae
AH	*Amaranthus cruentus*	sp|Q5K3T2	Plantae	Tracheophyta	Magnoliopsida	Caryophyllales	Amaranthaceae
AI	*Aethionema grandiflorum*	sp|A4QJL7	Plantae	Tracheophyta	Magnoliopsida	Brassicales	Brassicaceae
AJ	*Aethionema cordifolium*	sp|A4QJD3	Plantae	Tracheophyta	Magnoliopsida	Brassicales	Brassicaceae
AK	*Angiopteris evecta*	sp|A2T351	Plantae	Tracheophyta	Polypodiopsida	Marattiales	Marattiaceae
AL	*Anthoceros formosae*	sp|Q85AP7	Plantae	Anthocerotophyta	Anthocerotopsida	Anthocerotales	Anthocerotaceae
AM	*Atropa belladonna*	sp|P69402	Plantae	Tracheophyta	Magnoliopsida	Solanales	Solanaceae
AN	*Amborella trichopoda*	sp|Q70XY8	Plantae	Tracheophyta	Magnoliopsida	Amborellales	Amborellaceae
AO	*Agrostis stolonifera*	sp|A1EA26	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
AP	*Arabis hirsuta*	sp|A4QK36	Plantae	Tracheophyta	Magnoliopsida	Brassicales	Brassicaceae
AQ	*Buxus microphylla*	sp|A6MM54	Plantae	Tracheophyta	Magnoliopsida	Buxales	Buxaceae
AR	*Calycanthus floridus var. glaucus*	sp|Q7YJV7	Plantae	Tracheophyta	Magnoliopsida	Laurales	Calycanthaceae
AS	*Bistorta officinalis*	sp|Q5K3T8	Plantae	Tracheophyta	Magnoliopsida	Caryophyllales	Polygonaceae
AT	*Acorus gramineus*	sp|Q5QA80	Plantae	Tracheophyta	Magnoliopsida	Acorales	Acoraceae
AU	*Barbarea verna*	sp|A4QKC3	Plantae	Tracheophyta	Magnoliopsida	Brassicales	Brassicacea
AV	*Chenopodium bonus-henricus*	sp|Q5K3S8	Plantae	Tracheophyta	Magnoliopsida	Caryophyllales	Amaranthaceae
AW	*Chaetosphaeridium globosum*	sp|Q8M9Y3	Plantae	Charophyta	Coleochaetophyceae	Chaetosphaeridiales	Chaetosphaeridiaceae
AX	*Chara vulgaris*	sp|Q1ACI4	Plantae	Charophyta	Charophyceae	Charales	Characeae
AY	*Chloranthus spicatus*	sp|A6MME0	Plantae	Tracheophyta	Magnoliopsida	Chloranthales	Chloranthaceae
AZ	*Cuscuta gronovii*	sp|A7M915	Plantae	Tracheophyta	Magnoliopsida	Solanales	Convolvulaceae
BA	*Coffea arabica*	sp|A0A353	Plantae	Tracheophyta	Magnoliopsida	Gentianales	Rubiaceae
BB	*Chlorokybus atmophyticus*	sp|Q19V67	Plantae	Charophyta	Klebsormidiophyceae	Klebsormidiales	Klebsomidiaceae
BC	*Cuscuta reflexa*	sp|A7M981	Plantae	Tracheophyta	Magnoliopsida	Solanales	Convolvulaceae
BD	*Calycanthus floridus*	sp|Q5K3U6	Plantae	Tracheophyta	Magnoliopsida	Laurales	Calycanthaceae
BE	*Cycas taitungensis*	sp|A6H5J7	Plantae	Tracheophyta	Cycadopsida	Cycadales	Cycadaceae
BF	*Chlorella vulgaris*	sp|P56306	Plantae	Chlorophyta	Trebouxiophyceae	Chlorellales	Oocystaceae
BG	*Crucihimalaya wallichii*	sp|A4QKU9	Plantae	Tracheophyta	Magnoliopsida	Brassicales	Brassicaceae
BH	*Dioscorea elephantipes*	sp|A6MMM5	Plantae	Tracheophyta	Magnoliopsida	Dioscoreales	Dioscoreaceae

Continua na seguinte página

Tabela 30 – *Continuação da página anterior*

ID	Nome taxonômico	NCBI ID	Classificação Taxonômica				
BI	*Daucus carota*	sp|Q0G9U4	Plantae	Tracheophyta	Magnoliopsida	Apiales	Apiaceae
BJ	*Ficus carica*	sp|Q5K3R9	Plantae	Tracheophyta	Magnoliopsida	Rosales	Moraceae
BK	*Eucalyptus globulus subsp. globulus*	sp|Q49KY1	Plantae	Tracheophyta	Magnoliopsida	Myrtales	Myrtaceae
BL	*Gnetum parvifolium*	sp|A6BM21	Plantae	Tracheophyta	Gnetopsida	Ephedrales	Gnetaceae
BM	*Drimys granadensis*	sp|Q06GX9	Plantae	Tracheophyta	Magnoliopsida	Canellales	Winteraceae
BN	*Hamamelis virginiana*	sp|Q5K3S7	Plantae	Tracheophyta	Magnoliopsida	Saxifragales	Hamamelidaceae
BO	*Helianthus annuus*	sp|Q1KXU1	Plantae	Tracheophyta	Magnoliopsida	Asterales	Asteraceae
BP	*Gossypium barbadense*	sp|A0ZZ53	Plantae	Tracheophyta	Magnoliopsida	Malvales	Malvaceae
BQ	*Emiliania huxleyi*	sp|Q4G3B0	Chromalveolata	Haptophyta	Primnesiophyceae	Isochrysidales	Noelaerhabdaceae
BR	*Jasminum nudiflorum*	sp|Q06RB4	Plantae	Tracheophyta	Magnoliopsida	Lamiales	Oleaceae
BS	*Citrus sinensis*	sp|Q09MG0	Plantae	Tracheophyta	Magnoliopsida	Sapindales	Rutaceae
BT	*Lactuca sativa*	sp|Q332W0	Plantae	Tracheophyta	Magnoliopsida	Asterales	Asteraceae
BU	*Draba nemorosa*	sp|A4QL37	Plantae	Tracheophyta	Magnoliopsida	Brassicales	Brassicaceae
BV	*Illicium oligandrum*	sp|A6MMW2	Plantae	Tracheophyta	Magnoliopsida	Austrobaileyales	Schisandraceae
BW	*Lepidium virginicum*	sp|A4QLC4	Plantae	Tracheophyta	Magnoliopsida	Brassicales	Brassicaceae
BX	*Humulus lupulus*	sp|Q5K3S0	Plantae	Tracheophyta	Magnoliopsida	Rosales	Cannabaceae
BY	*Lobularia maritima*	sp|A4QLL2	Plantae	Tracheophyta	Magnoliopsida	Brassicales	Brassicacea
BZ	*Hordeum vulgare*	sp|A1E9K8	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
CA	*Magnolia stellata*	sp|Q5K3U3	Plantae	Tracheophyta	Magnoliopsida	Magnoliales	Magnoliaceae
CB	*Liriodendron tulipifera*	sp|Q0G9K1	Plantae	Tracheophyta	Magnoliopsida	Magnoliales	Magnoliaceae
CC	*Zea mays*	sp|P19445	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
CD	*Capsella bursa-pastoris*	sp|A4QKL0	Plantae	Tracheophyta	Magnoliopsida	Brassicales	Brassicacea
CE	*Nandina domestica*	sp|Q09FU3	Plantae	Tracheophyta	Magnoliopsida	Ranunculales	Berberidaceae
CF	*Anabaena variabilis*	sp|Q3M4V0	Bacteria	Cyanobacteria	Cyanophyceae	Nostocales	Nostocaceae
CG	*Mesostigma viride*	sp|Q9MUN4	Plantae	Charophyta	Mesostigmatophyceae	Mesostigmales	Mesostigmataceae
CH	*Marchantia polymorpha*	sp|P12179	Plantae	Marchantiophyta	Marchantiopsida	Marchantiales	Marchantiaceae
CI	*Morus indica*	sp|Q09WZ9	Plantae	Tracheophyta	Magnoliopsida	Rosales	Moraceae
CJ	*Nasturtium officinale*	sp|A4QLV1	Plantae	Tracheophyta	Magnoliopsida	Brassicales	Brassicacea
CK	*Nicotiana tomentosiformis*	sp|Q33C14	Plantae	Tracheophyta	Magnoliopsida	Solanales	Solanaceae
CL	*Nephroselmis olivacea*	sp|Q9TKY9	Plantae	Chlorophyta	Prasinophyceae	Psedoscourfieldiales	Pycnococcaceae
CM	*Nicotiana glutinosa*	sp|Q2UVE0	Plantae	Tracheophyta	Magnoliopsida	Solanales	Solanaceae
CN	*Odontella sinensis*	sp|P49524	Chromista	Ochrophyta	Bacillariophyceae	Triceratiales	Triceratiaceae
CO	*Oltmannsiellopsis viridis*	sp|Q20EX5	Plantae	Chlorophyta	Ulvophyceae	Oltmannsiellopsidales	Oltmannsidaceae
CP	*Oenothera elata subsp. hookeri*	sp|Q9MTK4	Plantae	Tracheophyta	Magnoliopsida	Myrtales	Onagraceae
CQ	*Magnolia grandiflora*	sp|Q5K3U4	Plantae	Tracheophyta	Magnoliopsida	Magnoliales	Magnoliaceae

Continua na seguinte página

Tabela 30 – *Continuação da página anterior*

ID	Nome taxonômico	NCBI ID	Classificação Taxonômica				
CR	*Oryza sativa*	sp|P0C393	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
CS	*Lotus japonicus*	sp|Q9BBR4	Plantae	Tracheophyta	Magnoliopsida	Fabales	Fabaceae
CT	*Huperzia lucidula*	sp|Q5SD33	Plantae	Tracheophyta	Lycopodiopsida	Lycopodiales	Lycopodiaceae
CU	*Panax ginseng*	sp|Q68RY8	Plantae	Tracheophyta	Magnoliopsida	Apiales	Araliaceae
CV	*Oryza sativa subsp. japonica*	sp|P12180	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
CW	*Phaeodactylum tricornutum*	sp|A0T0A1	Chromista	Ochrophyta	Bacillariophyceae	Naviculales	Phaeodactylaceae
CX	*Guillardia theta*	sp|O78468	Chromalveolata	Cryptophyta	Cryptophyaceae	Pyrenomonadales	Geminigeraceae
CY	*Gossypium hirsutum*	sp|Q2L924	Plantae	Tracheophyta	Magnoliopsida	Malvales	Malvaceae
CZ	*Nicotiana sylvestris*	sp|Q3C1L1	Plantae	Tracheophyta	Magnoliopsida	Solanales	Solanaceae
DA	*Phaseolus vulgaris*	sp|A4GGC4	Plantae	Tracheophyta	Magnoliopsida	Fabales	Fabaceae
DB	*Phytolacca americana*	sp|Q5K3T5	Plantae	Tracheophyta	Magnoliopsida	Caryophyllales	Phytolaccaceae
DC	*Oxybasis rubra*	sp|Q5K3S9	Plantae	Tracheophyta	Magnoliopsida	Caryophyllales	Amaranthaceae
DD	*Oryza sativa subsp. indica*	sp|P0C394	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
DE	*Oryza nivara*	sp|Q6ENF5	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
DF	*Phalaenopsis aphrodite subsp. formosana*	sp|Q3BAM2	Plantae	Tracheophyta	Magnoliopsida	Asparagales	Orchidaceae
DG	*Physcomitrella patens subsp. patens*	sp|Q6YXM0	Plantae	Bryophyta	Bryopsida	Funariales	Funariaceae
DH	*Populus deltoides*	sp|O20272	Plantae	Tracheophyta	Magnoliopsida	Malpighiales	Salicaceae
DI	*Populus trichocarpa*	sp|A4GYS9	Plantae	Tracheophyta	Magnoliopsida	Malpighiales	Salicaceae
DJ	*Piper cenocladum*	sp|Q06GP2	Plantae	Tracheophyta	Magnoliopsida	Piperales	Piperaceae
DK	*Populus alba*	sp|Q14FD9	Plantae	Tracheophyta	Magnoliopsida	Malpighiales	Salicaceae
DL	*Pseudendoclonium akinetum*	sp|Q3ZJ65	Plantae	Chlorophyta	Ulvophyceae	Ulvales	Kornmanniaceae
DM	*Platanus occidentalis*	sp|Q09G28	Plantae	Tracheophyta	Magnoliopsida	Proteales	Platanaceae
DN	*Porphyra purpurea*	sp|P51221	Plantae	Rhodophyta	Bangiophyceae	Bangiales	Bangiaceae
DO	*Potentilla anserina*	sp|Q5K3S1	Plantae	Tracheophyta	Magnoliopsida	Rosales	Rosaceae
DP	*Pyropia yezoensis*	sp|Q1XDR6	Plantae	Rhodophyta	Bangiophyceae	Bangiales	Bangiaceae
DQ	*Saccharum edule*	sp|Q4QYS8	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
DR	*Saccharum robustum*	sp|Q4QYT4	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
DS	*Saccharum sinense*	sp|Q4QYT1	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
DT	*Psilotum nudum*	sp|Q8WI03	Plantae	Monilophyta	Psilotopsida	Psilotales	Psilotaceae
DU	*Silene latifolia*	sp|Q06DU1	Plantae	Tracheophyta	Magnoliopsida	Caryophyllales	Caryophyllaceae
DV	*Saccharum officinarum*	sp|Q6ENU6	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
DW	*Nymphaea alba*	sp|Q6EW35	Plantae	Tracheophyta	Magnoliopsida	Nymphaeales	Nymphaeacea
DX	*Solanum tuberosum*	sp|Q2VEG0	Plantae	Tracheophyta	Magnoliopsida	Solanales	Solanaceae
DY	*Silene conica*	sp|Q06DU9	Plantae	Tracheophyta	Magnoliopsida	Caryophyllales	Caryophyllaceae
DZ	*Olimarabidopsis pumila*	sp|A4QJU9	Plantae	Tracheophyta	Magnoliopsida	Brassicales	Brassicaceae

Continua na seguinte página

Tabela 30 – *Continuação da página anterior*

ID	Nome taxonômico	NCBI ID	Classificação Taxonômica				
EA	*Staurastrum punctulatum*	sp|Q32RT4	Plantae	Charophyta	Zygnemophyceae	Desmidiales	Desmidiaceae
EB	*Saccharum barberi*	sp|Q4QYT2	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
EC	*Thalassiosira pseudonana*	sp|A0T0U4	Chromista	Ochrophyta	Bacillariophyceae	Thalassiosirales	Thalassiosiraceae
ED	*Sorghum bicolor*	sp|A1E9U2	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
EE	*Saccharum spontaneum*	sp|Q4QYS6	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
EF	*Vitis vinifera*	sp|Q5K3S2	Plantae	Tracheophyta	Magnoliopsida	Vitales	Vitaceae
EG	*Zygnema circumcarinatum*	sp|Q32RH7	Plantae	Charophyta	Zygnematophyceae	Zygnematales	Zygnemataceae
EH	*Welwitschia mirabilis*	sp|B2Y1X7	Plantae	Tracheophyta	Gnetopsida	Ephedrales	Welwitschiaceae
EI	*Spinacia oleracea*	sp|Q9M3L0	Plantae	Tracheophyta	Magnoliopsida	Caryophyllales	Amaranthaceae
EJ	*Solanum bulbocastanum*	sp|Q2MIG9	Plantae	Tracheophyta	Magnoliopsida	Solanales	Solanaceae
EK	*Solanum lycopersicum*	sp|Q2MI82	Plantae	Tracheophyta	Magnoliopsida	Solanales	Solanaceae
EL	*Glycine max*	sp|Q2PMR6	Plantae	Tracheophyta	Magnoliopsida	Fabales	Fabaceae
EM	*Rhodomonas salina*	sp|A6MVV7	Chromista	Cryptophycophyta	Cryptophyceae	Cryptomonadales	Cryptomonadaceae
EN	*Nicotiana tabacum*	sp|P69401	Plantae	Tracheophyta	Magnoliopsida	Solanales	Solanaceae
EO	*Triticum aestivum*	sp|P58247	Plantae	Tracheophyta	Magnoliopsida	Poales	Poaceae
EP	*Cyanidium caldarium*	sp|Q9TLR5	Plantae	Rhodophyta	Cyanidiophyceae	Cyanidiales	Cyanidiaceae
EQ	*Pelargonium hortorum*	sp|Q06FK8	Plantae	Tracheophyta	Magnoliopsida	Geraniales	Geraniaceae
ER	*Nostoc sp.*	sp|Q8YVQ2	Bacteria	Cyanobacteria	Cyanophyceae	Nostocales	Nostocaceae
ES	*Acorus calamus*	sp|Q3V541	Plantae	Tracheophyta	Magnoliopsida	Acorales	Acoraceae
ET	*Cicer arietinum*	sp|B5LMM3	Plantae	Tracheophyta	Magnoliopsida	Fabales	Fabaceae
EU	*Cuscuta gronovii*	sp|A7M8Z4	Plantae	Tracheophyta	Magnoliopsida	Solanales	Convolvulaceae
EV	*Cyanidium caldarium*	sp|Q9TLR6	Plantae	Rhodophyta	Cyanidiophyceae	Cyanidiales	Cyanidiaceae
EW	*Cuscuta obtusiflora*	sp|A8W3I0	Plantae	Tracheophyta	Magnoliopsida	Solanales	Convolvulaceae
EX	*Cyanidioschyzon merolae*	sp|Q85FX7	Plantae	Rhodophyta	Cyanidiophyceae	Cyanidiales	Cyanidiaceae

6.4.2 Análise filogenética

O Alinhamento final escolhido foi convertido à linguagem MEGA (TAMURA *et al.*, 2007) e três aproximações diferentes foram utilizadas para estimar a história evolutiva da reconstrução genética: o método algorítmico *Neighbor joining* (NJ), definido por Saitou e Nei (SAITOU; NEI, 1987); o algoritmo *Close-Neighbor-Interchange* (CNI) para encontrar as árvores de evolução mínima; e o algoritmo da máxima verosimilhança *Maximum Likelihood* (ML) (NEI; KUMAR, 2000). Para todos os casos, o desvio-padrão foi computado com 1000 réplicas de *Boostrap*.

Usando os métodos acima explicados, a reconstrução filogenética foi gerada com 118 sequências. Dez sequências do conjunto inicial de 128 proteínas foram removidas para evitar duplicação da informação (as sequências removidas mostraram identidades maiores que 99% entre elas).

6.4.3 Resultados e discussões sobre as proteínas *Cytochrome b6-f complex subunit 6-OS*

Foram analisadas 128 sequências através dos algoritmos: $BCH\_OneProt\_\mathbb{Z}_{20}$ e $BCH\_OneProt\_\mathbb{F}_4 \times \mathbb{Z}_5$, e os quatro casos de estudo (Taylor $\mathbb{F}_4 \times \mathbb{Z}_5$, Swanson $\mathbb{F}_4 \times \mathbb{Z}_5$, Taylor $\mathbb{Z}_{20}$ e Swanson $\mathbb{Z}_{20}$) foram considerados para assim determinar se essas sequências podem ser identificadas como palavras-código de um código cíclico corretor de erros com distância mínima de Hamming igual ou maior que três.

Dessas sequências, 31 foram identificadas por palavras-código que diferem em um único aminoácido e uma única posição com respeito as proteínas obtidas do NCBI (Tabela 31), isto é, todas as 31 sequências se encontram nas nuvens de sequências corregíveis, pois os códigos têm distância mínima de Hamming maior ou igual a três. Além disso, as 31 proteínas foram identificadas diferentemente entre os quatro diferentes casos de estudo (Figura 28). Estes fatos foram observados analogamente em trabalhos prévios quando analisadas as sequências de nucleotídeos, porém, a razão de sempre identificar sequências com uma única diferença com respeito a original permanece como um problema em aberto. Algumas abordagens tem sido realizadas para explicar esta característica; por exemplo, em (BRANDÃO *et al.*, 2015), os autores tentaram usar as sequências identificadas (com a mutação) para seguir a história evolutiva das proteínas. Sob outro ponto de vista, neste capítulo, conjectura-se que as sequências que pertencem aos mesmos códigos devem estar biologicamente relacionadas. As Tabelas 33, 34, 35 e 36 resumem todas as proteínas identificadas para cada caso de estudo. Nota-se que uma sequência pode ser identificada como palavra-código de dois ou mais códigos diferentes.

Os códigos CRT cíclicos sobre $\mathbb{Z}_{20} \simeq \mathbb{Z}_4 \times \mathbb{Z}_5$ e $\mathbb{F}_4 \times \mathbb{Z}_5$ serão construídos como a justaposição de dois códigos BCH, os quais são definidos completamente pelo polinômio

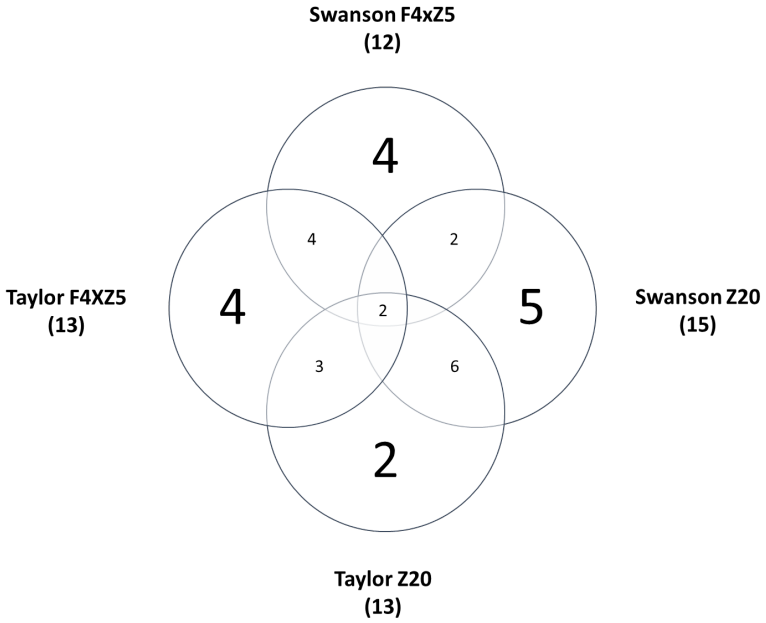


Figura 28 – Proteínas identificadas por todos os casos de estudo.

Tabela 31 – Sequência original de amino ácidos (Oaa) de cytochrome Zea mays e Sequências de aminoácidos de cytochrome Zea mays geradas pelo Código Corretor com uma única diferença (Gaa)

ID	Oaa:	M	L	T	I	T	S	Y	F	G	F	L	L	A	A	L	T	I	T	P	A	L	F	I	S	L	N	K	I	R	L	I
682	Gaa:	M	L	T	I	T	S	Y	F	G	F	L	L	A	A	L	T	I	T	P	A	L	F	I	S	L	N	K	I	R	**N**	I
683	Gaa:	M	L	T	I	T	S	Y	F	G	F	L	L	A	A	L	T	I	T	P	A	L	F	I	S	**G**	N	K	I	R	L	I
684	Gaa:	M	L	T	I	T	S	Y	F	G	**V**	L	L	A	A	L	T	I	T	P	A	L	F	I	S	L	N	K	I	R	L	I
685	Gaa:	**T**	L	T	I	T	S	Y	F	G	F	L	L	A	A	L	T	I	T	P	A	L	F	I	S	L	N	K	I	R	L	I
686	Gaa:	M	L	T	I	T	S	Y	F	G	F	L	L	A	A	L	T	I	T	P	A	L	F	I	S	L	**H**	K	I	R	L	I
687	Gaa:	M	L	T	I	T	S	Y	F	G	F	**G**	L	A	A	L	T	I	T	P	A	L	F	I	S	L	N	K	I	R	L	I
688	Gaa:	M	L	T	I	T	S	Y	F	G	F	L	L	A	A	L	T	I	T	P	A	L	F	I	S	L	N	K	I	R	**C**	I
689	Gaa:	M	L	T	I	**P**	S	Y	F	G	F	L	L	A	A	L	T	I	T	P	A	L	F	I	S	L	N	K	I	R	L	I
690	Gaa:	M	L	T	I	T	S	**S**	F	G	F	L	L	A	A	L	T	I	T	P	A	L	F	I	S	L	N	K	I	R	L	I
691	Gaa:	M	L	T	I	T	S	Y	F	G	F	L	L	A	A	L	T	I	T	P	A	L	F	I	S	L	N	**E**	I	R	L	I

gerador e os polinômios minimais que fatoram o referido polinômio gerador. Na Tabela 32, cada um dos polinômios utilizados no presente estudo são indexados com um número. De agora em diante, somente os índices dos polinômios minimais que fatoram os polinômios geradores para cada um dos códigos BCH serão mostrados.

O código $\mathcal{C} = CRT(\mathcal{C}_4, \mathcal{C}_5)$ é um código cíclico, um ideal e um subconjunto das possíveis sequências de aminoácidos, onde $\mathcal{C}_4$ e $\mathcal{C}_5$ são BCH, e portanto, $\mathcal{C}_4$ e $\mathcal{C}_5$ são ideais gerados pelos correspondentes polinômios geradores e formam subconjuntos de $(\mathbb{Z}_4)^n$ ou $(\mathbb{F}_4)^n$ e $(\mathbb{Z}_5)^n$, respectivamente. Dado que há um número finito de polinômios minimais, os quais fatoram os polinômios geradores, segue que há um número finito de códigos BCH que classificam as sequências de aminoácidos. Isto é, dadas duas sequências de tamanho n, elas são ditas biologicamente relacionadas se ambas sequências pertencem ao mesmo código.

As Equações 6.2 e 6.3 são propriedades de códigos cíclicos úteis para a classificação

Tabela 32 – Polinômios Minimais sobre $\mathbb{Z}_4$, $\mathbb{Z}_5$ e $\mathbb{F}_4$, usados na identificação de proteínas *Cythocrome b6-f complex subunit 6-OS*.

Índice	**Polinômio minimal**	**Alfabeto**
1	$3+x$	$\mathbb{Z}_4$
2	$4+x$	$\mathbb{Z}_5$
3	$3+2x+x^2+3x^3+x^4+x^5$	$\mathbb{Z}_4$
4	$3+3x+x^2+3x^3+2x^4+x^5$	$\mathbb{Z}_4$
5	$4+x+x^2+x^3$	$\mathbb{Z}_5$
6	$4+4x+4x^2+x^3$	$\mathbb{Z}_5$
7	$4+4x+2x^2+x^3$	$\mathbb{Z}_5$
8	$4+4x^2+x^3$	$\mathbb{Z}_5$
9	$4+x+x^3$	$\mathbb{Z}_5$
10	$3+2x+3x^2+x^5$	$\mathbb{Z}_4$
11	$3+x^3+2x^4+x^5$	$\mathbb{Z}_4$
12	$4+3x+x^2+x^3$	$\mathbb{Z}_5$
13	$3+x+3x^3+x^4+x^5$	$\mathbb{Z}_4$
14	$4+3x+4x^2+x^3$	$\mathbb{Z}_5$
15	$3+3x+x^2+3x^4+x^5$	$\mathbb{Z}_4$
16	$4+2x+x^3$	$\mathbb{Z}_5$
17	$4+3x^2+x^3$	$\mathbb{Z}_5$
18	$1+x$	$\mathbb{F}_4$
19	$1+x+x^2+x^4+x^5$	$\mathbb{F}_4$
20	$1+x+x^3+x^4+x^5$	$\mathbb{F}_4$
21	$1+x^2+x^3+x^4+x^5$	$\mathbb{F}_4$
22	$4+x+2x^2+x^3$	$\mathbb{Z}_5$
23	$1+x^2+x^5$	$\mathbb{F}_4$
24	$1+x^3+x^5$	$\mathbb{F}_4$

matemática de proteínas, onde as seguintes definições são necessárias: $\mathcal{C}_i = \langle f_i(x) \rangle$, $\mathcal{C}_j = \langle f_j(x) \rangle$, $\mathcal{C}_{ij} = \langle f_i(x) \cdot f_j(x) \rangle$, e com

$$\mathcal{C}_{ij} = \langle f_i(x) \cdot f_j(x) \rangle = \mathcal{C}_i \bigcap \mathcal{C}_j \tag{6.2}$$

$$\mathcal{C}_{ijk} \subset \mathcal{C}_{ij} \subset \mathcal{C}_i. \tag{6.3}$$

Para poder verificar a possibilidade que o algoritmo proposto detete relações biológicas entre sequências, primeiro, analisou-se se as sequências identificadas pertencem a códigos similares. A única maneira para que dois códigos sejam iguais é que tanto as componentes $\mathcal{C}_4$ como as componentes $\mathcal{C}_5$ tenham os mesmos polinômios minimais (MPs) sobre $\mathbb{Z}_4$ ou $\mathbb{F}_4$ e $\mathbb{Z}_5$. Como mostrado nas Tabelas 33, 34, 35 e 36 poucas coincidências são detetadas entre as proteínas identificadas quando se avalia por códigos iguais.

Considerando que duas sequências estão relacionadas matematicamente se elas possuem um MP comum (Equação 6.2), assim inicialmente, MPs foram analisados usando os casos para os quais mais de uma palavra-código foi derivada da mesma sequência de aminoácidos. Logo, se o algoritmo é capaz de detetar relações biológicas, então essas

sequências devem estar relacionadas matematicamente. Para todos os casos, os MPs $\mathbb{Z}_5$ são diferentes entre palavras-código derivadas da mesma sequência (Tabelas 33, 34, 35 e 36). Em contraste, uma maior quantidade de similaridades são observadas quando os MPs sobre $\mathbb{F}_4$ e $\mathbb{Z}_4$ para o código $\mathcal{C}_4$ são analisados. De fato, nota-se que quando o caso Taylor-$\mathbb{Z}_{20}$ é aplicado, os MPs sobre $\mathbb{Z}_4$ são comuns para as palavras-código derivadas da mesma sequência. Portanto, todos os próximos estudos somente consideram o caso Taylor-$\mathbb{Z}_{20}$. Lembre que os polinômios indexados como 1, 2 e 18, os quais correspondem ao polinômio $x-1$ para $\mathbb{Z}_4$, $\mathbb{Z}_5$ e $\mathbb{F}_4$, respectivamente, sempre aparecem quando o algoritmo é aplicado, pois o rotulamento com paridade zero sempre é escolhido, e portanto, as análises não os consideram.

Tabela 33 – Proteínas identificadas através do caso Taylor-$\mathbb{F}_4 \times \mathbb{Z}_5$.

prot_id	seq_id	filo_id	Organismo	pos_mut	Mut	PolsMinF4	PolsMinZ5
376	815	AL	*Anthoceros formosae*	20	L->R	[18, 20]	[2, 6]
376	817	AL	*Anthoceros formosae*	20	L->D	[18, 20]	[2, 14]
314	824	CC	*Zea mays*	0	M->R	[18, 21]	[2, 5]
314	825	CC	*Zea mays*	6	Y->M	[18, 20]	[2, 7]
342	886	EE	*Saccharum spontaneum*	14	L->K	[18, 24]	[2, 6]
342	887	EE	*Saccharum spontaneum*	28	R->Q	[18, 21]	[2, 22]
401	898	DF	*Phalaenopsis aphrodite subsp. Formosana*	7	F->I	[18, 19]	[2, 22]
401	899	DF	*Phalaenopsis aphrodite subsp. Formosana*	7	F->S	[18, 19]	[2, 6]
406	920	EA	*Staurastrum punctulatum*	28	Q->Y	[18, 20]	[2, 5]
406	921	EA	*Staurastrum punctulatum*	29	L->R	[18, 23]	[2, 5]
406	923	EA	*Staurastrum punctulatum*	28	Q->S	[18, 20]	[2, 12]
332	845	BZ	*Hordeum vulgare*	18	P->R	[18, 21]	[2, 14]
332	874	BZ	*Hordeum vulgare*	26	K->L	[18, 24]	[2, 12]
346	839	DA	*Phaseolus vulgaris*	7	F->S	[18, 23]	[2, 14]
352	915	DC	*Oxybasis rubra*	1	F->H	[18, 21]	[2, 8]
388	858	AX	*Chara vulgaris*	5	S->R	[18, 20]	[2, 17]
394	876	DK	*Populus alba*	14	L->D	[18, 21]	[2, 14]
398	885	BE	*Cycas taitungensis*	11	L->V	[18, 25]	[2, 5]
400	897	CU	*Panax ginseng*	30	I->Q	[18, 19]	[2, 22]
403	912	BO	*Helianthus annuus*	10	L->M	[18, 19]	[2, 7]

Tabela 34 – Proteínas identificadas através do caso Swanson-$\mathbb{F}_4 \times \mathbb{Z}_5$.

prot_id	seq_id	filo_id	Organismo	pos_mut	Mut	PolsMinF4	PolsMinZ5
331	926	DJ	*Piper cenocladum*	25	S->T	[18, 19]	[2, 12]
342	888	EE	*Saccharum spontaneum*	28	R->Q	[18, 21]	[2, 22]
346	838	DA	*Phaseolus vulgaris*	7	F->S	[18, 23]	[2, 14]
358	809	CB	*Liriodendron tulipifera*	16	I->N	[18, 23]	[2, 14]
376	816	AL	*Anthoceros formosae*	20	L->H	[18, 20]	[2, 6]
376	818	AL	*Anthoceros formosae*	20	L->D	[18, 20]	[2, 14]
388	859	AX	*Chara vulgaris*	5	S->H	[18, 20]	[2, 17]
393	875	BD	*Calycanthus floridus*	27	I->D	[18, 19]	[2, 6]
395	877	EL	*Glycine max*	1	L->W	[18, 23]	[2, 9]
398	884	BE	*Cycas taitungensis*	15	I->P	[18, 25]	[2, 12]
404	914	CT	*Huperzia lucidula*	25	N->M	[18, 23]	[2, 12]
406	919	EA	*Staurastrum punctulatum*	28	Q->Y	[18, 20]	[2, 5]
406	922	EA	*Staurastrum punctulatum*	28	Q->S	[18, 20]	[2, 12]
406	924	EA	*Staurastrum punctulatum*	29	L->H	[18, 23]	[2, 5]
407	925	BB	*Chlorokybus atmophyticus*	5	S->V	[18, 20]	[2, 12]

Dada a correspondência dos resultados do caso Taylor-$\mathbb{Z}_{20}$ entre palavras-código derivadas da mesma sequência, a seguir analisa-se se, baseado nos MPs obtidos, é possível

Tabela 35 – Proteínas identificadas através do caso Taylor-$\mathbb{Z}_{20}$.

prot_id	seq_id	filo_id	Organismo	pos_mut	Mut	PolsMinZ4	PolsMinZ5
312	679	CW	Phaeodactylum tricornutum	4	I->F	[1, 3]	[2, 9]
314	682	CC	*Zea mays*	29	L->N	[1, 11]	[2, 14]
314	683	CC	*Zea mays*	24	L->G	[1, 11]	[2, 6]
314	684	CC	*Zea mays*	9	F->V	[1, 11]	[2, 17]
314	685	CC	*Zea mays*	0	M->T	[1, 11]	[2, 5]
314	686	CC	*Zea mays*	25	N->H	[1, 11]	[2, 12]
314	687	CC	*Zea mays*	10	L->G	[1, 11]	[2, 16]
314	688	CC	*Zea mays*	29	L->C	[1, 11]	[2, 8]
314	689	CC	*Zea mays*	4	T->P	[1, 11]	[2, 22]
314	690	CC	*Zea mays*	6	Y->S	[1, 11]	[2, 7]
314	691	CC	*Zea mays*	26	K->E	[1, 11]	[2, 9]
317	696	AT	*Acorus gramineus*	1	P->S	[1, 3]	[2, 8]
318	698	AW	*Chaetosphaeridium globosum*	20	I->R	[1, 10]	[2, 7]
320	700	BC	*Cuscuta reflexa*	17	T->G	[1, 3]	[2, 16]
324	707	DP	*Pyropia yezoensis*	7	I->H	[1, 10]	[2, 22]
332	735	BZ	*Hordeum vulgare*	15	T->F	[1, 11]	[2, 6]
342	751	EE	*Saccharum spontaneum*	16	I->H	[1, 3]	[2, 12]
343	753	AK	*Angiopteris evecta*	19	V->N	[1, 10]	[2, 16]
346	756	DA	*Phaseolus vulgaris*	7	F->M	[1, 13]	[2, 14]
351	765	CI	*Morus indica*	23	G->M	[1, 4]	[2, 7]
352	766	DC	*Oxybasis rubra*	19	V->E	[1, 4]	[2, 17]
352	767	DC	*Oxybasis rubra*	19	V->I	[1, 4]	[2, 22]
355	772	CN	*Odontella sinensis*	14	F->T	[1, 15]	[2, 9]
355	774	CN	*Odontella sinensis*	14	F->W	[1, 15]	[2, 8]
355	776	CN	*Odontella sinensis*	14	F->M	[1, 15]	[2, 14]

identificar proteínas derivadas de organismos relacionados evolutivamente. Portanto, os resultados foram comparados com o agrupamento taxonômico das espécies e uma reconstrução filogenética da proteína *cytochrome b6-f complex subunit 6-OS*. A reconstrução filogenética foi desenvolvida sob condições de elevada restringência (Seção 6.4.2), a qual mostrou uma distribuição de táxons em 7 grupos parafiléticos, de acordo com a classificação taxonômica dos Reinos incluídos (Figuras 29 e 30). Adicionalmente, as arquiteturas das árvores obtidas por NJ e ML foram muito similares.

Estes resultados mostram que as sequências identificadas estão amplamente distribuídas em ambas árvores: filogenética e taxonômica, como mostram as Figuras 29 e 30. Além disso, padrões específicos não foram identificados, posto que, sequências com diferentes MPs estiveram presentes dentro dos grupos de sequências relacionadas nas árvores filogenética e taxonômica, e muitas sequências não foram nem identificadas pela metodologia Taylor-$\mathbb{Z}_{20}$.

Considerando os resultados anteriormente mencionados, uma análise filogenética adicional foi desenvolvida com a finalidade de identificar as relações potenciais entre as 13 sequências identificadas pelo caso de estudo Taylor-$\mathbb{Z}_{20}$. A Figura 31 mostra um grupo monofilético que inclui 8/11 sequências com um controle de replicação de 99%. Este resultado permite inferir uma alta similaridade entre estas sequências. Com base no fato de uma percentagem tão elevada na análise de réplica é incomum, mesmo em proteínas altamente conservadas evolutivamente, é notável que estas sequências possam ser identifi-

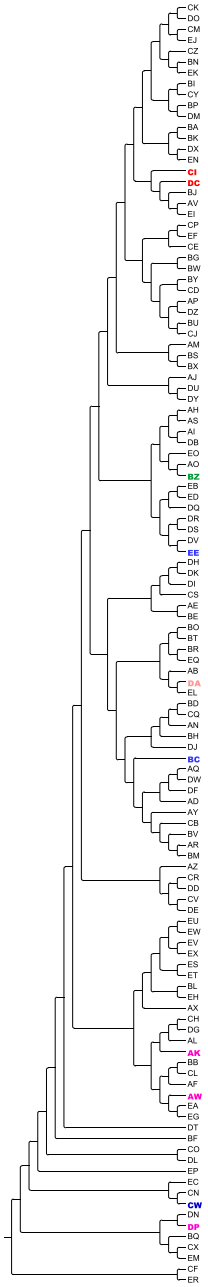


Figura 29 – Relações evolutivas entre táxons incluídos no estudo. As relações foram inferidas usando o método *neighbor-joining*, a percentagem de replicação foi calculada usando o parâmetro *boostrap* (ajustado em 1000). Proteínas identificadas pelo rotulamento Taylor e iguais polinômios minimais sobre $\mathbb{Z}_4$ são classificados por cores (ver Tabela 35). Táxons são identificados de acordo com o ID definido na Tabela 30.

Tabela 36 – Proteínas identificadas através do caso Swanson-$\mathbb{Z}_{20}$.

prot_id	seq_id	filo_id	Organismo	pos_mut	Mut	PolsMinZ4	PolsMinZ5
312	680	CW	Phaeodactylum tricornutum	4	I->F	[1, 3]	[2, 9]
316	693	DX	*Solanum tuberosum*	25	S->M	[1, 3]	[2, 12]
316	694	DX	*Solanum tuberosum*	26	K->F	[1, 13]	[2, 12]
316	695	DX	*Solanum tuberosum*	26	K->H	[1, 13]	[2, 8]
318	697	AW	*Chaetosphaeridium globosum*	20	I->H	[1, 10]	[2, 7]
324	706	DP	*Pyropia yezoensis*	7	I->R	[1, 10]	[2, 22]
331	725	DJ	*Piper cenocladum*	8	G->N	[1, 4]	[2, 22]
331	727	DJ	*Piper cenocladum*	10	L->C	[1, 4]	[2, 8]
331	728	DJ	*Piper cenocladum*	4	T->W	[1, 4]	[2, 6]
331	729	DJ	*Piper cenocladum*	25	S->E	[1, 4]	[2, 12]
331	730	DJ	*Piper cenocladum*	8	G->C	[1, 4]	[2, 16]
331	731	DJ	*Piper cenocladum*	0	M->P	[1, 4]	[2, 5]
331	732	DJ	*Piper cenocladum*	9	F->H	[1, 4]	[2, 14]
331	733	DJ	*Piper cenocladum*	10	L->G	[1, 4]	[2, 17]
331	734	DJ	*Piper cenocladum*	17	T->Q	[1, 4]	[2, 7]
334	738	EI	*Spinacia oleracea*	21	F->L	[1, 4]	[2, 6]
334	739	EI	*Spinacia oleracea*	21	F->R	[1, 4]	[2, 22]
336	742	EO	*Triticum aestivum*	16	I->F	[1, 10]	[2, 16]
342	750	EE	*Saccharum spontaneum*	21	F->I	[1, 10]	[2, 16]
343	752	AK	*Angiopteris evecta*	19	V->N	[1, 10]	[2, 16]
346	757	DA	*Phaseolus vulgaris*	7	F->M	[1, 13]	[2, 14]
347	758	BA	*Coffea arabica*	24	L->V	[1, 11]	[2, 17]
351	764	CI	*Morus indica*	12	A->G	[1, 15]	[2, 16]
355	771	CN	*Odontella sinensis*	14	F->W	[1, 15]	[2, 8]
355	773	CN	*Odontella sinensis*	14	F->T	[1, 15]	[2, 9]
355	775	CN	*Odontella sinensis*	14	F->M	[1, 15]	[2, 14]
358	781	CB	*Liriodendron tulipifera*	2	T->E	[1, 15]	[2, 14]
358	782	CB	*Liriodendron tulipifera*	24	L->E	[1, 10]	[2, 14]
363	726	DM	*Platanus occidentalis*	21	F->R	[1, 4]	[2, 9]

cadas e agrupadas pela metodologia matemática proposta. Além disso, esta similaridade não é obvia nem dentro da classificação taxonômica das espécies detetadas, nem dentro da árvore filogenética da Figura 29. A análise da sequência de acordo com o alinhamento desenvolvido por *Clustall W* revela identidade acima do 90% entre sequências (Figura 32).

6.5 Discussão e Considerações

Neste capítulo, a teoria de codificação foi aplicada pela primeira vez para na análise de sequências de aminoácidos e verificou-se a aplicabilidade do **Codificador Genético Concatenado**, introduzido neste capítulo, no contexto biológico. Durante o desenvolvimento do trabalho, uma metodologia e algoritmos, com sua respectiva justificação teórica, para a identificação de estruturas matemáticas que representam proteínas, no contexto da teoria da informação e codificação, através de códigos corretores de erros cíclicos (ECRT) foi introduzida. A metodologia para a identificação das sequências de aminoácidos foi determinada através de códigos cíclicos sobre $\mathbb{Z}_{20}$ ou $\mathbb{F}_4 \times \mathbb{Z}_5$, os quais são criados pela justaposição de dois códigos BCH não necessariamente tendo o mesmo comprimento (ECRT): um sobre $\mathbb{Z}_4$ ou $\mathbb{F}_4$ e outro sobre $\mathbb{Z}_5$. Esta metodologia é uma proposta original para

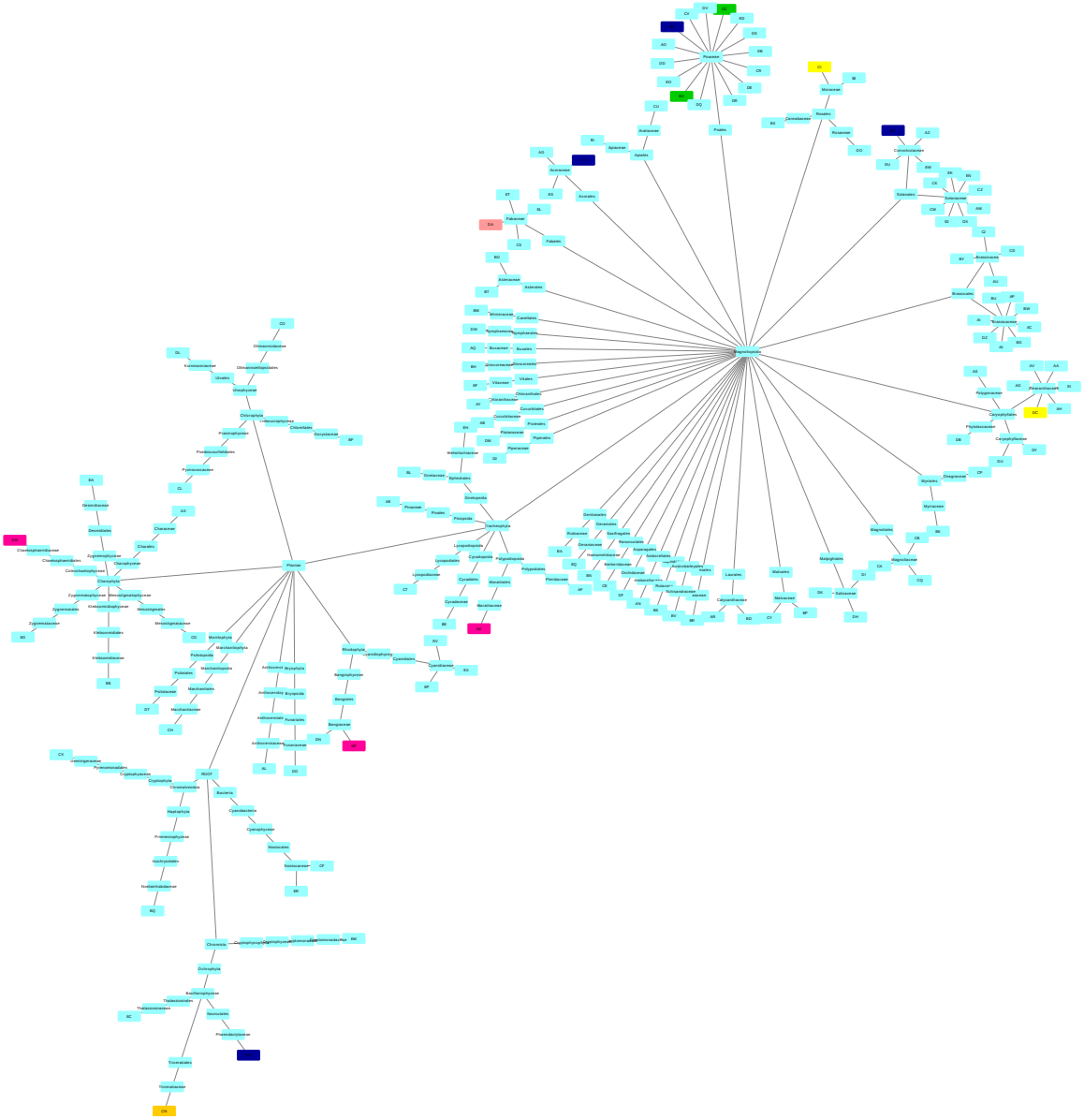


Figura 30 – Classificação taxonômica dos organismos da Tabela 30. Organismo tal que sua proteína *cytochrome b6-f complex subunit 6-OS* foi identificada através do caso de estudo Taylor-$\mathbb{Z}_{20}$ estão coloridos. Proteínas identificadas por iguais polinômios minimais sobre $\mathbb{Z}_4$ são classificados por cores (ver Tabela 35 e Figura 29). A ferramenta Cytoscape foi utilizada para construir esta figura (SHANNON *et al.*, 2003; ONO *et al.*, 2015)

analisar sequências de aminoácidos e para revelar sua estrutura matemática. Como visto na Seção 6.4, pela primeira vez, os resultados matemáticos foram comparados com dados biológicos para avaliar a aplicabilidade do uso da teoria de codificação no contexto biológico.

As relações biológicas entre as sequências analisadas foram estudadas para os quatro casos de estudo propostos, e assim, verificou-se que o caso de estudo Taylor-$\mathbb{Z}_{20}$ foi mais significativo em termos biológicos que os outros três casos de estudo: Taylor-$\mathbb{F}_4 \times \mathbb{Z}_5$,

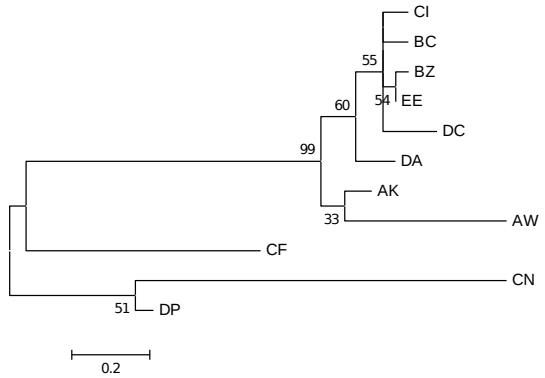


Figura 31 – Análises filogenética molecular usando o método Taylor-$\mathbb{Z}_{20}$ das proteínas *Cythocrome b6-f complex subunit 6-OS* identificadas pela metodologia Taylor-$\mathbb{Z}_{20}$.

```
                  10         20         30
AK          MLTLLSYFGF L-FAILTLTS VLFIGLNKIQ LI
AT          MPTITSYFGF LLAA-STITT ALFIGLSKIR LI
BC          MLTITSYFGF LVAAF-TITS ALFIGLNKIR LI
BZ          MLTLTSYFGF LLAA-LTITP ALFIGLNKIR LI
CC          MLTITSYFGF LLAA-LTITP ALFISLNKIR LI
CI          MSTITSYFGF LLAA-LTITS AIFIGLNKIR LI
DA          MLTITSYFGF LLA-VLIITS SLFIGLSKIQ LI
EE          MLTITSYFGF LLAA-LTITP ALFIGLNKIR LI
DC          MFTLTSYFGF LLAA-LTITF VLFIGLNKIR LI
DP          MSVFLGYIIF L-AAFFGLAT GLFLGLKAIK LI
CN          MTIAIDYFLL VGFCFAF-TS GLYLGLKSIK LI
```

Figura 32 – Alinhamento múltiplo das proteínas *Cythocrome b6-f complex subunit 6-OS* identificadas pela metodologia Taylor-$\mathbb{Z}_{20}$. O alinhamento foi gerado usando *Clusstall-W* com ajuste manual no programa *Bioedit*. Os táxons estão identificados de acordo com o ID definido na Tabela 30.

Swanson-$\mathbb{Z}_{20}$ e Swanson-$\mathbb{F}_4 \times \mathbb{Z}_5$. Para alcançar esta conclusão, as proteínas *Cythocrome b6-f complex subunit 6-OS* de diferentes organismos foi analisada através dos 4 casos de estudo propostos. Observou-se que os polinômios minimais sobre $\mathbb{Z}_4$ são compartilhados entre as proteínas mutadas derivadas da mesma proteína original.

Os códigos cíclicos construídos pelo uso de códigos BCH usados neste capítulo correspondem à construção mais simples e conhecida, e portanto, algumas considerações inerentes que não aplicam a proteínas foram assumidas; por exemplo, os códigos são cíclicos e na biologia não é sempre verdade que qualquer deslocamento cíclico de uma proteína funcional é também uma proteína funcional. Nesta direção, códigos mais ajustados ao contexto biológico devem ser construídos, porém, o projeto de códigos sobre os alfabetos $\mathbb{Z}_{20}$ e $\mathbb{F}_4 \times \mathbb{Z}_5$ implicam uma quantidade considerável de desafios para a teoria de codificação. Apesar das suposições, foi possível descobrir uma estrutura matemática para as proteínas identificadas e inclusive, para o caso das proteínas *Cythocrome b6-f complex subunit 6-OS*, algumas das sequências correspondentes a diferentes organismos foram agrupadas em um único grupo monofilético com um controle de replicação de 99%, não usual, inclusive para proteínas altamente conservadas evolutivamente.

Conclusões e Sugestões para Trabalhos Futuros

Este trabalho de Doutorado é o resultado da aplicação dos conceitos da teoria da informação e codificação para elucidar uma estrutura matemática nas estruturas biológicas (DNA, mRNA e proteínas) que armazenam a informação genética no contexto de sistemas digitais de comunicação. Dada a similaridade existente entre os sistemas tradicionais de comunicação digital e o processo de síntese de proteínas, neste trabalho: identificamos uma relação direta entre a estrutura das proteínas e os códigos corretores de erros; modelamos o Ribossomo e o processo de síntese de proteínas através de um **codificador genético concatenado**; definimos uma metodologia, com seus algoritmos respectivos, para a identificação de proteínas como palavras-código de dito codificador genético concatenado baseado em códigos BCH; e relacionamos 455 sequências de aminoácidos de um total de 4.787 sequências analisadas com alguma estrutura matemática no contexto da teoria de codificação e informação, do qual mostramos a existência de CCEs e estruturas matemáticas relacionadas com as estruturas das proteínas e verificamos que as proteínas são macromoléculas usadas na transmissão de informação.

Para levar a termo esta proposta de trabalho que mistura a engenharia com a biologia, a tese acaba tocando a disciplina da Teoria Algébrica. Além disso, sendo que o objetivo deste trabalho é aplicar a teoria da informação e codificação para elucidar e verificar hipóteses no contexto biológico. Durante o desenvolvimento deste trabalho, evidenciamos que a biologia também pode aportar na engenharia e na matemática, pelo fato que o problema da modelagem da síntese de proteínas levou a resolução de problemas novos para a engenharia e a matemática. Este é o caso da nova metodologia que introduzimos neste trabalho para o projeto de códigos **ECRT** sobre anéis comutativos com unidade, e dos algoritmos e demonstrações apresentamos para a identificação de sequências de nucleotídeos como palavras-código de códigos BCH, os quais generalizam e justificam fatos e afirmações descobertos em trabalhos anteriores neste grupo de pesquisa.

O resultado diferencial deste trabalho é a modelagem do processo de síntese de proteínas como um sistema de transmissão digital, onde o Ribossomo age como o modulador e as sequências mRNA como palavras-código de um codificador genético. Verificamos e introduzimos o ferramental teórico para validar o modelo através de casos de estudo específicos. Validamos o funcionamento do Ribossomo como modulador, de acordo com o sistema de transmissão tradicional, de duas maneiras: a primeira, verifica que as sequências mRNA são identificadas como palavras-código de um CCE e que o Ribossomo, simplesmente, age como um mapa sobrejetor de códons para aminoácidos, e a segunda, valida

que o codificador genético pode ser fusionado com o Ribossomo para formar um único **codificador genético concatenado** e modelado como um único CCE, para o qual as proteínas são identificadas como palavras-código.

Desenvolvimento do Trabalho

O Capítulo 2 é introdutório, onde se apresenta de forma sucinta alguns conceitos sobre a biologia celular e molecular com ênfase na síntese de proteínas e as principais definições e fatos referentes à teoria de códigos corretores de erros.

Os Capítulos 3, 4, 5 e 6 contêm as contribuições deste trabalho. No Capítulo 3 introduzimos e detalhamos um algoritmo para encontrar o maior código BCH com a maior distância de projeto BCH (incluindo a subclasse nsBCH que tem sido utilizada em trabalhos recentes) tal que uma sequência, dada como entrada ao algoritmo, é uma palavra-código desse código BCH. O algoritmo foi usado ao longo do trabalho na identificação de sequências mRNA e proteínas como palavras-código de códigos BCH.

No Capítulo 4, validamos o modelo tal que o Ribossomo age como um mapa sobrejetor de códons para aminoácidos onde as sequências mRNA são palavras-código de um código BCH. Usamos o algoritmo do Capítulo 3 para encontrar a estrutura matemática de sequências mRNA, no contexto da teoria da informação e codificação, como palavras-código de códigos BCH. Neste capítulo, seguimos a metodologia introduzida em (FARIA, 2011; ROCHA, 2010) e demonstramos alguns fatos e perguntas em aberto que foram propostas nesses trabalhos. Além disso, mostramos que a metodologia para a identificação, junto com o algoritmo do Capítulo 3, generaliza o algoritmo usado em (FARIA, 2011; ROCHA, 2010; BRANDÃO *et al.*, 2015; FARIA *et al.*, 2012; ROCHA *et al.*, 2010; FARIA *et al.*, 2010) para a identificação de sequências de nucleotídeos (mRNA, DNA, miRNA, etc), portanto a metodologia proposta neste capítulo permite identificar ou representar uma maior quantidade de sequências biológicas que as identificadas nos trabalhos mencionados.

No Capítulo 5, definimos os novos códigos **ECRT** e demonstramos todas as propriedades derivadas das definições (parâmetros do códigos, uma base ou conjunto de geradores,etc). Estes códigos podem ser utilizados para o projeto de códigos cíclicos sobre anéis comutativos com unidade. Porém, a cardinalidade do código, quando os comprimentos são diferentes, é muito pequena; o qual torna difícil encontrar proteínas que sejam palavras-código dos códigos mencionados.

No Capítulo 6, modelamos o Ribossomo e o processo de síntese de proteínas através de um **codificador genético concatenado** e validamos o modelo através de casos de estudo. Os resultados obtidos mostram que as proteínas são macromoléculas que contêm informação e são usadas pela célula na transmissão de informação. O agrupamento sugerido pelas propriedades matemáticas dos CCEs permitiu relacionar algumas

das sequências correspondentes a diferentes organismos em um único grupo monofilético com um controle de replicação de 99%, o qual é inusual, inclusive para proteínas altamente conservadas evolutivamente.

Sugestões para Trabalhos Futuros

Dados os resultados, há uma série de trabalhos futuros possíveis de se realizar. Listam-se a seguir alguns deles.

- De maneira similar a outros trabalhos relacionados, neste trabalho observamos que a proteína original sempre difere em único aminoácido com respeito à proteína inferida pelo CCE. O fato de nunca encontrar um CCE que represente exatamente a sequência original, tanto para mRNA como para proteínas, continua sendo um problema em aberto. Ainda mais complexo é identificar o significado biológico da diferença.

- Os códigos cíclicos construídos pelo uso de códigos BCH usados neste trabalho correspondem à construção mais simples e conhecida, portanto, assumimos algumas considerações inerentes que não aplicam às proteínas; por exemplo, os códigos são cíclicos e na biologia não é sempre verdade que qualquer deslocamento cíclico de uma proteína funcional é também uma proteína funcional. Nesta direção, códigos mais ajustados ao contexto biológico devem ser construídos. O projeto de códigos sobre os alfabetos $\mathbb{Z}_{20}$ e $\mathbb{F}_4 \times \mathbb{Z}_5$ implicam uma quantidade considerável de desafios para a teoria de codificação.

- O comprimento dos códigos BCH para os alfabetos $\mathbb{Z}_4$ e $\mathbb{F}_4$ deve pertencer ao seguinte conjunto:

  $$\{\text{divisores de: } 2^s - 1 \ \mid s = 1, 2, \dots\}.$$

  Na natureza, as sequências de nucleotídeos são encontradas para um conjunto muito mais variado de comprimentos, portanto, deve-se fundamentar a teoria matemática para tornar possível o projeto de algoritmos que sejam capazes de associar CCEs com sequências de nucleotídeos com comprimento variados.

- O mesmo problema anterior ocorre para as proteínas; onde o comprimento dos códigos BCH para os alfabetos $\mathbb{Z}_{20}$ e $\mathbb{F}_4 \times \mathbb{Z}_5$ deve pertencer ao seguinte conjunto:

  $$\{\text{divisores de: } 2^s - 1 \ \mid s = 1, 2, \dots\} \cap \{\text{divisores de: } 5^s - 1 \ \mid s = 1, 2, \dots\}$$

  Apesar que introduzimos e projetamos os códigos ECRT a partir desta necessidade, a cardinalidade do código, quando os comprimentos são diferentes, é muito pequena, o que torna difícil encontrar proteínas que sejam identificadas como palavras-código dos códigos ECRT.

- No Capítulo 6, utilizamos as análises filogenética e taxonômica como ajuda na interpretação dos resultados inferidos pelos CCEs. Ainda assim, análises biológicas mais profundas e detalhadas devem ser realizadas para poder interpretar os resultados apontados pelos CCEs.
  Em geral, os resultados indicados pelos CCEs para sequências de nucleotídeos e aminoácidos devem ser interpretados para que os CCEs possam se tornar ferramentas aplicáveis na inferência de fatos biológicos.

- A modelagem do processo de síntese de proteínas como um codificador genético concatenado abre as portas para o uso de códigos concatenados, como por exemplo, os códigos multi-nível. Este tipo de códigos permitem modelar a proteção desigual que existe nas sequências mRNA.

- O algoritmo que apresentamos no Capítulo 3 deve ser melhorado em termos de tempo computacional. Apesar de não ter sido feita uma análise de complexidade, pois não era o foco do trabalho, percebemos que a complexidade não era linear com respeito a n, o comprimento da sequência de nucleotídeos ou aminoácidos.

Referências

BARBIERI, M. *Introduction to Biosemiotics: The New Biological Synthesis*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2008. ISBN 1402083440, 9781402083440.

BATTAIL, G. Information Theory and Error-Correcting Codes In Genetics and Biological Evolution. In: _____. *Introduction to Biosemiotics: The New Biological Synthesis*. Dordrecht: Springer Netherlands, 2007. p. 299–345. ISBN 978-1-4020-4814-2. Disponível em: <http://dx.doi.org/10.1007/1-4020-4814-9_13>.

BATTAIL, G. *Information Theory and Error-Correcting Codes In Genetics and Biological Evolution*. Dordrecht: Springer Netherlands, 2007. 299–345 p. ISBN 978-1-4020-4814-2. Disponível em: <http://dx.doi.org/10.1007/1-4020-4814-9_13>.

BERG, J.; TYMOCZKO, J.; STRYER, L. *Bioquímica*. Guanabara Koogan, 2003. ISBN 9788527708722. Disponível em: <http://books.google.com.br/books?id=aukrPwAACAAJ>.

BINI, G.; FLAMINI, F. *Finite Commutative Rings and Their Applications*. 3Island Press, 2002. ISBN 9781461509585. Disponível em: <https://books.google.com.co/books?id=eaH3sgEACAAJ>.

BINI, G.; FLAMINI, F.; JUNGNICKEL, D. *Finite commutative rings and their applications*. Boston: Kluwer Academic Publ., 2002. (Kluwer international series in engineering and computer science). ISBN 1-402-07039-X. Disponível em: <http://opac.inria.fr/record=b1105884>.

BLAKE, I. F. Codes over certain rings. *Information and Control*, v. 20, n. 4, p. 396–404, 1972. ISSN 0019-9958. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0019995872902239>.

BRANDÃO, M. M.; SPOLADORE, L.; FARIA, L. C. B.; ROCHA, A. S. L.; SILVA-FILHO, M. C.; Palazzo Jr, R. Ancient DNA sequence revealed by error-correcting codes. *Scientific Reports*, The Author(s) SN -, v. 5, p. 12051 EP–, Jul 2015. Article. Disponível em: <http://dx.doi.org/10.1038/srep12051>.

BURTON, H.; WELDON, E. J. Cyclic product codes. *Information Theory, IEEE Transactions on*, v. 11, n. 3, p. 433–439, Jul 1965. ISSN 0018-9448.

CAO, Z.; CAO, H. On Fast Division Algorithm for Polynomials Using Newton Iteration. In: LIU, B.; MA, M.; CHANG, J. (Ed.). *ICICA (LNCS)*. Springer, 2012. (Lecture Notes in Computer Science, v. 7473), p. 175–180. ISBN 978-3-642-34061-1. Disponível em: <http://dblp.uni-trier.de/db/conf/icica/icica2012.html#CaoC12;http://dx.doi.org/10.1007/978-3-642-34062-8_23;http://www.bibsonomy.org/bibtex/2f3c42d11bcb336dee3582b3d6c983869/dblp>.

CRICK, F. Central Dogma of Molecular Biology. *Nature*, Nature Publishing Group, v. 227, n. 5258, p. 561–563, ago. 1970. Disponível em: <http://dx.doi.org/10.1038/227561a0>.

DAHL, C. Fast decoding of codes from algebraic curves. *Information Theory, IEEE Transactions on*, v. 40, n. 1, p. 223–229, Jan 1994. ISSN 0018-9448.

DAYHOFF, M.; FOUNDATION, N. B. R. *Atlas of Protein Sequence and Structure*. National Biomedical Research Foundation., 1979. (Atlas of Protein Sequence and Structure, v. 5). ISSN 0572-435X. ISBN 9780912466071. Disponível em: <https://books.google.com.co/books?id=V4RFAQAAIAAJ>.

DEBATA, P. P.; MISHRA, D.; SHAW, K.; MISHRA, S. A Coding Theoretic Model for Error-detecting in DNA Sequences. *Procedia Engineering*, v. 38, p. 1773–1777, 2012. ISSN 1877-7058. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877705812021297>.

DÓNAILL, D. A. M. Why Nature Chose A, C, G and U/T: An Error-Coding Perspective of Nucleotide Alphabet Composition. *Origins of life and evolution of the biosphere*, v. 33, n. 4, p. 433–455, 2003. ISSN 1573-0875. Disponível em: <http://dx.doi.org/10.1023/A:1025715209867>.

DOUGHERTY, S. T.; HARADA, M.; SOLÉ, P. Self-dual codes over rings and the Chinese remainder theorem. *Hokkaido Math. J.*, Hokkaido University, Department of Mathematics, v. 28, n. 2, p. 253–283, 02 1999. Disponível em: <http://dx.doi.org/10.14492/hokmj/1351001213>.

DOUGHERTY, S. T.; PARK, Y. H.; LIU, H. Lifted Codes over Finite Chain Rings. *Mathematical Journal of Okayama University*, v. 53, n. 3, p. 39–53, 2011. ISSN 0030-1566.

DOUGHERTY, S. T.; SHIROMOTO, K. MDR codes over $\mathbb{Z}_k$. *Information Theory, IEEE Transactions on*, v. 46, n. 1, p. 265–269, Jan 2000. ISSN 0018-9448.

FARIA, L. *Existence of error-correcting codes and communication protocols in DNA sequences*. Tese (Doutorado) — University of Campinas, 2011.

FARIA, L.; ROCHA, A.; KLEINSCHMIDT, J.; Palazzo Jr, R.; SILVA-FILHO, M. DNA sequences generated by BCH codes over GF(4). *Electronics Letters*, v. 46, n. 3, p. 203–204, Feb 2010. ISSN 0013-5194.

FARIA, L.; ROCHA, A.; Palazzo Jr, R. Transmission of intra-cellular genetic information: A system proposal. *Journal of Theoretical Biology*, v. 358, n. 0, p. 208–231, 2014. ISSN 0022-5193. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0022519314003233>.

FARIA, L. C. B.; L., R. A. S.; H., K. J.; C., S.-F. M.; EDSON, B.; H., H. R.; B., Y. M. E.; REGINALDO, P. J. Is a Genome a Codeword of an Error-Correcting Code? *PLoS ONE*, Public Library of Science, v. 7, n. 5, p. 1–9, 05 2012. Disponível em: <http://dx.doi.org/10.1371%2Fjournal.pone.0036644>.

FLICEK, P.; AMODE, M. R.; BARRELL, D.; BEAL, K.; BILLIS, K.; BRENT, S.; CARVALHO-SILVA, D.; CLAPHAM, P.; COATES, G.; FITZGERALD, S.; GIL, L.; GIRÓN, C. G.; GORDON, L.; HOURLIER, T.; HUNT, S.; JOHNSON, N.; JUETTEMANN, T.; KÄHÄRI, A. K.; KEENAN, S.; KULESHA, E.; MARTIN, F. J.; MAUREL, T.; MCLAREN, W. M.; MURPHY, D. N.; NAG, R.; OVERDUIN, B.; PIGNATELLI, M.; PRITCHARD, B.; PRITCHARD, E.; RIAT, H. S.; RUFFIER,

M.; SHEPPARD, D.; TAYLOR, K.; THORMANN, A.; TREVANION, S. J.; VULLO, A.; WILDER, S. P.; WILSON, M.; ZADISSA, A.; AKEN, B. L.; BIRNEY, E.; CUNNINGHAM, F.; HARROW, J.; HERRERO, J.; HUBBARD, T. J.; KINSELLA, R.; MUFFATO, M.; PARKER, A.; SPUDICH, G.; YATES, A.; ZERBINO, D. R.; SEARLE, S. M. Ensembl 2014. *Nucleic Acids Research*, v. 42, n. D1, p. D749–D755, 2014.

FORNEY, G. D. *Concatenated codes*. [S.l.]: Citeseer, 1966. v. 11.

FORSDYKE, D. Conservation of Stem-Loop Potential in Introns of Snake Venom Phospholipase A2 Genes: An Application of FORS-D Analysis. *Molecular Biology and Evolution*, v. 12, n. 6, p. 1157, 1995. Disponível em: <http://mbe.oxfordjournals.org/content/12/6/1157.short>.

FORSDYKE, D. R. Are introns in-series error-detecting sequences? *Journal of Theoretical Biology*, v. 93, n. 4, p. 861–866, 1981. ISSN 0022-5193. Disponível em: <http://www.sciencedirect.com/science/article/pii/0022519381903441>.

GEER, L. Y.; MARCHLER-BAUER, A.; GEER, R. C.; HAN, L.; HE, J.; HE, S.; LIU, C.; SHI, W.; BRYANT, S. H. The NCBI BioSystems database. *Nucleic Acids Res.*, v. 38, n. Database issue, p. D492–496, Jan 2010.

GOLAY, M. Notes on digital coding. *Proc. IEEE*, v. 37, p. 657, 1949.

GUENDA, K.; GULLIVER, T. A. MDS and self-dual codes over rings. *Finite Fields and Their Applications*, v. 18, n. 6, p. 1061–1075, 2012. ISSN 1071-5797. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1071579712000822>.

HALL, T. *BioEdit: a user-friendly biological sequence alignment editor and analysis program for Windows 95/98/NT*. [S.l.]: Nucl. Acids. Symp. Ser. 41:95-98., 2001.

HAMMING, R. W. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, Blackwell Publishing Ltd, v. 29, n. 2, p. 147–160, 1950. ISSN 1538-7305. Disponível em: <http://dx.doi.org/10.1002/j.1538-7305.1950.tb00463.x>.

HASHEM, Y.; GEORGES, A. des; FU, J.; BUSS, S. N.; JOSSINET, F.; JOBE, A.; ZHANG, Q.; LIAO, H. Y.; GRASSUCCI, R. A.; BAJAJ, C.; WESTHOF, E.; MADISON-ANTENUCCI, S.; FRANK, J. High-resolution cryo-electron microscopy structure of the trypanosoma brucei ribosome. *Nature*, Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved., v. 494, n. 7437, p. 385–389, Feb 2013. ISSN 0028-0836. Disponível em: <http://dx.doi.org/10.1038/nature11872>.

INTERLANDO, J.; Palazzo Jr, R.; ELIA, M. On the decoding of Reed-Solomon and BCH codes over integer residue rings. *Information Theory, IEEE Transactions on*, v. 43, n. 3, p. 1013–1021, May 1997. ISSN 0018-9448.

KANWAR, P.; LÓPEZ-PERMOUTH, S. R. Cyclic Codes over the Integers Modulo p^m. *Finite Fields and Their Applications*, v. 3, n. 4, p. 334–352, 1997. ISSN 1071-5797. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1071579797901891>.

LANG, S. *Algebra*. Menlo Park Cal: Addison-Wesley, 1993. ISBN 0-201-55540-9.

LIEBOVITCH, L.; TAO, Y.; TODOROV, A.; LEVINE, L. Is there an error correcting code in the base sequence in DNA? *Biophysical Journal*, v. 71, n. 3, p. 1539–1544, 1996. ISSN 0006-3495. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0006349596793566>.

LINT, J. van. *Introduction to Coding Theory*. New York University Press, 1999. (Graduate Texts in Mathematics). ISBN 9783540641339. Disponível em: <http://books.google.com.br/books?id=tvQhRUFh7EwC>.

LODISH, H. *Molecular Cell Biology*. W. H. Freeman, 2008. ISBN 9780716776017. Disponível em: <https://books.google.com.co/books?id=K3JbjG1JiUMC>.

LODISH, H. *Molecular Cell Biology*. W. H. Freeman, 2008. ISBN 9780716776017. Disponível em: <http://books.google.com.br/books?id=K3JbjG1JiUMC>.

MASSEY, J. Reversible Codes. *Information and Control*, v. 7, n. 3, p. 369–380, set. 1964. ISSN 00199958. Disponível em: <http://dx.doi.org/10.1016/s0019-9958(64)90438-3>.

MAY, E. *Comparative Analysis of Information Based Models for Initiating Protein Translation in Escherichia Coli K-12*. North Carolina State University., 1998. Disponível em: <http://books.google.com.br/books?id=EuA5OAAACAAJ>.

MAY, E. E.; VOUK, M. a.; BITZER, D. L.; ROSNICK, D. I. An error-correcting code framework for genetic sequence analysis. *Journal of the Franklin Institute*, v. 341, n. 1-2, p. 89–109, jan. 2004. ISSN 00160032. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0016003203000929>.

NEI, M.; KUMAR, S. *Molecular Evolution and Phylogenetics*. Oxford University Press, 2000. ISBN 9780195135855. Disponível em: <https://books.google.com.co/books?id=0nt-qaAfIbAC>.

NELSON, D.; LEHNINGER, A.; COX, M. *Lehninger Principles of Biochemistry*. W. H. Freeman, 2008. (Lehninger Principles of Biochemistry). ISBN 9780716771081. Disponível em: <https://books.google.com.co/books?id=5Ek9J4p3NfkC>.

NORTON, G.; SALAGEAN, A. On the Hamming distance of linear codes over a finite chain ring. *Information Theory, IEEE Transactions on*, v. 46, n. 3, p. 1060–1067, May 2000. ISSN 0018-9448.

ONO, K.; MUETZE, T.; KOLISHOVSKI, G.; SHANNON, P.; DEMCHAK, B. CyREST: Turbocharging Cytoscape Access for External Tools via a RESTful API [version 1; referees: 2 approved]. *F1000Research*, v. 4, n. 478, 2015.

PETERSON, W.; WELDON, E. *Error-correcting Codes*. MIT Press, 1972. ISBN 9780262160391. Disponível em: <http://books.google.com.br/books?id=5kfwlFeklx0C>.

ROCHA, A. *Digital communication system model for mitochondrial protein import by use of error-correcting codes*. Tese (Doutorado) — University of Campinas, 2010.

ROCHA, A.; FARIA, L.; KLEINSCHMIDT, J.; Palazzo Jr, R.; SILVA-FILHO, M. DNA sequences generated by $\mathbb{Z}_4$-linear codes. In: *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*. [S.l.: s.n.], 2010. p. 1320–1324.

ROKDE, C. N.; KSHIRSAGAR, M. Bioinformatics: Protein structure prediction. *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, Ieee, p. 1–5, jul. 2013. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6726753>.

ROMÁN-ROLDÁN, R.; BERNAOLA-GALVÁN, P.; OLIVER, J. Application of information theory to {DNA} sequence analysis: A review. *Pattern Recognition*, v. 29, n. 7, p. 1187–1194, 1996. ISSN 0031-3203. Disponível em: <http://www.sciencedirect.com/science/article/pii/003132039500145X>.

ROSEN, G. Examining coding structure and redundancy in DNA. *IEEE Engineering in Medicine and Biology Magazine*, v. 25, n. 1, p. 62–68, Jan 2006. ISSN 0739-5175.

RZESZOWSKA-WOLNY, J. Is genetic code error-correcting? *Journal of Theoretical Biology*, v. 104, n. 4, p. 701–702, 1983. ISSN 0022-5193. Disponível em: <http://www.sciencedirect.com/science/article/pii/0022519383902576>.

SAITOU, N.; NEI, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, v. 4, n. 4, p. 406–425, 1987. Disponível em: <http://mbe.oxfordjournals.org/content/4/4/406.abstract>.

SCHNEIDER, h. D.; SCHNEIDER, T. D.; STEPHENS, R. M. Sequence Logos: A New Way to Display Consensus Sequences. *Nucleic Acids Res*, v. 18, p. 6097–6100, 1990.

SCHNEIDER, T. D.; SPOUGE, J. Information content of individual genetic sequences. *J. Theor. Biol.*, v. 189, p. 427–441, 1997. http://alum.mit.edu/www/toms/papers/ri/.

SCHNEIDER, T. D.; STORMO, G. D.; GOLD, L.; EHRENFEUCHT, A. Information content of binding sites on nucleotide sequences. *Journal of Molecular Biology*, v. 188, n. 3, p. 415–431, 1986. ISSN 0022-2836. Disponível em: <http://www.sciencedirect.com/science/article/pii/0022283686901658>.

SHANKAR, P. On BCH codes over arbitrary integer tings (Corresp.). *Information Theory, IEEE Transactions on*, v. 25, n. 4, p. 480–483, Jul 1979. ISSN 0018-9448.

SHANNON, C. E. A Mathematical Theory of Communication. *Bell System Technical Journal*, Blackwell Publishing Ltd, v. 27, n. 3, p. 379–423, 1948. ISSN 1538-7305. Disponível em: <http://dx.doi.org/10.1002/j.1538-7305.1948.tb01338.x>.

SHANNON, P.; MARKIEL, A.; OZIER, O.; BALIGA, N. S.; WANG, J. T.; RAMAGE, D.; AMIN, N.; SCHWIKOWSKI, B.; IDEKER, T. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, v. 13, n. 11, p. 2498–2504, 2003. Disponível em: <http://genome.cshlp.org/content/13/11/2498.abstract>.

SPIEGEL, E. Codes over $\mathbb{Z}_m$, revisited. *Information and Control*, v. 37, n. 1, p. 100–104, 1978. ISSN 0019-9958. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0019995878904618>.

SWANSON, R. A unifying concept for the amino acid code. *Bulletin of Mathematical Biology*, Kluwer Academic Publishers, v. 46, n. 2, p. 187–203, 1984. ISSN 0092-8240. Disponível em: <http://dx.doi.org/10.1007/BF02460068>.

TAMURA, K.; DUDLEY, J.; NEI, M.; KUMAR, S. MEGA4: Molecular Evolutionary Genetics Analysis (MEGA) Software Version 4.0. *Mol. Biol. Evol.*, v. 24, n. 8, p. 1596–1599, 2007.

TAYLOR, W. R. The classification of amino acid conservation. *Journal of Theoretical Biology*, v. 119, n. 2, p. 205–218, 1986. ISSN 0022-5193. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0022519386800753>.

WALKER, J. L. *Codes and Curves*. American Mathematical Society, 2000. (Student mathematical library). ISBN 9780821826287. Disponível em: <http://books.google.com.br/books?id=w7oWu70CnV8C>.

WAN, Z.-X. X.; WAN, C.-H. *Quaternary Codes*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 1998. ISBN 9810232748.

YOCKEY, H. *Information Theory and Molecular Biology*. Cambridge University Press, 1992. ISBN 9780521350051. Disponível em: <https://books.google.com.co/books?id=MIxjQgAACAAJ>.

ZHANG, J. Protein-length distributions for the three domains of life. *Trends in Genetics*, v. 16, n. 3, p. 107–109, 2000. ISSN 0168-9525. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0168952599019228>.

ZYABLOV, V.; SHAVGULIDZE, S.; BOSSERT, M. An Introduction to Generalized Concatenated Codes. *European Transactions on Telecommunications*, Wiley Subscription Services, Inc., A Wiley Company, v. 10, n. 6, p. 609–622, 1999. ISSN 1541-8251. Disponível em: <http://dx.doi.org/10.1002/ett.4460100606>.