Luiz Antonio Leandro Franco

Análise das propriedades matemáticas associadas ao splicing alternativo através dos códigos BCH e de Varshamov-Tenengolts

Campinas
2014

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Luiz Antonio Leandro Franco

Análise das propriedades matemáticas associadas ao splicing alternativo através dos códigos BCH e de Varshamov-Tenengolts

Dissertação de mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Telecomunicações e Telemática.

Orientador: Reginaldo Palazzo Júnior

Este exemplar corresponde à versão final da tese defendida pelo aluno, e orientada pelo Prof. Dr. Reginaldo Palazzo Júnior

_____

Campinas
2014

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

F848a

Franco, Luiz Antonio Leandro, 1984-
Análise das propriedades matemáticas associadas ao splicing alternativo através dos códigos BCH e de Varshamov-Tenengolts / Luiz Antonio Leandro Franco. – Campinas, SP : [s.n.], 2014.

Orientador: Reginaldo Palazzo Júnior.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Códigos corretores de erros (Teoria da informação). 2. Genomas. 3. Teoria da informação. I. Palazzo Júnior, Reginaldo,1951-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Analysis of the mathematical properties associated to the alternative splicing through BCH and Varshamov-Tenengolts codes
Palavras-chave em inglês:
Brokers error codes (Information theory)
Genomes
Information theory
Área de concentração: Telecomunicações e Telemática
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Reginaldo Palazzo Júnior [Orientador]
Carlos Eduardo Camara
Andréa Santos Leite de Rocha
Data de defesa: 31-07-2014
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Luiz Antonio Leandro Franco

Data da Defesa: 31 de julho de 2014

Título da Tese: "Análise das Propriedades Matemáticas Associadas ao Splicing Alternativo Através dos Códigos BCH e de Varshamov-Tenengolts"

Prof. Dr. Reginaldo Palazzo Júnior (Presidente): _____

Prof. Dr. Carlos Eduardo Camara: _____

Dra. Andréa Santos Leite da Rocha: _____

Resumo

Durante milhões de anos, o homem, os animais e plantas vêm se transformando e evoluindo para se adaptar ao ambiente. Um processo que auxilia na evolução é o splicing alternativo, consistindo de uma codificação bastante conveniente, que a partir de um único gene consegue gerar várias proteínas, combinando éxons e íntrons de diferentes formas, aumentando assim a capacidade proteômica. Várias pesquisas buscam uma melhor compreensão dos mecanismos envolvidos no splicing altenativo e quais as consequências dos erros cometidos durante este processo. Este trabalho tem como objetivo principal analisar as propriedades matemáticas envolvidas no splicing alternativo por meio dos códigos corretores de erros. Os códigos (BCH) foram utilizados nos casos que ocorreram erros de substituição de nucleotídeos e os códigos de Varshamov-Tenengolts nos casos que ocorreram erros de inserção e deleção de nucleotídeos. Neste trabalho verificamos a possibilidade reproduzir matematicamente o splicing alternativo de acordo com as restrições biológicas. Para atingir este objetivo, consideramos o gene TRAV7 presente no genoma humano e o gene Hint-1 presente no nematoide Caenorhabditis Elegans.

Palavras-chave: Códigos BCH, Códigos de Varshamov-Tenengolts e Splicing Alternativo.

Abstract

During millions of years mankind, animals and plants have transformed themselves, continuing to evolve in order to adapt themselves to the environment. A process that helps in the evolution is the alternative splicing, consisting of a rather suitable codification, that manages to produce several proteins from a single gene, combining exons and introns of different forms, in this way increasing the proteomic capacity. Several surveys search for both a better understanding of the mechanisms involved in alternative splicing and the consequences of errors committed during this process. This study has as its main objective to analyze the mathematical properties involved in the alternative splicing through correcting codes of errors. The codes (BCH) were used in the cases when errors of substitution of nucleotides occurred and Varshamov-Tenengolts codes in the cases when errors of insertion and deletion of nucleotides occurred. In this study we verified the possibility of reproducing mathematically the splicing alternative in accordance with the biological restrictions. To achieve this objective we considered the gene TRAV7 present in the human genome and the gene Hint-1 present in the nematode Caenorhabditis Elegans.

Key-words: BCH codes, Varshamov-Tenengolts codes and Alternative Splicing.

Sumário

1 Introdução **1**
- 1.1 **Proposta de trabalho** . . . 2
- 1.2 **Descrição do trabalho** . . . 2

2 Estrutura Biológica **5**
- 2.1 **Funcionamento de uma célula** . . . 5
- 2.2 **DNA, RNA e mRNA** . . . 8
  - 2.2.1 **O DNA** . . . 8
  - 2.2.2 **Do DNA ao RNA** . . . 10
- 2.3 **Genoma e Genes** . . . 12
- 2.4 **Proteínas** . . . 13
- 2.5 **Splicing Alternativo** . . . 16

3 Códigos Corretores de Erros de Substituição, Deleção e Inserção **21**
- 3.1 **Códigos Corretores de Erros de Deleção e Inserção** . . . 22
  - 3.1.1 **Códigos de Varshamov-Tenengolts** . . . 24
- 3.2 **Códigos Corretores de Erros de Substituição** . . . 26
  - 3.2.1 **Anéis** . . . 26
  - 3.2.2 **Corpos algébricos de Galois** . . . 27
  - 3.2.3 **Códigos de bloco** . . . 29
  - 3.2.4 **Códigos lineares** . . . 30
  - 3.2.5 **Códigos cíclicos sobre $\mathbb{Z}_q$** . . . 32
  - 3.2.6 **Códigos BCH sobre anéis e corpos** . . . 33

4 Análise do Splicing Alternativo via CCE **43**
- 4.1 **Modelo para a Geração de Partes de uma Sequência** . . . 44
  - 4.1.1 **Geração de partes de uma sequência de informação** . . . 44
  - 4.1.2 **Utilização do código de Varshamov-Tenengolts** . . . 48
- 4.2 **Modelo para a geração de éxons e íntrons** . . . 51
  - 4.2.1 **Gene Trav7** . . . 51
  - 4.2.2 **Gene Hint-1** . . . 59
- 4.3 **Modelo para a geração de partes de um genoma** . . . 69
- 4.4 **Uso do código de Varshamov-Tenengolts** . . . 75
  - 4.4.1 **Gene Trav7** . . . 76

4.4.2 **Gene Hint-1** 82

5 Conclusões e Sugestões de Trabalhos Futuros **93**

Bibliografia **96**

A todos os meus familiares;
em especial:
minha esposa márcia e meus pais josé e catarina, pela paciência, apoio, estímulo e amor.
Dedico

Agradecimentos

Agradeço,

ao Prof. Dr. Reginaldo Palazzo Júnior os mais de dois anos de sua valorosa orientação, pela sua dedicação, paciência e a oportunidade de compartilhar um pouco de seu conhecimento.

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

Aos colegas de trabalho mais próximos: Anderson, Mário, Rodrigo, Lailson, Gustavo, Lucas, Fernando, Cintia, Luzinete, Nelson, Gabriela, Cibele, Clarice, Cátia, Leandro e Diogo a convivência descontraída e as trocas de experiências.

Aos demais colegas do Departamento de Comunicação a ótima convivência.

Aos professores da FEEC: Yuzo Iano, Von Zuben e Palazzo pelos ótimos cursos oferecidos.

Ao professor do CBMEG: Paulo Arruda pelo ótimo curso oferecido.

Aos professores Dr. Mario Henrique Bengtson (IB/UNICAMP) e Dra. Katlin Brauer Massirer (CBMEG/UNICAMP) pelas parcerias e discussões frutíferas.

À agência CAPES o apoio financeiro concedido durante todo o período do mestrado.

À minha esposa, Márcia, que sempre esteve do meu lado com muita paciência, dedicação e amor. O seu companheirismo foi fundamental na concretização deste trabalho. Muito obrigada por fazer parte da minha vida.

Aos meus pais José e Catarina: a minha eterna gratidão, por tudo o que fizeram por mim e por todo apoio que sempre tive para concluir meus estudos - Jamais se esqueçam que eu levarei para sempre um pedaço do ser de cada um dentro do meu ser.

À minha sogra, dona Rosa, e meu sogro, Sr. Lúcio meu muito obrigada pelo carinho e confiança e por me presentear com uma de suas filhas, Márcia.

À todos os meus familiares, que souberam compreender minhas ausências e sempre me deram a força necessária para seguir em frente.

À minha irmã, Juliana e meu cunhado Amilton e suas filhas Júlia e Jéssica, meu enorme carinho.

Meu muito obrigado a meu cunhado Thiago e minhas cunhadas Pollyanna e Vanessa pelo apoio.

À FEEC/UNICAMP a ótima estrutura que oferece aos estudantes e pesquisadores.

Aos funcionários da Faculdade de Engenharia Elétrica e de Computação, que de alguma forma

contribuíram para a realização deste trabalho.

O que sabemos não é muito. O que não sabemos é imenso.

Pierre Simon Laplace

Lista de Figuras

2.1 Principais características das células eucarióticas, encontrada em [1] 6
2.2 Maquinarias do Splicing em relação ao éxon, encontrada em [2] 17
2.3 Maquinarias do Splicing em relação ao íntron, encontrada em [2] 18
2.4 Principais tipos de Splicing, encontrados em [2]. 19

4.1 Sequência em nucleotídeos do gene Trav7. 52
4.2 Sequência em nucleotídeos do gene Hint-1 59
4.3 Sequência em nucleotídeos do genoma do Plasmídeo 70

Lista de Tabelas

3.1 Arranjo padrão. 31
3.2 Polinômios primitivos da extensão de Galois $r = 6$ 38
3.3 Rotulamentos determinados pelas 24 permutações, encontrada em [3]. 40

4.1 Palavra-código v 44
4.2 Matriz geradora G separada em partes 46
4.3 Palavra-código v separada em partes 47
4.4 Vetor u separado em partes 48
4.5 Palavra-código v 48
4.6 Vetor α resultante da Palavra-código v 48
4.7 Vetor v' resultante da deleção de um elemento da palavra-código v 49
4.8 Vetor α' resultante do vetor v' 49
4.9 Vetor α'_1 50
4.10 Sequência reconstruida 50
4.11 Vetor v' 51
4.12 Vetor α' 51
4.13 Vetor α'_1 51
4.14 Palavra-código v_1 gene Trav7 53
4.15 Vetor u_1 referente ao gene Trav7 54
4.16 Palavra-código separada em éxons e íntrons do gene Trav7 55
4.17 Vetor u_1 separado em éxons e íntrons do gene Trav7 57
4.18 Primeiro caso de splicing alternativo do gene Trav7 57
4.19 Segundo caso de splicing alternativo do gene Trav7 58
4.20 Palavra-código w_1 do gene Hint-1 60
4.21 Palavra-código w_1 separada em éxons e íntrons do gene Hint-1 61
4.22 Vetor y_1 referente ao gene Hint-1 62
4.23 Vetor y_1 separado em éxons e íntrons do gene Hint-1 65
4.24 Primeiro caso de splicing alternativo do gene Hint-1 66
4.25 Segundo caso de splicing alternativo do gene Hint-1 67
4.26 Terceiro caso de splicing alternativo do gene Hint-1 67
4.27 Quarto caso de splicing alternativo do gene Hint-1 68
4.28 Quinto caso de splicing alternativo do gene Hint-1 68
4.29 Parte 1 da palavra-código d_1 do Genoma do Plasmídeo 71
4.30 Parte 2 da palavra-código d_1 rótulo caso 1 do Genoma do Plasmídeo 72
4.31 Parte 1 vetor c_1 do Genoma do Plasmídeo 73

4.32 Parte 2 vetor c_1 do Genoma do Plasmídeo 74
4.33 Vetor α obtido através do vetor A referente ao gene Trav7 76
4.34 Vetor A' obtido após a deleção de informação referente ao gene Trav7 77
4.35 Vetor α' obtido através do vetor A' referente ao gene Trav7 78
4.36 Vetor α'_1 referente ao gene Trav7 79
4.37 Vetor A' referente ao gene Trav7 80
4.38 Vetor A originado durante o splicing alternativo referente ao gene Trav7 80
4.39 Vetor α gerado a partir do vetor A referente ao gene Trav7 81
4.40 Vetor A' referente ao gene Trav7 82
4.41 Vetor α' gerado a partir do vetor A' referente ao gene Trav7 82
4.42 Vetor α'_1 gerado a partir do vetor α' referente ao gene Trav7 83
4.43 Vetor A' corrigido referente ao gene Trav7 83
4.44 Vetor α gerado através do vetor A da tabela 4.20 referente ao gene Hint-1 84
4.45 vetor A' oriundo do vetor A referente ao gene Hint-1 85
4.46 vetor α' oriundo do vetor A' referente ao gene Hint-1 86
4.47 Vetor α'_1 oriundo do vetor α' referente ao gene Hint-1 87
4.48 Vetor A' oriundo do A após uma inserção referente ao gene Hint-1 88
4.49 Vetor α' oriundo do vetor A' referente ao gene Hint-1 89
4.50 Vetor α'_1 oriundo do vetor α' referente ao gene Hint-1 90
4.51 Palavra-código de um RNA maduro referente ao gene Hint-1 90
4.52 Vetor α correspondente palavra-código do RNA maduro referente ao gene Hint-1 91
4.53 Vetor A' oriuindo do vetor A referente ao gene Hint-1 91
4.54 Vetor α' corresponde do vetor A' referente ao gene Hint-1 91
4.55 Vetor α'_1 referente ao gene Hint-1 92

Capítulo 1

Introdução

Os códigos corretores de erros (CCEs) são uma forma organizada de acrescentar algum dado a cada informação que será transmitida ou armazenada, mas que ao recuperar esta informação possamos detectar e corrigir possíveis erros ocorridos no processo de transmissão. A teoria dos códigos faz a união de conceitos e técnicas importantes da Álgebra abstrata com aplicações em nosso cotidiano, assim mostrando que existe uma sofisticação tecnológica que torna cada vez mais imperceptível a relação entre a matemática pura e a matemática aplicada. O sistema biológico transmite e armazena informações fazendo uso do código genético. Neste trabalho, mostraremos como códigos corretores de erros podem ser associados ao código genético, sendo uma área de pesquisa bastante promissora tanto no mundo acadêmico como no mundo industrial.

Os CCEs são utilizados sempre que se deseja transmitir ou armazenar uma informação, na biologia o código genético transmite e armazena a informação ao longo do tempo. A união destas duas áreas do conhecimento é um grande desafio para os pesquisadores que estudam a biologia molecular e a teoria da informação e codificação. Uma grande dificuldade é mostrar a existência de CCEs na estrutura do DNA. De acordo com, [3] os resultados podem ser direcionados em metodologias voltadas em análises mutacionais e de polimorfismos, reduzindo tempo e custos laboratoriais.

Nos trabalhos, [4] e [3] pela primeira vez sequências primárias de uma fita simples do DNA, com características biológicas distintas e comprimentos variados, são identificadas como palavras-código de um CCE e reproduzidas em termos dos nucleotídeos e dos aminoácidos correspondentes. Outro avanço é com relação a identificação da sequência da dupla hélice do DNA como palavra-código de um CCE e sua reprodução em termos das bases complementares.

Na Seção 1.1 apresentaremos a proposta do presente trabalho que tem como objetivo analisar as propriedades matemáticas envolvidas no Splicing Alternativo, fazendo uso dos códigos corretores de erros de substituição, inserção e deleção. Na Seção 1.2 mostraremos como o presente trabalho está organizado. A seguir, de maneira resumida, comentaremos a proposta de pesquisa e os principais objetivos deste trabalho.

1.1 Proposta de trabalho

Para que possamos entender a proposta de trabalho é necessário o entendimento de alguns conceitos iniciais que serão mostrados a seguir. Como o trabalho usa conceitos de teoria da informação e conceitos biológicos se faz necessário um breve relato sobre os mesmos.

As sequências codificantes de genes eucarióticos são caracteristicamente interrompidas por sequências intervenientes não-codificantes (íntrons). Tanto as sequências de íntrons quanto de éxons são transcritas em RNA. As sequências dos íntrons são removidas do RNA transcrito por meio de um processo denominado *splicing de RNA*. Grande parte do splicing de RNA que ocorre nas células atua na produção de mRNA, sendo denominado splicing do precursor de mRNA (ou pré-mRNA).

Neste trabalho, estamos interessados em analisar as propriedades matemáticas associadas ao splicing alternativo, usando a estrutura de códigos corretores de erros, visto que ainda não há relatos na literatura deste tipo de abordagem referente ao splicing alternativo. Neste caso estaremos usandos dois tipos de códigos para fazer as análises matemáticas, usaremos o código BCH e o código de Varshamov-Tenengolts, tendo como objetivos principais:

1. Dada a localização dos éxons e íntrons na sequência genética, encontramos a localização de éxons e íntrons na matriz geradora do código BCH, na palavra-código v e no vetor de sinalização u associado ao gene.

2. Localizar as submatrizes referentes a éxons e íntrons e fazer uma possível associação a um código de memória unitária parcial.

3. Mostrar que o splicing alternativo pode ser modelado matematicamente para os casos dos genes Trav7 e Hint-1.

4. Apresentar uma análise usando o código de Varshamov-Tenengolts para demonstrar que além do código BCH outros códigos podem ser associados ao processamento da informação genética.

A seguir, mostraremos como o trabalho está organizado e estruturado para facilitar o entendimento do leitor.

1.2 Descrição do trabalho

No Capítulo 2 faremos uma revisão dos principais conceitos biológicos, onde será apresentada uma breve introdução sobre o funcionamento da célula, em seguida introduziremos os conceitos e propriedades do DNA, RNA e mRNA. Ainda neste capítulo apresentaremos as definições de genoma e gene. Em seguida apresentaremos as principais propriedades e conceitos sobre as proteínas. Finalmente, apresentaremos as propriedades e os principais tipos de splicing alternativo (AS).

O Capítulo 3 tem como principal objetivo introduzir os conceitos de CCEs de substituição, deleção e inserção. Neste capítulo tanto códigos corretores de erros de deleção e inserção para

alfabetos binários, como a generalização dos códigos de Varshamov-Tenengolts para alfabeto q-ário são mostrados. Para uma melhor compreensão de códigos corretores de erros de substituição introduziremos os conceitos das estruturas algébricas de anéis e corpos, em seguida mostraremos as definições dos códigos de blocos. Além disso, os principais conceitos de códigos lineares e códigos cíclicos sobre $\mathbb{Z}_q$ são apresentados, finalmente as definições e teoremas relacionados aos códigos BCH sobre anel são considerados, conduzindo dessa forma ao algoritmo de codificação para a geração de sequências gênicas e genômicas.

No Capítulo 4 são apresentados os resultados obtidos na análise do splicing alternativo. Primeiramente será mostrado um exemplo de como gerar partes de uma informação e de como corrigir erros de inserção e deleção usando os códigos q-ário de Varshamov-Tenengolts. Em seguida, consideraremos exemplos do modelo de geração de éxons e íntrons, associados ao gene Trav7 do genoma humano do gene Hint-1 do nematoide Caenorhabditis Elegans do genoma do Plasmídeo. Finalmente é mostrado como corrigir erros de deleção e inserção de nucleotídeos usando o código de Varshamov-Tenengolts.

No Capítulo 5 as conclusões são apresentadas, bem como as tendências e trabalhos futuros.

Capítulo 2

Estrutura Biológica

Neste capítulo apresentaremos alguns conceitos da estrutura e funcionamento da biologia molecular, imprescindíveis para o desenvolvimento do presente trabalho. Na seção 2.1 faremos um breve resumo da unidade básica da vida a célula e seu funcionamento. Na Seção 2.2 apresentaremos uma introdução da molécula de DNA (ácido desoxirribonucleico) que contém as instruções genéticas que coordenam o desenvolvimento e funcionamento de todos os seres vivos, introduziremos também os conceitos do RNA (ácido ribonucleico) responsável pela síntese de proteínas da célula e abordaremos as funções do mRNA (RNA mensageiro). Na Seção 2.3 faremos uma breve introdução sobre os genes e suas propriedades bem como alguns aspectos relevantes sobre genoma. Na Seção 2.4 revisaremos alguns conceitos fundamentais sobre as proteínas como aminoácidos, funções e estrutura, para que possamos compreender melhor o sistema biológico. Na Seção 2.5 definiremos o splicing alternativo relatando: os organismos que sofrem splicing alternativo, o aumento da capacidade de codificação dos genes, o splicing alternativo em plantas e os tipos mais comuns de splicing alternativo. Neste capítulo foram adotadas como referências os livros [5] e [1].

2.1 Funcionamento de uma célula

A célula representa a menor porção de matéria viva, onde encontram-se as unidades estruturais e funcionais dos organismos vivos, sendo a unidade básica da vida. As formas mais simples de vida são organismos unicelulares que se reproduzem dividindo-se em duas partes, cissiparidade. Cada célula que forma nosso corpo deve crescer, reproduzir-se, processar informações, responder a estímulos e realizar uma série considerável de reações químicas.

As células são envolvidas pela membrana celular e preenchidas com uma solução aquosa concentrada de substâncias químicas, substâncias físicas e o citoplasma. De acordo com [5], o citoplasma é o material celular localizado entre a membrana celular e o núcleo, sendo o local onde se concentra o maior número de atividades. As organelas citoplasmáticas são compartimentos celulares especializados, cada um realizando o seu próprio trabalho para manter a vida da célula. O citoesqueleto, os centríolos e os ribossomos são exemplos desses tipos de organelas. As organelas membranáceas da célula incluem as mitocôndrias, os peroxissomos, os lisossomos, o retículo endoplasmático e o aparelho de Golgi. Além de fornecer um excelente isolamento,

uma organela membranácea muitas vezes une-se ao restante de um sistema intracelular interativo conhecido como sistema de endomembranas, sendo as principais características das células eucarióticas mostradas na Figura 2.1.

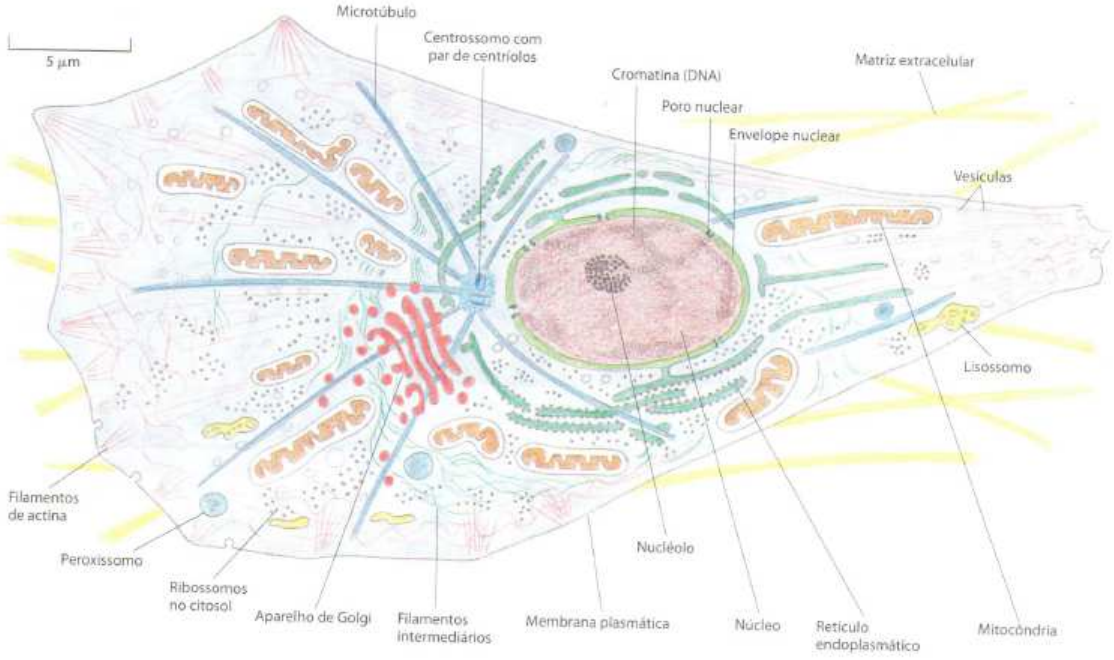

Figura 2.1: Principais características das células eucarióticas, encontrada em [1]

As mitocôndrias são organelas membranáceas alongadas ou em forma de salsicha. Nas células elas se torcem e se alongam, mudando de forma quase que continuamente, elas são as usinas de energia de uma célula, fornecendo a maior parte do suprimento celular de ATP. As células muito ativas, como as células renais e hepáticas, possuem centenas de mitocôndrias, enquanto as células relativamente inativas (como um linfócito virgem) possuem apenas algumas. As mitocôndrias são envolvidas por duas membranas, cada uma com uma estrutura semelhante à da membrana plasmática. A membrana externa é lisa e sem características especiais, mas a membrana interna contém cristas em forma de prateleiras, sendo encontrada uma substância gelatinosa na parte interna. Os produtos intermediários obtidos do combustível alimentar (como glicose, entre outros) são metabolizados até dióxido de carbono e água por conjuntos de enzimas, algumas dissolvidas na matriz mitocondrial e outras fazendo parte das cristas.

As mitocôndrias são organelas complexas: elas contêm seu próprio DNA e RNA e são capazes de se auto-reproduzir. Os genes mitocondriais (cerca de 37) controlam a síntese de cerca de 5% das proteínas necessárias para a função mitocondrial, e o DNA nuclear codifica as demais proteínas necessárias para a respiração celular. As mitocôndrias são similares aos representantes de um grupo específico de bactérias, e o DNA mitocondrial é semelhante ao DNA encontrado nas células bacterianas. Atualmente, é amplamente aceita a ideia de que as mitocôndrias originaram-se de bactérias que invadiram as células ancestrais de plantas e de animais.

Os ribossomos, pequenos grânulos de coloração escura, são formados por proteínas e por um tipo de RNA chamado RNA ribossômico. Cada ribossomo tem duas subunidades globu-

lares que se acoplam, sendo locais da síntese protéica. Alguns ribossomos flutuam livremente pelo citoplasma, outros entretanto, estão ligados às membranas, formando um complexo chamado retículo endoplasmático rugoso. Os ribossomos livres produzem as proteínas solúveis que atuam no citoplasma, os ribossomos ligados à membrana sintetizam as proteínas destinadas à incorporação em membranas celulares ou à exportação (para fora da célula).

O retículo endoplasmático (RE) é um extenso sistema de tubos interconectados e membranas paralelas que encerram cavidades preenchidas por líquido, as cisternas, que se curvam e serpenteiam ao longo do citoplasma. O RE é contínuo com a membrana da célula. Existem dois tipos de RE: o RE rugoso e o RE liso. O RE rugoso tem a sua superfície externa crivada de ribossomos, as proteínas produzidas nesses ribossomos passam para o interior das cisternas do RE, preenchidas por líquido, de onde podem seguir vários destinos. O RE liso esta em comunicação com RE rugoso e consiste de túbulos organizados na forma de uma rede. Suas enzimas (proteínas integrais que fazem parte de suas membranas) não atuam na síntese de proteínas. Em vez disso, elas catalisam reações envolvidas em vários processos.

O aparelho de Golgi consiste de uma pilha de sacos membranosos achatados, como pratos rasos ocos, associados a abundantes vesículas membranosas diminutas. O aparelho de Golgi é o mais importante controlador de tráfego das proteínas celulares. Suas principais atribuições são: modificar, concentrar e empacotar as proteínas e os lipídeos formados no RE rugoso.

Os lisossomos são organelas membranáceas esféricas que contêm enzimas digestivas. Como se pode esperar, os lisossomos são grandes e abundantes nos fagócitos, as células que eliminam as bactérias invasoras e os restos celulares. As enzimas lisossômicas podem digerir quase todos os tipos de moléculas biológicas. Elas funcionam melhor em condições ácidas e, por isso, são chamadas hidrolases ácidas. Os lisossomos trabalham como uma equipe de demolição da célula executando as seguintes tarefas: digerindo partículas captadas por endocitose, particularmente bactérias, vírus e toxinas ingeridas; degradando organelas esgotadas ou não-funcionais; realizando funções metabólicas, como a degradação e a liberação de glicogênio; destruindo tecidos inúteis, como o tecido presente entre os dedos das mãos e dos pés de um feto em desenvolvimento; degradando o tecido ósseo liberando íons cálcio no sangue.

Os peroxissomos são sacos membranáceos contendo diversas enzimas poderosas, das quais as mais importantes são as oxidases e as catalases. Sua principal função é a de neutralizar os perigosos radicais livres, substâncias químicas contendo elétrons desemparelhados e extremamente reativos que podem alterar a estrutura das moléculas biológicas. As oxidases convertem os radicais livres em peróxido de hidrogênio, o qual também é reativo e perigoso, mas é rapidamente convertido em água pela enzima catalase.

O citoesqueleto é uma elaborada rede de filamentos distribuída pelo citosol. Esta rede funciona como os ossos, os músculos e os ligamentos da célula, sustentando as estruturas celulares e fornecendo os dispositivos necessários para gerar diversos movimentos celulares. Os três tipos de filamentos do citoesqueleto são os microtúbulos, os microfilamentos e os filamentos intermediários, sendo que nenhum deles é revestido por membrana.

A grande maioria das células contém um núcleo que funciona como uma biblioteca genética, contendo as instruções necessárias para construir todas as proteínas do corpo, podendo ser comparado a um computador, sendo a maior organela citoplasmática. O núcleo determina os

tipos e as quantidades das proteínas que devem ser sintetizadas.

A maioria das células apresenta apenas um núcleo, mas algumas, incluindo as células musculares esqueléticas, as células responsáveis pela reabsorção óssea, e algumas células hepáticas, são multinucleadas, isto é, possuem muitos núcleos. Todas as células corporais são nucleadas, com uma exceção os eritrócitos, células que têm os núcleos expulsos antes da entrada na corrente sanguínea. Sem um núcleo, uma célula é incapaz de produzir o mRNA para a síntese.

2.2 DNA, RNA e mRNA

2.2.1 O DNA

Uma molécula de ácido desoxirribonucleico (DNA) consiste de duas longas cadeias polipeptídicas compostas por quatro tipos de subunidades nucleotídicas. Cada uma dessas cadeias é conhecida como uma cadeia de DNA, ou fita de DNA. As ligações de hidrogênio entre as bases dos nucleotídeos mantêm as duas cadeias unidas. No caso dos nucleotídeos do DNA, o açúcar é uma desoxirribose ligada a um único grupo fosfato, e a base pode ser adenina (A), citosina (C), guanina (G) ou timina (T). A forma na qual as subunidades nucleotídicas estão ligadas confere uma polaridade química à fita de DNA. A estrutura tridimensional do DNA a dupla-hélice é decorrente das características químicas e estruturais de suas duas cadeias polinucleotídicas.

As formas e a estrutura química das bases permitem que as ligações de hidrogênio sejam formadas eficientemente apenas entre A e T e entre G e C, assim essa complementaridade de bases permite que os pares de bases sejam dispostos em um arranjo energético mais favorável no interior da dupla-hélice. O DNA codifica a informação por meio da ordem de nucleotídeos ao longo da fita. Cada base A, C, T ou G pode ser considerada como uma letra de um alfabeto de quatro letras que escreve mensagens biológicas na estrutura química do DNA.

A sequência linear de nucleotídeos em um gene deve, portanto, corresponder à sequência linear de aminoácidos em uma proteína. O processo em que a célula primeiro converte a sequência nucleotídica de um gene em uma sequência de nucleotídeos na molécula de RNA, e então na sequência de aminoácidos de uma proteína é conhecido como expressão gênica. A série completa de informação do DNA de um organismo é chamada genoma e contém a informação para todas as proteínas e moléculas de RNA que o organismo irá sintetizar durante sua existência. A cada divisão celular, a célula deve copiar seu genoma e passá-lo para as duas células-filhas. A capacidade de cada fita de DNA de atuar como um molde para a produção de uma fita complementar permite que a célula possa copiar ou replicar seus genes antes de passá-los a seus descendentes.

Quase todo o DNA de uma célula eucariótica está contido em um núcleo que ocupa cerca de 10% do volume celular total. Esse compartimento é delimitado por um envelope nuclear formado por duas membranas lipídicas concêntricas. O envelope nuclear permite que muitas proteínas que atuam no DNA estejam concentradas onde são necessárias à célula e mantendo as enzimas nucleares separadas das enzimas citoplasmáticas, uma característica crucial para o funcionamento adequado das células eucarióticas. A função mais importante do DNA é carregar os genes, a informação que especifica todas as proteínas e moléculas de RNA que constituem um organismo.

Embora o DNA seja extremamente compactado, essa compactação é feita de forma a permitir

que ele esteja prontamente disponível às muitas enzimas nas células que irão replicá-lo, repará-lo e usar seus genes para produzir moléculas de RNA e proteínas. Nos eucariotos, o DNA nuclear é dividido em uma série de diferentes cromossomos. Cada cromossomo consiste de uma única e enorme molécula de DNA linear com proteínas associadas que dobram e empacotam a fina fita de DNA em uma estrutura mais compacta. O complexo DNA e proteínas é chamado cromatina.

Os cromossomos carregam os genes, as unidades funcionais da hereditariedade. Um gene normalmente é definido como um segmento de DNA que contém as instruções para produzir uma determinada proteína ou uma série de proteínas relacionadas. Com a publicação do primeiro rascunho de todo o genoma humano em 2001 e a sequência de DNA finalizada em 2004, a informação genética em todos os cromossomos humanos está disponível. A primeira característica marcante do genoma humano é que apenas uma parte muito pequena codifica proteínas. A maioria do DNA cromossômico é constituído por pequenos segmentos móveis de DNA que gradualmente foram inseridos nos cromossomos com o passar do tempo.

Uma segunda característica marcante do genoma humano é o tamanho médio dos genes, cerca de 27.000 pares de nucleotídeos. A maior parte do DNA restante no gene consiste em inúmeros segmentos de DNA não-codificante que interrompem uma sequência relativamente curta de pequenos segmentos de DNA codificante para a proteína. As sequências codificantes são chamadas éxons, as sequências intercalantes não-codificantes são denominadas íntrons. Embora alterações genéticas ocasionais aumentem a sobrevivência a longo prazo de uma espécie, a sobrevivência de um organismo requer alta estabilidade genética. Raramente os processos de manutenção do DNA celular falham, resultando em uma alteração permanente no DNA. Tal alteração é chamada mutação, podendo destruir um organismo, se ocorrer em uma posição vital na sequência de DNA.

Em todas as células, as sequências de DNA são mantidas e replicadas com alta fidelidade. A taxa de mutação é de aproximadamente um nucleotídeo alterado por 10^9 nucleotídeos cada vez que o DNA é replicado, é praticamente a mesma em organismos tão diferentes como bactérias e seres humanos. Devido a essa incrível precisão, a sequência do genoma humano (cerca de $3x10^9$ pares de nucleotídeos) é alterada em apenas três nucleotídeos a cada divisão celular. Isso permite que a maioria dos seres humanos transmita instruções genéticas precisas de uma geração a outra e, também, evita que as alterações nas células somáticas originem câncer.

A replicação do DNA ocorre em uma estrutura em forma de Y, chamada forquilha de replicação. A enzima DNA-polimerase autocorretiva, catalisa a polimerização de nucleotídeos na direção 5'-3', copiando uma fita-molde de DNA com extraordinária fidelidade. Como as duas fitas da dupla-hélice de DNA são antiparalelas, essa síntese de DNA 5'-3' só pode ser realizada continuamente em uma das fitas da forquilha de replicação (fita-líder). Na fita descontínua, pequenos fragmentos de DNA são sintetizados de trás para frente. Uma vez que a DNA-polimerase autocorretiva não pode iniciar uma nova cadeia, esses fragmentos da fita descontínua são iniciados por pequenas moléculas de RNA, que são, subsequentemente, removidas e substituídas por DNA.

A informação genética só pode ser armazenada de modo estável nas sequências de DNA devido a um grande grupo de enzimas de reparo do DNA que continuamente verificam o DNA e substituem qualquer nucleotídeo alterado. A maioria dos tipos de reparo do DNA depende da

presença de uma cópia separada da informação genética em cada uma das duas fitas da dupla-hélice de DNA. Uma lesão acidental em uma fita pode, portanto, ser removida por uma enzima de reparo, e uma fita correta é ressintetizada, tendo como referência a informação contida na fita não-danificada.

Outros sistemas críticos de reparo com base nos mecanismos de junção de extremidades não-homólogas e recombinação homóloga unem quebras acidentais nas duas fitas que ocorrem na dupla fita de DNA. Na maioria das células, um nível elevado de lesões no DNA provoca um retardo no ciclo celular pelos pontos de verificação, que asseguram que o DNA danificado seja corrigido antes da divisão celular.

A recombinação homóloga (também chamada recombinação geral) resulta na transferência de informação genética entre dois segmentos de DNA de dupla-hélice com sequências nucleotídicas semelhantes. Esse processo é essencial para o reparo correto, livre de erros de cromossomos danificados em todas as células, sendo também responsável pelo entrecruzamento de cromossomos que ocorre durante a meiose. O evento de recombinação é guiado por um conjunto de proteínas especializadas. Embora possa ocorrer em qualquer sítio em uma molécula de DNA, uma extensa interação de pareamento de bases entre fitas complementares é sempre necessária entre as duas duplexes participantes.

2.2.2 Do DNA ao RNA

O DNA genômico não direciona a síntese proteica diretamente, mas utiliza o RNA como uma molécula intermediária. Quando a célula necessita de uma proteína específica, a sequência de nucleotídeos da região apropriada de uma molécula de DNA muito longa em um cromossomo é inicialmente copiada sob a forma de RNA (por meio de um processo denominado *transcrição*). São estas cópias de RNA de segmentos de DNA que são usadas diretamente como moldes para direcionar a síntese da proteína (em um processo denominado *tradução*). O fluxo de informação genética nas células é, portanto, de DNA para RNA para proteína. Todas as células, desde a bactéria até seres humanos, expressam sua informação genética dessa maneira, um princípio tão fundamental que é denominado o *dogma central* da biologia molecular.

A transcrição e a tradução são os meios pelos quais as células leem, ou expressam, as instruções genéticas de seus genes. Como muitas cópias idênticas de RNA podem ser produzidas a partir do mesmo gene, e como cada molécula de RNA pode direcionar a síntese de várias moléculas idênticas de proteína, as células podem, quando necessário, sintetizar rapidamente uma grande quantidade de proteína. Porém, cada gene também pode ser transcrito e traduzido sob taxas diferentes, permitindo que a célula faça enormes quantidades de certas proteínas e mínimas quantidades de outras.

Assim como o DNA, o RNA é um polímero linear composto de quatro tipos diferentes de subunidades nucleotídicas unidas entre si por ligações fosfodiéster. O RNA difere quimicamente do DNA em dois aspectos: os nucleotídeos do RNA são ribonucleotídeos, isto é, eles contêm o açúcar ribose em vez de desoxirribose; assim como o DNA, o RNA contém as bases adenina (A), guanina (G), e citosina (C), e uracila (U), ao invés da timina (T), que ocorre no DNA. Uma vez que U, assim como T, pode formar pares pelo estabelecimento de ligações de hidrogênio, as propriedades de complementaridade por pareamento de bases descritas para o DNA na Subseção

2.2.1 também se aplicam ao RNA como G ligando-se a C, e A com U. No entanto, é possível encontrar outros tipos de pareamento de bases no RNA: por exemplo, G ocasionalmente forma pares com U.

A transcrição começa com a desespiralização de uma pequena porção da dupla-hélice de DNA, que então, age como um molde para a síntese de uma molécula de RNA. Assim como na replicação de DNA, a sequência de nucleotídeos da cadeia de RNA é determinada pela complementaridade do pareamento de bases entre os nucleotídeos a serem incorporados e o DNA-molde. As enzimas que realizam a transcrição são denominadas RNA-polimerases. Assim como a DNA-polimerase catalisa a replicação do DNA, as RNA-polimerases catalisam a formação de ligações fosfodiéster que conectam os nucleotídeos entre si formando uma cadeia linear.

Embora as RNA-polimerases não sejam tão exatas quanto as DNA-polimerases que replicam DNA, elas têm um pequeno mecanismo de correção. Se um ribonucleotídeo incorreto for adicionado à cadeia de RNA em formação, a polimerase pode retornar, e o sítio ativo da enzima pode realizar uma reação de excisão semelhante ao procedimento reverso da reação de polimerização, exceto que será utilizada água em vez de pirofosfato e um monofosfato de nucleosídeo é liberado.

A maioria dos genes carregados no DNA das células especifica a sequência de aminoácidos de proteínas, as moléculas de RNA que são copiadas a partir desses genes (e que definem a síntese de proteínas) são chamadas moléculas de RNA mensageiro (mRNA). O produto final de uma minoria de genes, entretanto, é o próprio RNA. Tais RNAs, assim como as proteínas, servem como componentes estruturais e enzimáticos para uma ampla gama de processos na célula. Apesar de vários desses RNAs não-codificantes não terem suas funções conhecidas algumas moléculas de pequenos RNAs nucleares (snRNA, small nuclear RNA)direcionam o splicing (excisão de íntrons) do pré-RNA para formar o mRNA. Moléculas de RNA ribossomal (rRNA) formam o cerne dos ribossomos e moléculas de RNA transportador (tRNA) formam os adaptadores que selecionam aminoácidos e os colocam no local adequado nos ribossomos para serem incorporados em proteínas.

Antes que a síntese de uma determinada proteína possa ocorrer, a molécula de mRNA correspondente deve ser produzida por transcrição. As bactérias contêm um único tipo de RNA-polimerase (a enzima que realiza a transcrição de DNA em RNA). Uma molécula de mRNA é produzida quando esta enzima inicia a transcrição em um promotor, sintetiza o RNA pela extensão da cadeia, finaliza a transcrição em um terminador e libera tanto o DNA-molde quanto a molécula de mRNA finalizada. Nas células eucarióticas, o processo de transcrição é muito mais complexo, onde existem três RNA-polimerases denominados como polimerase I, II e III.

O mRNA dos eucariotos é sintetizado pela RNA-polimerase II. Essa enzima necessita de uma série de proteínas adicionais, denominadas fatores gerais de transcrição, para iniciar a transcrição sobre um DNA-molde purificado, e ainda de mais proteínas (como complexos remodeladores de cromatina e enzimas modificadoras de histonas) para iniciar a transcrição sobre a cromatina-molde dentro da célula.

Durante a fase de extensão da transcrição, o RNA em formação sofre três tipos de eventos de processamento: um nucleotídeo é adicionado à sua extremidade 5' (capeamento), os íntrons são removidos do meio da molécula de RNA (splicing) e a extremidade 3' do RNA é gerada (por

clivagem e poliadenilação). Cada um desses processos é iniciado por proteínas que acompanham a RNA-polimerase II por interação com sítios sobre sua longa cauda estendida C-terminal. O splicing difere dos demais pelo fato de muitas de suas etapas-chave serem mediadas por moléculas especializadas de RNA e não por proteínas. Os mRNAs adequadamente processados são transportados através de complexos de poro nuclear para o citosol, onde serão traduzidos em proteínas.

2.3 Genoma e Genes

As moléculas de DNA são muito grandes contendo as especificações para milhares de proteínas. Os segmentos individuais da sequência inteira de DNA são transcritos em moléculas de mRNA separadas, com cada seguimento codificando uma proteína diferente. Cada um desses segmentos de DNA representa um gene. Existe uma complexidade na qual uma molécula de RNA transcrita a partir de um mesmo segmento de DNA pode ser processada em mais de uma forma, originando assim um grupo de versões alternativas de uma proteína, especialmente em células mais complexas como as de plantas e animais. Portanto, um gene é, na maioria das vezes, definido como um segmento de DNA correspondente a uma única proteína, ou como um grupo de variantes proteicas (ou como uma única molécula de RNA catalítica ou estrutura para aqueles genes que produzem RNA, mas não proteínas).

Em todas as células, a expressão de genes individuais é regulada: em vez de manufaturar todo seu repertório de possíveis proteínas com toda intensidade, o tempo todo, a célula ajusta a velocidade de transcrição e de tradução de diferentes genes independentemente, de acordo com a necessidade. Os segmentos de DNA reguladores são interespaçados entre os segmentos que codificam as proteínas, e essas regiões não-codificadoras ligam-se a moléculas especiais de proteínas que controlam a velocidade local de transcrição. Outros segmentos de DNAs não-codificadores também estão presentes, alguns deles servindo, por exemplo, como uma pontuação, definindo onde começa e termina a informação para uma determinada proteína. A quantidade e organização dos DNAs reguladores e de outros não-codificadores variam muito de uma classe de organismos para a outra, mas a estratégia básica é universal. Dessa maneira, o genoma de uma célula, isto é, todas as informações genéticas contida em sua sequência completa de DNA, comanda não somente a natureza das proteínas da célula, mas também quando e onde elas serão sintetizadas.

Na manutenção e na cópia da informação genética, ocorrem acidentes e erros aleatórios alterando a sequência de nucleotídeos, isto é, dando origem a mutações. Consequentemente, quando uma célula se divide suas duas células-filhas muitas vezes não são idênticas umas às outras, ou à célula parental. Em raras ocasiões, o erro pode representar mudanças para melhor; mas provavelmente, isso não causará uma diferença significativa na perspectiva da célula; em muitos casos, o erro pode acarretar um sério dano, por exemplo, pela interrupção da sequência codificante para uma proteína essencial. As mudanças que ocorrem devido a erros do primeiro tipo tendem a ser perpetuadas, pois a célula alterada tem uma maior probabilidade de se autorreproduzir. As mudanças ocorridas devido a erros do segundo tipo, mudanças seletivamente neutras, podem ser perpetuadas ou não: em uma competição por recursos limitados, será uma

questão de chance o sucesso das células alteradas ou de seus parentes. Porém, as mudanças que causam sérios danos levam a lugar nenhum: as células que sofrem tais mudanças morrem, não deixando progênie. Por meio de intermináveis repetições desse ciclo de tentativas e erros de mutação e seleção natural, os organismos evoluem: suas especificações genéticas mudam, proporcionando novos caminhos para explorar o ambiente de modo efetivo para sobreviver em competições com outros e para se reproduzir com sucesso.

Claramente, algumas partes do genoma mudam com mais facilidade que outras no curso da evolução. Um segmento de DNA que não codifica proteínas e que não tem papel regulador significante esta livre para sofrer mudanças limitadas apenas pela frequência randômica dos erros. Em contraste, um gene que codifica uma proteína essencial ou uma molécula de RNA não pode se alterar tão facilmente: quando ocorrem erros, as moléculas defeituosas são quase sempre eliminadas. Portanto, os genes destes tipos são altamente conservados. Ao longo de 3,5 bilhões de anos ou mais da história evolutiva, muitas características do genoma têm mudado, mas a maioria dos genes altamente conservados permanece perfeitamente reconhecível em todas as espécies vivas.

Os genes altamente conservados são os únicos que devem ser examinados quando desejamos traçar as relações familiares entre os organismos relacionados mais distantemente na árvore da vida. Os estudos que levam à classificação do mundo vivo em três domínios, bactérias, arquebactérias e eucariotos, têm como base, sobre tudo a análise de um dos dois principais componentes do rRNA, o RNA da subunidade menor do ribossomo. Como o processo de tradução é fundamental a todos os organismos vivos, esse componente do ribossomo tem sido bem conservado desde o início da história da vida na Terra.

Vários dos genes dentro de um único organismo mostram fortes semelhanças familiares em suas sequências de DNA, sugerindo que tenham se originado do mesmo gene ancestral por duplicação e divergência gênica. As semelhanças familiares (homologias) são também claras quando sequências gênicas são comparadas entre diferentes espécies, mas 200 famílias de genes altamente conservadas podem ser identificadas como sendo comuns a todas as espécies dos três domínios do mundo vivo. Portanto, dada uma sequência de DNA de um gene descoberto recentemente é possível deduzir a sua função a partir da função de um gene homólogo em um organismo-modelo intensivamente estudado, como a bactéria E. coli.

2.4 Proteínas

A maioria dos genes de uma célula produz moléculas de mRNA que são utilizadas como intermediárias na produção de proteínas. Uma vez que o mRNA tenha sido produzido por meio da transcrição a informação presente em sua sequência de nucleotídeos é usada para sintetizar uma proteína. A transcrição como forma de transferência de informação é de fácil compreensão, uma vez que o DNA e o RNA são químicamente e estrturalmente semelhantes. A conversão da informação de RNA para proteína representa uma tradução da informação para uma outra linguagem que usa símbolos bastante diferentes. Como existem somente quatro diferentes nucleotídeos no mRNA e 20 tipos distintos de aminoácidos em uma proteína, não se pode atribuir nessa tradução uma correspondência direta entre um nucleotídeo no RNA e um aminoácido na

proteína.

A sequência de nucleotídeos em uma molécula de mRNA é lida em grupos consecutivos de três. O RNA é um polímero linear cujos elementos consistem de quatro nucleotídeos diferentes, de tal forma que existem 4 x 4 x 4 = 64 combinações possíveis de três nucleotídeos. Entretanto, somente 20 aminoácidos diferentes normalmente são encontrados nas proteínas. Ou alguns tripletes de nucleotídeos nunca são usados, ou o código é redundante e alguns aminoácidos são determinados por mais de um triplete. Cada grupo de três nucleotídeos consecutivos no RNA é denominado códon, e cada códon especifica um aminoácido, ou a finalização do processo de tradução.

Esse código genético é utilizado universalmente em todos os organismos. Embora algumas pequenas diferenças no código tenham sido encontradas, elas localizam-se principalmente no DNA das mitocôndrias. As mitocôndrias possuem seus próprios sistemas de transcrição e de síntese de proteínas, os quais operam com bastante independência dos sistemas equivalentes do restante da célula. Em princípio, uma sequência de RNA pode ser traduzida em qualquer uma de três fases de leitura diferente, dependendo de onde se inicia o processo de decodificação. Entretanto, somente uma das três possíveis fases de leitura em um mRNA codifica a proteína necessária.

Em uma molécula de mRNA os códons não reconhecem diretamente os aminoácidos que determinam, os grupos de três nucleotídeos, por exemplo, não se ligam diretamente aos aminoácidos. Mas a tradução do mRNA em proteínas depende de moléculas adaptadoras que podem reconhecer e se ligar ao códon e, em outra região de sua superfície, ao aminoácido. Esses adaptadores consistem em um conjunto de pequenas moléculas de RNA conhecido como RNAs transportadores (tRNAs) cada um com tamanho de aproximadamente 80 nucleotídeos.

Todos os tRNAs também são alvo de modificações químicas, aproximadamente um em cada 10 nucleotídeos de uma molécula de tRNA madura é uma versão alterada dos ribonucleotídeos G, U, C ou A padrão. Mais de 50 tipos diferentes de modificações de tRNA são conhecidos. Alguns dos nucleotídeos modificados, mais notadamente a inosina, produzida pela desaminação da adenosina sendo uma enzima envolvida no metabolismo de purinas, que afetam a conformação e o pareamento de bases do anticódon e, assim, facilita o reconhecimento do códon apropriado no mRNA pela molécula de tRNA. Outras afetam a exatidão com a qual o tRNA é ligado ao aminoácido correto.

O mapeamento de cada códon no código genético ao anticódon no DNA, faz com que as células produzam uma série de tRNAs diferentes. Consideraremos agora como cada molécula de tRNA liga-se a um dentre os 20 aminoácidos, o qual é seu parceiro apropriado. O reconhecimento e a ligação ao aminoácido correto depende de enzimas denominadas aminoacil-tRNA-sintetases, as quais acoplam covalentemente cada aminoácido ao seu conjunto apropriado de moléculas de tRNA. Na maioria das células existe uma enzima sintetase diferente para cada aminoácido (ou seja, 20 sintetases); uma enzima liga glicina a todos os tRNAs que reconhecem códons glicina, outra enzima liga alanina a todos os tRNAs que reconhecem códons alanina, e assim por diante.

Diversas bactérias, no entanto, tem menos de 20 sintetases, e uma mesma enzima sintetase é responsável pelo acoplamento de mais de um aminoácido aos seus tRNAs apropriados. Nesses casos uma única sintetase posiciona o aminoácido idêntico em dois tipos diferentes de tRNAs,

mas apenas um deles tem o anticódon que combina com o aminoácido. Uma segunda enzima, então, modifica quimicamente cada aminoácido ligado "incorretamente"de tal forma que este agora corresponda ao anticódon exibido pelo tRNA ao qual ele se encontra covalentemente ligado.

A reação catalizada pela sintetase que liga o aminoácido à extremidade 3' do tRNA é uma das muitas reações celulares associadas à hidrólise de ATP com liberação de energia, produzindo uma ligação altamente energética entre o tRNA e o aminoácido. A energia desta ligação é usada em um estágio posterior, nas sínteses de proteínas, para ligar covalentemente o aminoácido à cadeia polipeptídica em formação.

A síntese de proteínas é guiada pela informação presente nas moléculas de mRNA. Para manter a fase de leitura correta e para assegurar a exatidão (aproximadamente 1 erro a cada 10 mil aminoácidos), assim a síntese proteíca é realizada no ribossomo, uma maquinaria catalítica complexa feita a partir de mais de 50 proteínas diferentes (*as proteínas ribossomais*) e diversas moléculas de RNA, os RNAs ribossomais (rRNAs). Uma célula eucariótica típica contém milhões de ribossomos no citoplasma. As subunidades ribossomais eucarióticas são montadas nos nucléolos pela associação de rRNAs recém-transcritos e modificados com proteínas ribossomais, as quais foram transportadas para o interior do núcleo após sua síntese no citoplasma. As duas subunidades ribossomais são então transportadas para o citoplasma, onde serão unidas para realizar a síntese de proteínas.

Os ribossomos operam com uma eficiência notável, em um segundo um único ribossomo de uma célula eucariótica adiciona aproximadamente 2 aminoácidos à cadeia polipeptídica; os ribossomos das células bacterianas operam ainda mais rapidamente, a taxas de cerca de 20 aminoácidos por segundo. Como o ribossomo organiza os muitos movimentos coordenados necessários para uma tradução eficiente? Um ribossomo contém 4 sítios de ligação para moléculas de RNA: um é para o mRNA e três (denominados sítio A, sítio P e sítio E) são para os tRNAs. Uma molécula de tRNA adere fortemente aos sítios A e P apenas se seus anticódons formam pares de bases com códon complementar (permitindo-se oscilação) na molécula de mRNA que esta ligada ao ribossomo. Os sítios A e P estão suficientemente próximos para que suas duas moléculas de tRNA sejam forçadas a formarem pares de bases com códons adjacentes na molécula de mRNA. Essa característica do ribossomo mantém a fase de leitura correta no mRNA.

A iniciação e terminação da tradução compatilham características com o ciclo de extensão da tradução. O sítio em que a síntese da proteína começa no mRNA é especialmente importante uma vez que ele indica a fase de leitura para todo o comprimento da mensagem. Um erro de um nucleotídeo para mais ou para menos, nesse estágio, fará com que todos os códons subsequentes na mensagem sejam lidos de maneira errada, de tal forma que uma proteína não-funcional, com uma sequência distorcida de aminoácidos, seja produzida. A etapa de iniciação também é importante, uma vez que para a maioria dos genes, é o último ponto no qual a célula pode decidir se o mRNA será traduzido e a proteína será sintetizada; assim, a taxa de iniciação determina a taxa em que a proteína é sintetizada.

A tradução de um mRNA inicia com um códon AUG, e um tRNA especial é necessário para iniciar a tradução. Esse tRNA iniciador sempre carrega o aminoácido metionina (nas bactérias, uma forma modificada de metionina é utilizada: a formilmetionina), portanto todas as bactérias

recém-formadas possuem metionina como seu primeiro aminoácido em suas extremidades N-terminal, a extremidade de uma proteína que é sintetizada primeiro. Após, essa metionina geralmente é removida por uma protease específica. O tRNA iniciador pode ser especialmente reconhecido pelos fatores de iniciação, pois têm uma sequência nucleotídica distinta do tRNA que normalmente carrega a metionina.

O final da mensagem codificadora de uma proteína é sinalizado pela presença de um dos três *códons de terminação* (UAA, UAG ou UGA). Eles são reconhecidos por um tRNA e não determinam um aminoácido; em vez disso, sinalizam para o ribossomo o final da tradução. As proteínas conhecidas como *fatores de liberação* ligam-se a qualquer ribossomo que possua um códon de terminação posicionado no sítio A, e esta ligação força a peptidil-transferase no ribossomo a catalisar a adição de uma molécula de água em vez de um aminoácido no peptidil-tRNA. Essa reação libera a extremidade carboxila da cadeia polipeptídica em crescimento de sua conexão a uma molécula de tRNA. Tendo em vista que apenas esta conexão normalmente mantém unido o polipeptídeo em crescimento ao ribossomo, a cadeia de proteína finalizada é imediatamente liberada no citoplasma. O ribossomo, então, libera o mRNA e separa-se nas duas subunidades grande e pequena, as quais podem associar-se sobre essa mesma ou outra molécula de mRNA para iniciar um novo ciclo de síntese de proteínas.

Nos passos finais da síntese de proteínas, dois tipos distintos de chaperonas moleculares guiam o dobramento das cadeias polipeptídicas. Essas chaperonas, conhecidas como Hsp60 e Hsp70, reconhecem regiões hidrofóbicas expostas nas proteínas e servem para evitar a agregação da proteína que poderia competir com o dobramento das proteínas recentemente sintetizadas em suas conformações tridimensionais corretas. Esse processo de dobramento da proteína deve também competir com um mecanismo de controle de qualidade altamente elaborado que destrói proteínas que contenham as regiões hidrofóbicas expostas. Nesse caso, a ubiquitina é covalentemente ligada a uma proteína erroneamente dobrada por uma ubiquitina-ligase, e a cadeia poliubiquitina resultante é reconhecida pela capa em um proteossomo que move a proteína como um todo para o interior do proteossomo onde sofrerá degradação proteolítica. Um mecanismo proteolítico intimamente relacionado, com base em sinais de degradação especiais reconhecidos pelas ubiquitina-ligases, é utilizado para determinar o tempo de vida de muitas proteínas corretamente dobradas. Através desse método, as proteínas normais selecionadas são removidas da célula em resposta a sinais específicos.

2.5 Splicing Alternativo

As sequências codificantes de genes eucarióticos são caracteristicamente interrompidas por sequências intervenientes não-codificantes (íntrons). Descoberta em 1977, essa característica dos genes eucarióticos foi uma surpresa para os cientistas, que estavam familiarizados, até aquele momento, apenas com genes bacterianos, os quais, caracteristicamente, consistem de uma porção contínua de DNA codificante diretamente transcrita em mRNA. Em contraste extremo, os genes eucarióticos são encontrados sob a forma de pequenos pedaços de sequências codificantes (sequências expressas ou éxons) intercaladas por sequências muito mais longas, as sequências intervenientes ou íntrons; assim a porção codificante de um gene eucariótico é, em geral, apenas

uma pequena fração do comprimento do gene.

Tanto as sequências de íntrons quanto de éxons são transcritas em RNA. Os íntrons são removidos do RNA sintetizado por meio de um processo denominado **splicing de RNA**. Grande parte do splicing de RNA que ocorre nas células atua na produção de mRNA, sendo denominado splicing do precursor de mRNA (ou pré-mRNA). Somente após ter ocorrido o splicing e o processamento das extremidades 5' e 3' esse RNA será denominado mRNA.

O splicing alternativo (AS) é um mecanismo de grande importância para a diversidade proteómica e controle da expressão gênica. Neste processo os genes são justapostos em diferentes arranjos para formação do mRNA maduro, assim o AS é um dos mecanismos responsáveis pelo aumento na capacidade de codificação de genes, sendo encontrado em quase todos os organismos eucarióticos, incluindo animais, plantas e em alguns casos em fungos. As reações de transferências de sequências fosfodiéster envolvidas no splicing alternativo são catalisadas por grandes complexos (macromoléculas) conhecidos por spliceossomo, podendo ser uma das maquinarias mais complexas em uma célula [6], [7], [8], sendo que mesmo decisões aparentemente simples podem ser resultado de uma complexa interação de sinais, como pode ser visto nas Figuras 2.2 e 2.3.

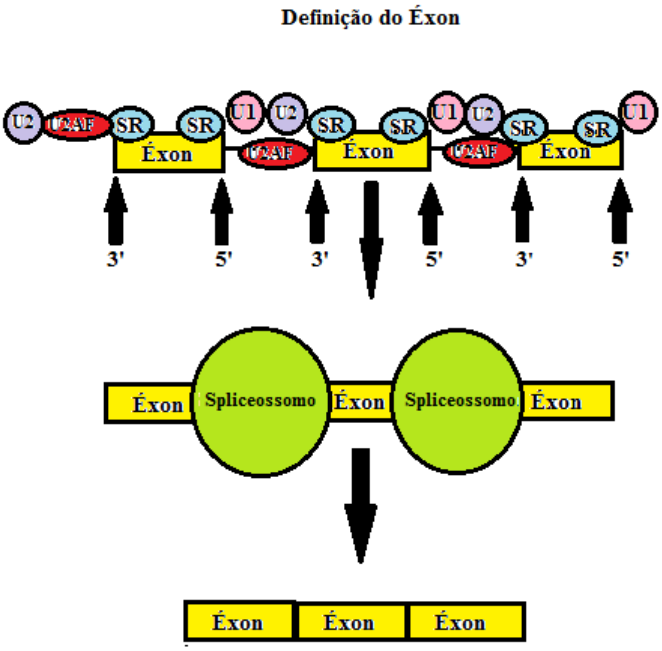

Figura 2.2: Maquinarias do Splicing em relação ao éxon, encontrada em [2]

A maioria dos genes humanos sofre AS, gerando múltiplas isoformas de splicing contendo diferentes combinações de éxons, [9]. O AS amplia a capacidade de codificação de genes, mas também afeta muitos aspectos no metabolismo do RNA incluindo a degradação através da decadência mediada do mRNA e de recrutamento para o ribossomo e eficiência na tradução.

Com os recentes estudos estima-se que, em humanos a proporção de genes que sofre AS pode chegar a mais de 90%, [10], [11], [12], [13], já em plantas o AS era considerado raro, sendo pouco estudado até 2001, limitando-se a alguns genes e a extensão do AS em plantas

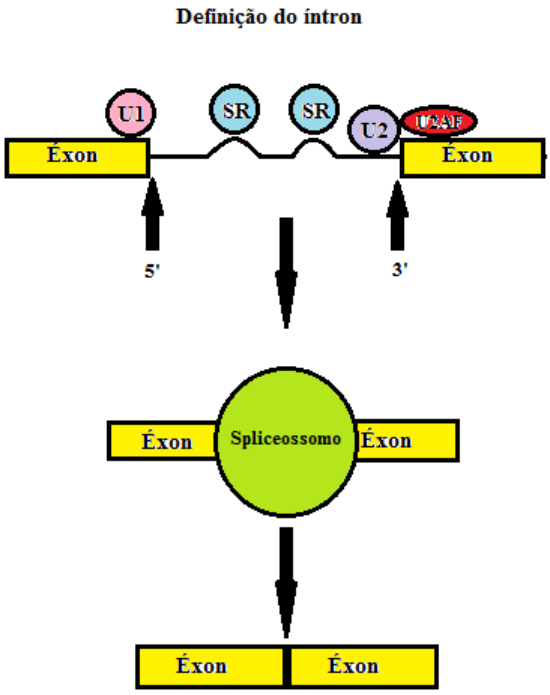

Figura 2.3: Maquinarias do Splicing em relação ao íntron, encontrada em [2]

não era conhecida, [14]. Nas plantas, agora é claro que o AS pode gerar transcriptoma e proteoma com uma grande complexidade, [15]. Para um melhor entendimento sobre a montagem da macromolécula spliceossomo que tem função importante na realização do AS bem como a localização e dinâmica das proteínas, referimos o leitor para as referências, [14] e [16]- [17].

Durante os últimos anos a estimativa de splicing alternativo em Arabidopsis aumentou de cerca de 5% para 22%, [15], [18]- [19]. O número bem menor de ocorrência de splicing em plantas se deve ao pouco estudo dos genes de plantas se comparado aos genes humano ao longo do tempo, de acordo com a evolução dos estudos a tendência é que este número de ocorrências de splicing em plantas aumente.

De acordo com [2], o tipo mais comum de splicing alternativo em plantas é a retenção do íntron com cerca de 56% em Arabidopsis e 53,5% no arroz sendo que este tipo só representa apenas 5% dos casos de splicing em humanos, e em plantas o salto do éxon representa apenas 8% do casos, sendo o caso mais comum em humanos. Os íntrons em genes de plantas são muito mais curtos em comparação com os íntrons de seres humanos, [20], isso talvez explique a grande ocorrência dos casos de retenção de introns no AS de plantas, sendo que os íntrons têm composições diferentes em monocotiledôneas e dicotiledôneas, isso sugere diferenças de splicing nas duas linhagens, [2]. Mesmo com novos estudos o AS em plantas e a sua maquinaria ainda não são bem compreendidas sendo um assunto de grande interesse.

A composição e função do spliceossomo aparece em plantas para ser semelhante ao dos seres humanos, [21], este fato indica que houve uma evolução do spliceossomo em relação aos ancestrais, sendo que nas plantas não se sabe quantas proteínas compõem o spliceossomo, [2]. Uma análise proteômica de spliceossomo de animais purificados resultou na identificação de cerca

de 300 proteínas distintas deste complexo, [6]. Em vegetais não foram isoladas as proteínas que compõem o spliceossomo, uma análise em Arabidopsis revelou a presença de muitas das proteínas encontradas em metazoários [22].

Os tipos mais comuns de splicing alternativos são mostrados na Figura 2.4 encontrados em [2], onde os éxons estão representados por caixas coloridas e os íntrons por linhas horizontais onde os pré-mRNAs estão à direita e os mRNAs à esquerda encontram-se. Cada letra na Figura 2.4 denota um tipo de splicing. A letra (a) mostra o éxon skpping: um éxon está incluído ou excluído do mRNA. A letra (b) mostra éxons mutuamente exclusivos (Mutually exclusive exons): os éxons são unidos de tal maneira que apenas um deles é incluído em algum momento no mRNA. Na letra (c) temos o splicing alternativo no local 5' (Alternative 5' Splice site): diferentes tamanhos de mRNAs são produzidos de acordo com o uso de uma extremidade proximal ou distal 5'. Na letra (d) temos o splicing alternativo no local 3' (Alternative 3' Splice site): diferentes tamanhos de mRNAs são produzidos de acordo com o uso de uma extremidade proximal ou distal 3'. E finalmente na letra (e) temos a retenção do íntron (Intron retention): um íntron é mantido ou excluído no mRNA, resultando em transcritos de diferentes tamanhos, dois ou mais tipos de splicing alternativo pode ocorrer em um único pré-mRNA e a geração de múltiplos mRNA maduro a partir de um único gene.

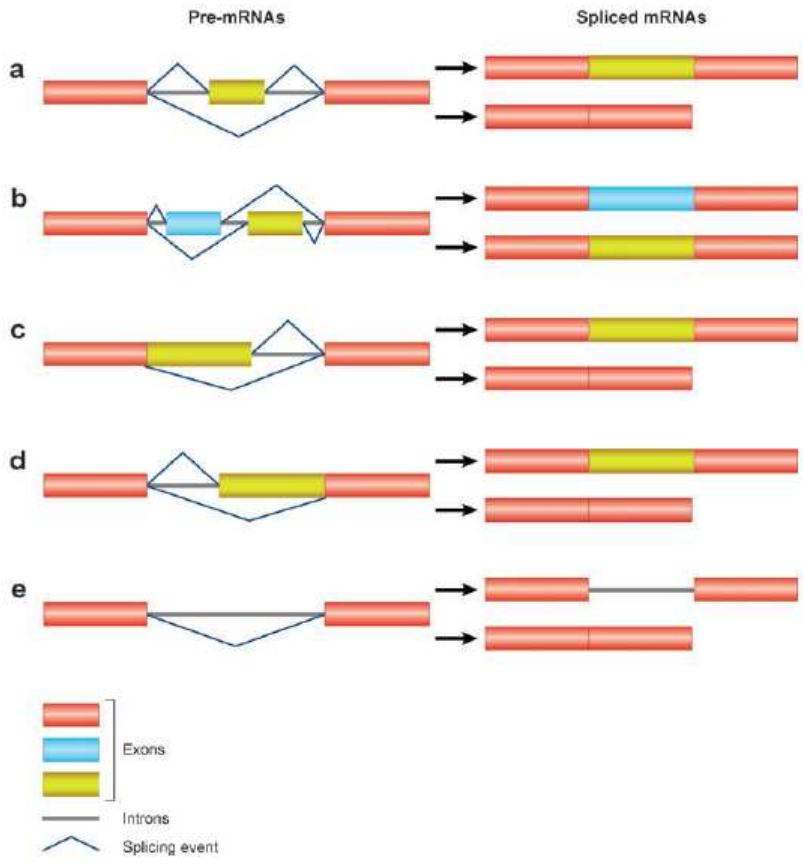

Figura 2.4: Principais tipos de Splicing, encontrados em [2].

Os genes humanos em geral contêm éxons relativamente curtos (Em geral, 50-250 pares de bases (PB)), separados por íntrons muito maiores (Em geral, centenas ou milhares de pares de

base (PB)) [20] [29], esta geometria na transcrição favorece o tipo de splicing conhecido como skipping exon ou salto do éxon. Devido as características do splicing alternativo e a grande ocorrência entre os organismos, estudos com enfoque evolutivo vem se tornando interessante. Assim algumas questões necessitam ser respondidas: como tal mecanismo poderia ter surgido? Quais seriam as diferenças encontradas em grupos distintos? Como o splicing alternativo atuaria na geração da diversidade biológica e especiação?

Capítulo 3

Códigos Corretores de Erros de Substituição, Deleção e Inserção

Os códigos corretores de erros (CCE) são objetos de pesquisa em diversas áreas do conhecimento como: matemática, computação, engenharia elétrica e estatística entre outras. Um CCE é um modo organizado de acrescentar algum dado adicional a cada informação que se queira transmitir ou armazenar e que permita, ao recuperar a informação, detectar e corrigir erros.

Em [23], Hamming propôs a construção de um código capaz de detectar até dois erros e corrigir um erro. A publicação deste trabalho ocorreu com um atraso devido ao pedido de patentes destes códigos, durante o tempo transcorrido desde a elaboração do trabalho até sua publicação Hamming publicou alguns memorandos questionando sobre a possibilidade de criar códigos mais eficientes que aquele proposto inicialmente. Os questionamentos de Hamming são respondidos indiretamente por C. E. Shannon, [24] em 1948, dando início a teoria da informação.

Shannon em seu trabalho demonstrou a existência de códigos corretores de erros tais que a probabilidade de erro seja tão pequena quanto se desejar. Para isso, o comprimento da palavra-código deve crescer, porém mantendo fixa a taxa do código, e esta por sua vez seja menor que a capacidade de canal.

Golay [25], tendo como base o código de Hamming propôs a construção de um código corretor de um único erro cujo comprimento é um primo p. Neste artigo, Golay propôs a construção dos códigos denominados códigos de Golay (23, 12) e (11, 6). Golay, Hamming e Shannon foram os grande pioneiros nas áreas de teoria da codificação e teoria da informação, desenvolvendo estudos e idéias que são usadas até hoje como por exemplo em: comunicações móveis, aparelhos de armazenamentode dados, além de comunicação via satelite e processamento de imagens digitais, etc.

Este capítulo esta organizado da seguinte maneira. Na Seção 3.1, mostramos os CCEs de deleção e inserção, dando enfase aos códigos de Varshamov-Tenengolts para alfabetos q-ário a serem descritos na Subseção 3.1.1. Já na Seção 3.2 faremos uma breve introdução sobre CCEs de substituição, dando enfase aos códigos BCH sobre anéis que são utilizados na identificação de sequências gênicas e genômicas, sendo mostrado em detalhes na Subseção 3.2.6.

3.1 Códigos Corretores de Erros de Deleção e Inserção

De acordo com Sellers [26], em sistemas de comunicação digital é possível que uma mensagem recebida tenha um número diferente de bits (dígitos binários) que a mensagem transmitida. Uma das causas de tal erro é a perda temporária de sincronização entre o transmissor e o receptor. Os problemas atuais com sincronização continuam a ser uma parte integrante dos sistemas que operam em ambientes sob interferência temporais ou aleatórias, [27]. Esses sistemas incluem armazenamento de dados, como gravação magnética e óptica, [28], dispositivos semicondutores e circuitos integrados, [29] e comunicação digital síncrona de redes, [30]. O ruído pode introduzir inserções e deleções de símbolos, e como resultado, os sistemas corrompidos por erros de sincronização não sabem a posição exata no processamento de dados, [27].

Sincronização e ruído aditivo são normalmente tratados como problemas diferentes e portanto fazem uso de técnicas diferentes, [27]. Ambos têm o mesmo efeito sobre os canais de comunicação, isto é, reduzindo a sua capacidade. De acordo com, [31] a hipótese de que os CCEs são capazes de corrigir erros de sincronização poderia melhorar o desempenho global dos sistemas de comunicação, sendo bastante desafiadora a concepção, o que explica em parte por que uma grande coleção de técnicas de sincronização não baseada em codificação foram desenvolvidas e implementadas ao longo dos anos. Canais corrompidos por erros de sincronização tem memória, daí as técnicas desenvolvidas para canais sem memória e com ruído aditivo raramente podem ser usadas diretamente, [27].

De acordo com, [27] as ferramentas desenvolvidas para os códigos corretores de erros contra erros de temporização podem também ser de interresse para uma série de problemas que podem ser resolvidos por meio de modelos de sincronização, como por exemplo reconhecimento de padrões, [32]. Quando os códigos com símbolos de comprimento variável são usados para transmitir informação, o ruído aditivo pode causar erros de sincronização alterando os símbolos terminais, [27]. As inserções e deleções podem também ocorrer para uma grande classe de problemas distribuídos envolvendo recombinação de dados correlacionados, [33] tais como recombinação de nucleotídeo em sequências de moléculas de DNA, [34] o armazenamento de dados remoto, [35] e sincronização de dados móveis, [36].

Shannon [24], mostrou que a informação pode ser codificada e transmitida de forma confiável na presença de ruído, em qualquer taxa inferior à capacidade do canal. Shannon mostrou a existência de bons códigos, desde então muitas pesquisas foram e continuam sendo realizadas, com o intuito de construir bons códigos corretores de erros com algoritmos eficientes de codificação e principalmente de decodificação, sendo utilizado hoje códigos confiáveis e eficientes em uma grande variedade de sistemas digitais. Para um melhor entendimento sobre códigos corretores de erros e suas aplicações, referimos o leitor para as referências, [37], [38] e [39].

De acordo com, [27], a grande maioria dos códigos corretores de erros pressupõe que o transmissor e receptor estejam sincronizados. Em particular, o receptor sabe onde e quando a mensagem recebida inicia e termina sendo que os símbolos transmitidos e recebidos têm o mesmo comprimento ou duração, a redundância introduzida pelos códigos é usada para corrigir símbolos corrompidos pelo canal. Todos os códigos pesquisados consideram símbolos discretos, bem como erros de sincronização discretos. Em alguns casos práticos os símbolos discretos são inadequados para a transmissão ou armazenamento da informação, e os erros de sincronização

podem ser pequenas frações do comprimento ou duração de um símbolo.

Um erro de sincronização é equivalente à um erro de deleção ou um erro de inserção excluindo-se os erros de substituição. Além dos desafios comuns que enfrentam os projetistas de códigos como a construção de códigos com boas propriedades de distância e algoritmos de decodificação, os erros de sincronização introduzem dificuldades que não ocorrem em outras classes de erros. Uma delas é que um único erro de sincronização não corrigido pode ter consequências catastróficas, causando uma enorme rajada de erros de substituição com duração até o sistema ser sincronizado novamente, [27]. Em grande parte dos sistemas de comunicação, as mensagens longas são divididas em blocos, portanto, um outro desafio introduzido por erros de sincronização é que os limites dos blocos podem ser desconhecidos para o receptor.

Os códigos de Varshamov-Tenengolts, [40] consistem de vetores binários de comprimento n capazes de corrigir um erro de inserção ou deleção.

Os códigos corretores de erros de sincronização binários foram estudados primeiramente por Levenshtein em, [41]. Percebendo que os códigos de Varshamov-Tenengolts, originalmente construídos para corrigir um erro assimétrico, eram também assintoticamente ótimos para a correção de um erro sincronização, Ullman, [42], apresentou de forma independente, uma família de códigos ligeiramente diferente e com maior redundância capaz de corrigir um erro de sincronização.

O sistema de congruência está por trás de uma grande parcela do trabalho algébrico em corrigir os erros de sincronização. Em, [41] é mostrado que os códigos, consistindo de palavras-código de comprimento n satisfazendo a congruência de Varshamov-Tenengolts com $m = 2n$ podem corrigir um erro de sincronização ou um erro de substituição. Tenengolts mostra em, [43] que códigos podem corrigir um erro de substituição, imediatamente seguido por um erro de exclusão, usando uma família semelhante de congruência.

Infelizmente, o código de Varshamov-Tenengolts construído para corrigir um único erro de sincronização não pode ser generalizado para a correção de vários erros de sincronização, [27]-[44]. Esta construção foi utilizada na extensão de até cinco erros de inserção e deleção, todovia a perda na taxa é significativa não propiciando uma codificação e nem uma decodificação eficientes. Portanto, não sendo uma boa técnica para utilização em blocos com grande comprimentos. Códigos corretores de erros de sincronização com um alfabeto não binário foram primeiramente estudados em [45]- [46]. Tenengolts em [47], generaliza o código corretor de erros de sincronização proposto por Levenshtein para alfabetos não binários, sendo mostrado em detalhes na Subseção 3.1.1.

Levenshtein provou que todo código binário corretor de sincronização cujas palavras-código satisfazem a congruência de Varshamov-Tenengolts mostradas nas Relações 3.2 e 3.3 são códigos perfeitos de correção de deleção. Usando sistemas ordenados de Steiner, [48] foram construídos códigos corretores de deleção perfeitos de comprimento três e com qualquer tamanho de alfabeto. Nos trabalhos, [49]- [50] foi demonstrado que os códigos Reed-Solomon generalizados podem ser usados para corrigir deleções e decodificado em tempo polinomial usando o algoritmo de decodificação de lista, [51].

Alguns códigos foram concebidos para corrigir rajadas de erros de sincronização. O primeiro tipo de construção surgiu a partir da proposta de Levenshtein, [52] que utiliza uma família

de congruências semelhante a introduzida nas Relações 3.2 e 3.3, construindo assintoticamente códigos ideais capazes de corrigir duas eliminações consecutiva de erros. Iizuka, Kasahara e Namekava, [53] propuseram códigos que podem corrigir uma rajada de erros de substituição, bem como uma rajada de erros de inserção ou deleção que ocorrem na rajada de erros de substituição. Iwamura e Imai, [54], construiram um código que divide a sequência de informação em k segmentos de q bits, e pode corrigir um erro de sincronização e uma rajada de erros de substituição desde que todos os erros estejam localizados no mesmo segmento.

De acordo com, [55]- [56], um código "comma-free"sobre um alfabeto A é um conjunto $C \subseteq A^*$ de palavras sobre A tal que dadas quaisquer duas palavras $w, v \in C$, qualquer sub-palavra, u, da concatenação, wv, não está no código. No trabalho de Bours, [57], foram construídos códigos que podem corrigir pequenas rajadas de erros de deleção ou inserção. Sua construção é um código de matriz de duas dimensões, em que as linhas da matriz são palavras-código de um código "comma-free"usado para recuperar a sincronização, e as colunas da matriz são palavras-código de um código Reed-Muller (32, 16, 8), utilizado para corrigir substituição de erros, bem como apagamentos causados pela perda temporária de sincronização.

3.1.1 Códigos de Varshamov-Tenengolts

Depois de uma breve introdução sobre os CCEs de deleção e inserção para alfabetos binários, vamos considerar nesta seção como é realizada a correção de inserção ou deleção em códigos com alfabetos q-ários, sendo este tópico baseado no artigo de Tenengolts, [47], em que é demonstrado através de uma relação de congruência a forma para correção de um único erro de deleção ou inserção.

Dada uma sequência não binária a_1, a_2, ...,a_n onde $a_i \in \{0, 1, ..., q-1\}$ e associando uma sequência binária α_1, α_2, ..., α_n pela Relação 3.1.

$$\alpha_i = \begin{cases} 1 & \text{se } a_i \geq a_{i-1}; \\ 0 & \text{se } a_i < a_{i-1}. \end{cases} \tag{3.1}$$

O α_1 pode ser qualquer símbolo binário, porém consideraremos α_1=1. Assim, um sistema de congruência, dado pelas Relações 3.2 e 3.3, onde β e γ são inteiros fixos arbitrários e n é o comprimento do código, é formado.

$$\sum_{i=1}^{n} a_i \equiv \beta (mod\, q) \tag{3.2}$$

$$\sum_{i=1}^{n} (i-1)\alpha_i \equiv \gamma (mod\, n) \tag{3.3}$$

Teorema 3.1 *[47] O conjunto de sequências q-árias a_1, a_2, ...,a_n, tal que as sequências binárias que estão associadas pela Relação 3.1 satisfazem os sistemas de congruência mostrada nas Relações 3.2 e 3.3, é um código que corrige uma única deleção ou inserção.*

Vamos considerar o caso de eleminação, como resultado da deleção de um único símbolo, a sequência a'_1, a'_2, ..., a'_{n-1} estará sendo recebida no receptor. É a partir desta sequência q-ária que se determina uma sequência binária associada α'_1, α'_2, ..., α'_{n-1} pelas Relações 3.2 e 3.3. Calculamos os parâmetros W, S_1 e S_2, em que W é o peso (número de símbolos diferentes de zero) da sequência α'_1, α'_2, ..., α'_{n-1} e S_1 e S_2 são os menores resíduos não negativos das congruências:

$$S_1 \equiv \beta - \sum_{i=1}^{n} a'_i (mod\, q) \tag{3.4}$$

e

$$S_2 \equiv \gamma - \sum_{i=1}^{n} (i-1)\alpha'_i (mod\, n) \tag{3.5}$$

Vamos mostrar que os parâmetros W, S_1 e S_2 permitem uma única decodificação. S_1 é igual ao valor do símbolo perdido, a localização do símbolo perdido pode ser encontrado da seguinte maneira: sequências binárias α_1, α_2, ..., α_n que satisfazem a segunda congruência do sistema mostrado nas Relações 3.2 e 3.3, constituem um código binário simples que corrige deleção. Portanto, podemos restaurar exclusivamente a sequência α_1, α_2, ..., α_n a partir da sequência α'_1, α'_2, ..., α'_{n-1}.

Se o símbolo α_1=1 foi perdido, então S_2=$i-1+n_i$=$w+n_0 \geq W$, em que todo, n_i indica o número de uns do lado direito do bit perdido e não é o número de zeros no lado esquerdo dos uns perdidos. Se o símbolo α_i=0 foi perdido, então S_2=n_i<w, assim α'_i=1. Analisando S_2 e W podemos determinar com exclusividade que um dos símbolos binários (0 ou 1) foi perdido.

Se $S_2 \geq W$ então na sequência α'_1, α'_2, ..., α'_{n-1} inserimos o símbolo "1"de modo que o número de zeros do lado esquerdo do qual o símbolo foi inserido seja igual a $S_2 - W$. No caso em que $S_2 < W$, inserimos o símbolo "0"na sequência α'_1, α'_2, ..., α'_{n-1} de modo que o número de uns do lado direito do onde o símbolo foi inserido seja igual a S_2.

De acordo com a Relação 3.1 com a perda do símbolo q-ário a_i, perdemos um símbolo binário corresponde na sequência α_1, α_2, ..., α_n. Ainda de acordo com o mapeamento da Relação 3.1 uma sequência α_1, α_2, ..., α_n correspondente a um conjunto monotonicamente decrescente ou diminuído a sequência de símbolos a_1, a_2, ..., a_n.

No caso de inserção da sequência recebida a'_1, a'_2, ..., a'_{n+1}, podemos determinar a sequência binária associada α'_1, α'_2, ..., α'_{n+1} pela Relação 3.1 e calcular os parâmetros W, S_1 e S_2, em que W é o peso da sequência α'_1, α'_2, ..., α'_{n+1} e S_1 e S_2, são os resíduos não negativos das congruências.

$$S_1 \equiv \sum_{i=1}^{n+1} a'_i - \beta (mod\, q) \tag{3.6}$$

e

$$S_2 \equiv \sum_{i=1}^{n+1} (i-1)\alpha'_i - \gamma (mod\, n) \tag{3.7}$$

S_1 é igual ao valor do símbolo inserido, a sequência binária correspondente à sequência α_1, α_2, ..., α_n, pode ser restaurada exclusivamente a partir da sequência α'_1, α'_2, ..., α'_{n+1} da seguinte

maneira. Note que α'_1=1, se S_2=0, então eliminamos o último símbolo da sequência α'_1, α'_2, ..., α'_{n+1}. Se $0 < S_2 < W-1$, então eliminamos qualquer zero de modo que o número de uns à direita deste símbolo na sequência α'_1, α'_2, ..., α'_{n+1} é igual a S_2. No caso quando $S_2 = W-1$ eliminamos o segundo símbolo da sequência α'_1, α'_2, ..., α'_{n+1}. Se $S_2 > W-1$ então eliminamos qualquer símbolo 1 de modo que o número de zeros do lado direito deste símbolo seja igual a $n - S_2$.

Tal como a deleção, de acordo com as Relações 3.2 e 3.3 o símbolo inserido q-ário a'_i correspondente a um símbolo binário da sequência α'_1, α'_2, ..., α'_{n+1} que está localizado tanto no limitante a partir do qual se excluiu o símbolo binário ou na execução anterior. O valor do símbolo inserido é igual a S_1, sabendo disso o valor é novamente levado em consideração, a sequência que segue α_1, α_2, ..., α_n corresponde a um conjunto monotonicamente decrescente (ou diminuído) da sequência de símbolos a_1, a_2, ..., a_n, podendo determinar com exclusividade a sequência a_1, a_2, ..., a_n.

Disso segue que uma sequência q-ária é um código que corrige uma única deleção ou inserção. Um outro teorema que mostra o limitante superior para a cardinalidade do código q-ário que corrige uma única deleção ou inserção será apresentado a seguir. Seja o código C q-ário ideal de comprimento n (isto é, o código com o maior número de palavras-código possíveis) que corrige uma única deleção. Vamos denotar a cardinalidade de C por $M(q,n)$.

Teorema 3.2 *[47] Para um q fixo e $n \to \infty$, então*

$$M(q,n) \lesssim \frac{q^n}{\langle (q-1)n \rangle} \tag{3.8}$$

3.2 Códigos Corretores de Erros de Substituição

Nesta seção consideraremos os CCEs de substituição, mostrando suas características e aspectos matemáticos envolvidos na codificação e decodificação. Vamos mostrar através de exemplos alguns tipos de códigos, com o objetivo de facilitar o entendimento do algoritmo de identificação de sequências de DNA, usando códigos BCH sobre anéis. Nas Subseções 3.2.1 e 3.2.2 apresentaremos uma breve introdução das principais definições e propriedades das estruturas algébricas de anéis e corpos. Na Subseção 3.2.3, revisaremos os conceitos relacionados a códigos de bloco e suas principais características, na Subseção 3.2.4 apresentaremos uma revisão sobre códigos lineares e na Subseção 3.2.5 mostraremos os conceitos relacionados aos códigos cíclicos. Finalmente na Subseção 3.2.6 apresentaremos os conceitos e propriedades de códigos BCH sobre anéis e suas extensões de Galois.

3.2.1 Anéis

A estrutura de anel é parte fundamental na teoria de CCEs, facilitando os processos de codificação e decodificação e análise de desempenho. Os conceitos e definições apresentados nesta subseção podem ser encontrados em [58], [59], [3], [4].

Definição 3.2.1 *Um anel $\langle R, +, \cdot \rangle$ é um conjunto não vazio R juntamente com duas operações binárias + e · definidas sobre R, as quais chamamos de adição e multiplicação, tal que os seguintes axiomas são satisfeitos:*

1. $\langle R, + \rangle$ é um grupo abeliano;

2. A operação de multiplicação é associativa, isto é, (ab)c=a(bc), ∀ a, b, c ∈ R;

3. Para todo a, b, c ∈ R, é válida a lei distributiva à esquerda, a(b+c)=(ab)+(ac) e a lei distributiva à direita, (a+b)c=(ac)+(bc).

Definição 3.2.2 *Se a e b são elementos não nulos de um anel R tais que ab=0 ou ba=0, então a e b são divisores de zero.*

Definição 3.2.3 *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 pela esquerda, tais que para todo $\alpha, \beta, \in G$ e $r, s, \in R$, as seguintes condições são satisfeitas:*

1. $(r\alpha) \in G$;

2. $r(\alpha + \beta) = r\alpha + r\beta$;

3. $(r + s)\alpha = r\alpha + s\alpha$;

4. $(rs)\alpha = r(s\alpha)$.

3.2.2 Corpos algébricos de Galois

A estrutura de corpo é importante na teoria de CCEs pois, facilita os processos de codificação e decodificação bem como a análise de desempenho. Os conceitos e definições apresentados nesta subseção podem ser encontrados em [58], [59], [3], [4].

Definição 3.2.4 *Um corpo é um anel comutativo com unidade e tal que todo elemento não-nulo é inversível.*

Assim, podemos dizer que F é um corpo sob as operações binárias + e · se, e somente se, F constitui um grupo abeliano sob estas operações e, para a operação ·, é válida a lei distributiva. Portanto, um corpo apresenta no mínimo dois elementos: as identidades das operações + e ·. O número de elementos num corpo é a ordem do mesmo e um corpo onde este número é finito é chamado corpo finito.

Teorema 3.3 *As classes residuais de polinômios módulo um polinômio f(x) de grau n formam uma álgebra de dimensão n sobre o corpo dos coeficientes.*

Teorema 3.4 *Seja p(x) um polinômio com coeficientes em um corpo F. Se p(x) for irredutível em F, i. e., se p(x) não possuir fatores com coeficientes em F, então a álgebra de polinômios sobre F módulo p(x) será um corpo.*

Os corpos finitos são usados na maioria das construções dos códigos conhecidos, estes corpos são também conhecidos como corpos algébricos de Galois ou corpos de Galois e são denotados por $GF(q)$ ou F_q onde $q \geq 2$ é o número de elementos do corpo.

O corpo formado por polinômios sobre um corpo F módulo um polinômio irredutível $p(x)$ de grau r é chamado corpo de extensão de grau r sobre F.

Teorema 3.5 *Seja F^* o conjunto dos $q-1$ elementos não-nulos de $GF(q)$, onde $q = p^r$. Então, F^* é um grupo cíclico multiplicativo de ordem p^{r-1}.*

Definição 3.2.5 *Um polinômio de grau $n-1$ sobre um corpo 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, $0 \leq i \leq n-1$, são elementos de F_q.

Definição 3.2.6 *Um polinômio mônico é aquele cujo coeficiente líder (coeficiente da variável com maior expoente) p_{n-1} é igual a 1, a identidade multiplicativa de F_q.*

É conhecido que 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 $F_q[x]$.

Definição 3.2.7 *Um elemento $\beta \in F_q$ é uma raiz ou zero do polinômio $p(x) \in F_q[x]$ se $p(\beta) = 0$.*

Teorema 3.6 *Se G é um subgrupo multiplicativo do grupo $(F^*, \cdot)$ de elementos não nulos de um corpo F, então G é cíclico.*

Teorema 3.7 *O anel de polinômios módulos um polinômio $p(x)$ sobre F_q é um corpo se, e somente se, $p(x)$ é um polinômio primo.*

Definição 3.2.8 *Um gerador do grupo multiplicativo de F_q é denominado um elemento primitivo de F_q.*

Corolário 3.2.1 *Todo corpo finito F_q contém um elemento primitivo.*

Uma consequência imediata do Corolário 3.2.1 é a de 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.

Definição 3.2.9 *Seja $GF(q')$ um corpo finito e $GF(q)$ um subcorpo de $GF(q')$. Seja $\beta \in GF(q')$. O polinômio primo $p(x)$ de menor grau sobre $GF(q)$, tal que $p(\beta) = 0$, é chamado polinômio minimal de β sobre $GF(q)$.*

Teorema 3.8 *Considere os corpos $GF(q')$ e $GF(q)$ como definidos acima. Cada elemento β de $GF(q')$ tem um único polinômio minimal sobre $GF(q)$. 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 $g(x)$.*

3.2.3 Códigos de bloco

A principal característica dos códigos de bloco é a ausência de memória. As definições e teoremas mostrados nesta subseção podem ser encontrados em, [58], [59], [3], [4], [60]. Começaremos nosso estudo pelo conjunto A, que pode ser finito ou infinto, chamado alfabeto.

Definição 3.2.10 *Um código C sobre um alfabeto A é qualquer subconjunto não-vazio do espaço de sequências A^I, onde A é chamado alfabeto do código e I é o conjunto de índices das sequências $c = \{c_i | i \in I\}$. Chamamos de palavra-código os elementos, ou símbolos, no alfabeto A que compõem o código C.*

A partir dessa definição, identificamos o alfabeto A com os elementos do corpo F_q. O codificador para um código de bloco divide a sequência de informação em blocos de k símbolos, onde cada um desses blocos é representado por uma k-upla $u = (u_1, \cdots, u_k)$ chamada mensagem. Assim existe um total de q^k mensagens diferentes. Após a divisão da sequência de informação, o codificador transforma cada mensagem u em uma n-upla $V = (v_1, \cdots, v_n)$ de símbolos discretos chamada palavra-código. Se cada uma das q^k mensagens distintas é tranformada em uma palavra-código, então existem também q^k palavras-código diferentes.

Neste trabalho, estamos interessados em alfabetos finitos, sendo conveniente que o mesmo seja "estruturado". Entendemos "estruturados"como sendo aqueles que formam alguma estrutura algébrica de anel, corpo ou grupo. Quando isto acontece o conjunto formado pelas q^k palavras-código de comprimento n é chamado código de bloco.

Definição 3.2.11 *Um código de bloco C de comprimento n sobre um alfabeto A é qualquer subconjunto A^n das sequências $c = \{c_i | 1 \leq i \leq n\}$.*

Um código de bloco é caracterizado por três parâmetros principais: seu comprimento, sua dimensão e sua distância mínima.

Definição 3.2.12 *A dimensão de um código C é dada por*

$$k = \log_{|A|} |C| \tag{3.9}$$

onde $|\cdot|$ denota a cardinalidade do conjunto.

Definição 3.2.13 *Seja C um código de comprimento n tal que $|C| \geq 2$. A distância mínima de Hamming de C, denotada por $d_{min}(C)$ é dada por:*

$$d_{min}(C) = \min_{x, y \in C, x \neq y} d(x, y). \tag{3.10}$$

A distância d utilizada na caracterização do código depende da métrica utilizada no alfabeto em questão. Assim, um código de bloco C de comprimento n, dimensão k e distância mínima de Hamming $d = d_{min}(C)$ é representado por (n, k, d_{min}). O teorema a seguir fornece um limitante superior para a distância mínima em função dos parâmetros n e k.

Teorema 3.9 *(Desigualdade de Singleton) Para qualquer código de bloco (n, k, d_{min}), vale a seguinte desigualdade:*

$$d \leq n - k + 1. \tag{3.11}$$

Um outro parâmetro muito importante na caracterização de um código de bloco, indicador de desempenho deste, é a chamada taxa do código, definida pela razão entre a dimensão do código e seu comprimento, ou seja,

$$r_C = \frac{k}{n}. \tag{3.12}$$

Códigos de bloco podem ser usados como CCE com a capacidade de correção de erros de um código (n, k, d), denotada por t, esta relacionada a distância mínima deste código da seguinte forma:

$$d_{min} \leq 2t + 1. \tag{3.13}$$

Portanto, quanto maior a distância mínima do código, maior é a capacidade deste código de corrigir erros. Em geral, bons códigos são longos e, por isso, torna-se impraticável descrevê-los através de listas de palavra-código. Para facilitar o problema, o caminho usual é associar aos códigos estruturas matemáticas que facilitem a execução das operações de codificação e decodificação. Assim, a principal classe dos códigos de bloco é a dos códigos lineares.

3.2.4 Códigos lineares

As definições contidas nesta subseção podem ser encontrados em [3] e [4]. Os códigos conhecidos até hoje em sua maioria pertecem à classe dos códigos lineares. Um código (n, k, d_{min}) é dito linear se, e somente se, todas as suas palavras-código formam um subespaço vetorial de dimensão k do espaço vetorial F_q^n, o conjunto das n-uplas do corpo F_q. Assim, podemos representar este código matricialmente como

$$G = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_n \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{bmatrix} \tag{3.14}$$

conhecida como matriz geradora do código (n, k, d_{min}), cujas linhas formam uma base do código linear C. Portanto, o processo de codificação pode ser escrito como:

$$v = u.G, \tag{3.15}$$

onde u é a mensagem a ser codificada ou informação e v é a palavra-código correspondente. Para toda palavra-código v vale a seguinte relação

$$v.H^T = 0, \tag{3.16}$$

onde a matriz $(n-k) \times n$, denotada por H, é chamada matriz verificação de paridade de C, e qualquer vetor ortogonal a suas linhas pertence ao espaço vetorial das linhas da matriz geradora G associada e vice-versa. O código gerado pela matriz H é chamado código dual do código C, denotado por $C^{\perp}$.

Dada uma matriz geradora na forma sistemática, existe uma maneira simples de determinar uma matriz verificação de paridade. Se C é o espaço linha da matriz $G = (I_k|P)$, então C é o espaço ortogonal de $H = (-P^T|I_{n-k})$, onde I_{n-k} é a matriz identidade de ordem $n-k$ e P^T é a matriz transposta de P.

Definição 3.2.14 *Dado um código C com matriz verificação de paridade H, a síndrome de um vetor $v \in \mathbb{F}_q$ é o vetor $v.H^T = s$.*

A síndrome é um conceito usado para fazer a correção de erros em códigos lineares. A expressão padrão de erro denomina a diferença entre a palavra-código recebida e a palavra-código enviada. Em um código linear C com parâmetros (n, k), considere um padrão de erro $e \in \mathbb{F}_q^n$. Como C é um subgrupo, então $e + C = \{e + v | v \in C\}$ é uma classe lateral de $\mathbb{F}_q^n$. Assim estabelecemos uma tabela da seguinte maneira: a primeira linha da tabela deve conter todas as palavras-código de C começando com a palavra toda nula; das n-uplas de F_q^n que não foram usadas, escolha aquela com menor peso e chame-a de e_1. A segunda linha da tabela ser a composta pela classe lateral $e_1 + C$; a j-ésima linha da tabela é formada pela classe $e_j + C$, onde e_j é sempre escolhido como a n-upla em F_q^n de menor peso que ainda não foi usada; esse procedimento termina quando todas as palavras de F_q^n tenham sido usadas.

Usando o procedimento descrito acima temos a Tabela 3.1 chamada de arranjo padrão, Algumas observações importantes devem ser feitas sobre o arranjo padrão. Cada palavra aparece uma única vez na tabela. Duas palavras estão na mesma classe lateral se, e somente se, possuem a mesma síndrome. A primeira coluna da tabela é formada pelas palavras de peso mínimo dentro de cada classe, e são denominadas os líderes das classes laterais.

Tabela 3.1: Arranjo padrão.

$v_1 = 0$	v_2	v_3	$\cdots$	v_q^k
e_1	$e_1 + v_2$	$e_1 + v_3$	$\cdots$	$e_1 + v_q^k$
e_2	$e_2 + v_2$	$e_2 + v_3$	$\cdots$	$e_2 + v_q^k$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$e_{q^{n-k}}$	$e_{q^{n-k}} + v_2$	$e_{q^{n-k}} + v_3$	$\cdots$	$e_{q^{n-k}} + v_q^k$

Uma regra de decodificação por máxima verossimilhança para um código linear é completamente descrita pelo arranjo padrão. O receptor utiliza o arranjo padrão para decodificar uma palavra recebida da seguinte maneira: recebido v, calcule sua síndrome; ache o padrão de erro e correspondente a essa síndrome na tabela; $v - e$ é a palavra-código.

Para um código (n, k) sobre F_q, uma lista completa consiste de q^n palavras. Como cada linha na tabela do arranjo padrão contém q^k elementos então o número de classes laterais será q^{n-k}. Note que para valores grandes de n e k a utilização do arranjo padrão se torna um trabalho impraticável.

3.2.5 Códigos cíclicos sobre $\mathbb{Z}_q$

Nesta subseção, vamos apresentar definições e teoremas relacionados a códigos cíclicos sobre anéis $\mathbb{Z}_q$ ($q \geq 4$ e inteiro). Com isso, teremos uma base para o desenvolvimento da construção do código BCH sobre as estruturas algébricas de anéis e corpos e suas extensões de Galois, sendo adotada nesta subseção como referências os trabalhos, [3], [4], [38], [39] e [61].

Definição 3.2.15 *[3] Seja R um anel. Um módulo livre é um R-módulo gerado por um conjunto de vetores linearmente independentes.*

Definição 3.2.16 *[61] 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 3.2.17 *[61] Um código linear C com parâmetros (n, k) sobre $\mathbb{Z}_q$ é cíclico se, para $v = (v_0, v_1, v_2, \ldots, v_{n-1}) \in C$, todo deslocamento cíclico $v^{(1)} = (v_{n-1}, v_0, v_1, v_2, \ldots, v_{n-2}) \in C$, com $v_i \in \mathbb{Z}_q, 0 \leq i \leq n-1$.*

Geralmente os códigos cíclicos são representados na forma polinomial. Assim, considere a palavra-código $v = (v_0, v_1, v_2, \ldots, v_{n-1})$ de um código cíclico C. Podemos representá-la pelo polinômio:

$$v(x) = v_0 + v_1 x + v_2 x^2 + \ldots + v_{n-1} x^{n-1}. \tag{3.17}$$

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

$$v^{(1)}(x) = v_{n-1} + v_0 x + v_1 x^2 + \ldots + v_{n-2} x^{n-1}, \tag{3.18}$$

que corresponde à palavra-código

$$\underline{v}^{(1)} = (v_{n-1}, v_0, v_1, \ldots, v_{n-2}), \tag{3.19}$$

sendo esta um deslocamento cíclico da palavra:

$$\underline{v} = (v_0, v_1, v_2, \ldots, v_{n-1}). \tag{3.20}$$

Portanto, $v^{(1)}(x)$ é obtido através do produto $x.v(x)$ no anel quociente $R_n = \frac{\mathbb{Z}_q[x]}{\langle x^n - 1 \rangle}$, onde $\langle x^n - 1 \rangle$ representa o ideal gerado por $x^n - 1$. A adição de duas palavras-código é feita em $\mathbb{Z}_q[x]$.

Note que o conjunto de todas as palavras pertencentes a um código cíclico C forma um subconjunto do anel R_n, isto é, o conjunto de todos os polinômios cujo grau é menor que n.

Teorema 3.10 *[61] Um conjunto S de elementos em R_n é um código cíclico se, e somente se, S é um ideal em R_n.*

Proposição 3.2.1 *[61] Seja C um ideal em $R_n = \frac{\mathbb{Z}_q[x]}{\langle x^n - 1 \rangle}$, isto é, um código cíclico de comprimento n. Se existir um polinômio de grau mínimo em C, cujo coeficiente dominante é um elemento inversível em $\mathbb{Z}_q$, então o polinômio mônico (ou seja, aquele cujo coeficiente dominante é um) de grau mínimo em C é único.*

Teorema 3.11 *[61] Seja C um ideal em $R_n = \frac{\mathbb{Z}_q[x]}{\langle x^n-1\rangle}$ e $g(x)$ um polinômio mônico com o menor grau em C. Assim, $C = \langle g(x)\rangle$, e portanto, o código C consiste de todos os múltiplos de $g(x)$. Dizemos então que C é um ideal principal.*

Teorema 3.12 *[61] Seja C um ideal em R_n. Se o coeficiente dominante do polinômio de menor grau em C, $g(x)$, é um elemento inversível, então $g(x)$ divide $(x^n - 1)$. Note que se este polinômio for mônico, então $g(x)$ divide $(x^n - 1)$.*

O Teorema 3.12 fornece um método de construção de códigos cíclicos sobre anéis de inteiros residuais análogo ao método de construção de códigos cíclicos sobre corpos finitos, ou seja, através da fatoração do polinômio $(x^n - 1)$ sobre o anel de interesse para então tomar um fator (ou produto de fatores) como polinômio gerador do código em questão. O Teorema 3.13, mostrado a seguir, está relacionado a representação matricial dos códigos cíclicos sobre anéis que possuem uma matriz geradora.

Teorema 3.13 *[61] Se $g(x)$ divide $(x^n - 1)$ e o grau de $g(x)$ é $(n - k)$, então a dimensão de $C = \langle g(x)\rangle$ é k. Se*

$$g(x) = g_0 + g_1x + g_2x^2 + \ldots + x^{n-k} \tag{3.21}$$

então a matriz geradora do código C é dada por:

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k-1} & 1 & 0 & \cdots & 0 \\ 0 & 0 & g_0 & \cdots & g_{n-k-2} & g_{n-k-1} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & g_0 & g_1 & g_2 & \cdots & 1 \end{bmatrix} \tag{3.22}$$

3.2.6 Códigos BCH sobre anéis e corpos

Os códigos BCH foram propostos por R. C. Bose, D. K. Chaudhuri e A. Hocquenghem e representam uma excelente generalização dos códigos de Hamming, permitindo a múltipla correção de erros. Os códigos BCH formam uma importante classe de códigos cíclicos devido a sua simplicidade nos processos de codificação e decodificação, sendo uma das melhores classes de códigos construtivos para canais onde os erros afetam os símbolos de forma independente.

Uma deficiência apresentada pelo código BCH é que assintoticamente a capacidade de correção de erros não cresce na mesma proporção que o comprimento da palavra-código. A seguir, faremos algumas considerações sobre extensões de anéis e corpos de Galois e, em seguida, sobre os códigos BCH e por fim mostraremos o algoritmo de codificação genética e genômica proposto em, [3] e [4].

A utilização do conceito de extensão de Galois em teoria da codificação está relacionada diretamente com a construção de códigos cíclicos sobre anéis locais $\mathbb{Z}_q$, onde q é uma potência de um primo, $q = p^k, k \geq 2$. A principal diferença da construção de códigos cíclicos sobre anéis para a construção de códigos cíclicos sobre corpos está no fato de que as raízes do polinômio

gerador dos códigos cíclicos sobre anéis encontram-se na extensão do anel $\mathbb{Z}_q$, ao invés de serem encontradas na extensão do corpo $\mathbb{F}_q \cong GF(p^r)$.

Definição 3.2.18 *Um código cíclico sobre $\mathbb{Z}_q$ com comprimento $n = q^r - 1$, onde $q = p^k$ e r é o grau da extensão de Galois, é denominado código cíclico primitivo.*

Vamos assumir que p e n são relativamente primos, isto é, o máximo divisor comum é um, denotado por $mdc(p, n) = 1$, pois assim garantimos que $(x^n - 1)$ não apresenta fatores quadráticos. Um código cíclico de comprimento n sobre $\mathbb{Z}_q$ é o ideal principal no anel de polinômios sobre $\mathbb{Z}_q$ módulo $(x^n - 1)$ e que este ideal é gerado por qualquer polinômio $g(x)$ que divide $(x^n - 1)$. Seja $\mathbb{Z}_q[x]$ o anel de polinômios na variável x sobre $\mathbb{Z}_q$ onde $p(x)$ é um polinômio primitivo de grau r, irredutível sobre $GF(p)$ e, consequentemente, sobre $\mathbb{Z}_q$. Representamos por $GR(p^k, r)$ o quociente $\mathbb{Z}_q[x]$ pelo ideal gerado por $p(x)$, ou seja,

$$R \simeq GR(p^k, r) \cong \frac{\mathbb{Z}_q[x]}{\langle p(x) \rangle}. \tag{3.23}$$

Assim o anel R é formado por todas as classes laterais de polinômios em x sobre $\mathbb{Z}_q$ mod $p(x)$, isto é, consiste do conjunto dos polinômios de grau menor ou igual a $r-1$ cujas operações binárias de adição e multiplicação são realizadas módulo $p(x)$. Além disso, R é um anel comutativo com identidade denominado extensão de Galois de dimensão r de $\mathbb{Z}_q$. Esta extensão é única a menos de isomorfismo, [62].

O anel $R \cong GR(p^k, r)$ é um anel local, [62], assim seus elementos divisores de zero formam um grupo abeliano aditivo e consistem dos polinômios de grau menor ou igual a $r - 1$ cujos coeficientes são divisores de zero em $\mathbb{Z}_q$. Um polinômio $p(x) \in R$ com pelo menos um coeficiente inversível em $\mathbb{Z}_q$ não é divisor de zero em R e, portanto, pertence a R^* (grupo das unidades de R), ou seja, é sempre possível encontrar um polinômio $q(x) \in R$, tal que $p(x) \cdot q(x) = 1$.

Definição 3.2.19 *[63] Um polinômio não nulo $p(x)$ é um divisor de zero em $\mathbb{Z}_q[x]$ se existe um polinômio $q(x) \in \mathbb{Z}_q[x]$, $q(x) \neq 0$, tal que $p(x) \cdot q(x) = 0$.*

Definição 3.2.20 *[63] Um polinômio $p(x)$ é dito regular se ele não é um divisor de zero no anel $\mathbb{Z}_q[x]$.*

Definição 3.2.21 *[63] Um polinômio regular $p(x)$ é chamado local se $\frac{\mathbb{Z}_q[x]}{\langle p(x) \rangle}$ é uma extensão local de $\mathbb{Z}_q$.*

A irredutibilidade do polinômio $p(x)$ sobre $\mathbb{Z}_q$ é garantida pelo seguinte teorema:

Teorema 3.14 *[63] Seja $p(x)$ um polinômio regular em $\mathbb{Z}_q$. Se existe uma aplicação μ, chamada projeção natural, tal que $\mu(p(x))$ seja diferente de zero e irredutível em $GF(p)$, então $p(x)$ é irredutível em $\mathbb{Z}_q$.*

Como, neste momento, estamos interessados na classe dos códigos cíclicos, nosso objetivo é fornecer um procedimento para a construção de tais códigos. O passo inicial esta relacionado com a fatoração de $(x^n - 1)$. Como o grupo das unidades de R, R^*, é um grupo abeliano

multiplicativo, ele pode ser expresso como um produto de grupos cíclicos. Uma vez encontrado este grupo multiplicativo, o problema da construção de códigos cíclicos se reduz a escolha de determinados elementos deste grupo que sejam raízes do polinômio gerador $g(x)$, que divide $(x^n - 1)$.

Os resultados a seguir fornecem os elementos necessários para a construção do subgrupo cíclico G_n do grupo multiplicativo R^*, que contém todas as raízes de $(x^n - 1)$.

Teorema 3.15 *[62] Existe um único subgrupo cíclico de R^* cuja ordem é relativamente prima a p. Este subgrupo tem ordem $p^r - 1$.*

Teorema 3.16 *[64] Suponha que $f \in R$ gere um subgrupo de ordem n em R^*, onde $mdc(n, p) = 1$. Então o polinômio $(x^n - 1)$ pode ser fatorado como $x^n - 1 = (x - f)(x - f^2) \cdots (x - f^n)$ se, e somente se, $R_p(f)$ tem ordem n em F^* (grupo multiplicativo de $GF(p^r)$), onde $R_p(f)$ é o resto da divisão de f por p (redução de f módulo p).*

Corolário 3.2.2 *[64] Um polinômio $h(x)$, que divide $(x^n - 1)$ e tem coeficientes em $\mathbb{Z}_q$, pode ser fatorado sobre G_n como:*

$$h(x) = (x - \beta^{e1})(x - \beta^{e2}) \cdots (x - \beta^{ej}), \tag{3.24}$$

se, e somente se, $R_p(h(x))$ pode ser fatorado sobre $GF(p^r)$ como:

$$R_p(h(x)) = (x - (R_p(\beta))^{e1})(x - (R_p(\beta))^{e2}) \cdots (x - (R_p(\beta))^{ej}), \tag{3.25}$$

onde β é um elemento primitivo de G_n e $e_j \in \mathbb{Z}$.

Teorema 3.17 *[64] Suponha que $f_1 = R_p(f)$ gere um subgrupo cíclico de ordem n em F^*. Então f gera um subgrupo cíclico de ordem nd em R^*, onde d é um inteiro maior ou igual a um, e f^d gera um subgrupo cíclico g_n de R^*.*

O subgrupo cíclico G_n é obtido do Teorema 3.17, enquanto pelo Corolário 3.2.2, o polinômio minimal $M_i(x)$ associado ao elemento β^i sobre R^* (onde β é um elemento primitivo em G_n) tem como raízes todos os elementos na sequência

$$\beta^i, (\beta^i)^p, (\beta^i)^{p^2}, \cdots (\beta^i)^{p^{r-1}}. \tag{3.26}$$

Portanto, o polinômio minimal $M_i(x)$ pode ser construído de forma similar a construção do polinômio minimal $m_i(x)$ de $R_p(\beta^i)$ sobre $GF(p)$.

Temos ainda a seguinte propriedade:

Teorema 3.18 *[61] Seja β um elemento primitivo em G_n, onde $n = p^r - 1$. Então o elemento $\delta = \beta^{l_1} - \beta^{l_2}$ possui inverso em R se $0 \leq l_1 \neq l_2 \leq n - 1$.*

Definição 3.2.22 *Um código cíclico de comprimento n sobre $GF(p)$ é denominado um código BCH com distância de projeto d se o seu gerador $g(x)$ for o mínimo múltiplo comum dos polinômios minimais de*

$$\beta^m, \beta^{m+1}, \beta^{m+2}, \cdots, \beta^{m+d-2}, \tag{3.27}$$

para algum m inteiro não negativo, onde β é uma raiz primitiva (elemento primitivo) de $(x^n - 1)$, em alguma extensão $GF(p^r)$ de $GF(p)$.

Definição 3.2.23 *Se $n = p^r - 1$, ou seja, se β for um elemento primitivo em F_q, então o código BCH é chamado primitivo.*

Normalmente, consideramos $m = 1$, o que nos fornece o chamado código BCH no sentido estrito. Os códigos BCH no sentido estrito definidos sobre anéis de inteiros, com distância de projeto d e comprimento n, apresentam $\beta, \beta^2, \beta^3, \cdots, \beta^{2t}$ e seus conjugados como raízes de cada um de seus polinômios. Esta propriedade, juntamente com a Definição 3.2.17 de códigos cíclicos sobre anéis $\mathbb{Z}_q$, nos permite especificar a seguinte matriz:

$$H = \begin{bmatrix} 1 & \beta & \beta^2 & \cdots & \beta^{n-1} \\ 1 & \beta^2 & (\beta^2)^2 & \cdots & (\beta^2)^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \beta^{2t} & (\beta^{2t})^2 & \cdots & (\beta^{2t})^{n-1} \end{bmatrix} \tag{3.28}$$

A matriz H acima é a matriz verificação de paridade para um código BCH. Note que os elementos β^i, $1 \leq i \leq 2t$ de H pertecem a G_n, e portanto, os coeficientes de β são tomados módulo n. Substituindo os elementos de β^i pelos vetores linha de comprimento $r(r - uplas)$ correspondentes, temos a matriz H sobre $\mathbb{Z}_q$.

A construção de códigos BCH sobre anéis $\mathbb{Z}_q$, para $q = p^k$ e $k \geq 2$, é análoga à construção de códigos BCH sobre corpos, [64]. A diferença entre essas duas construções reside no fato de que, na primeira, as raízes do polinômio gerador BCH encontram-se na extensão do anel $\mathbb{Z}_q$, ao invés de serem encontradas na extensão do corpo F_q. Vale lembrar também que iremos considerar o caso no qual $mdc(n, p) = 1$.

Podemos especificar um código BCH de comprimento n sobre $\mathbb{Z}_q$, onde $n = p^r - 1$, em termos das raízes de seu polinômio gerador $g(x)$, que pertencem ao subgrupo cíclico G_n. Seja β um elemento primitivo de G_n. Se $\beta^{e_1}, \beta^{e_2}, \cdots, \beta^{e_j}$ são raízes de $g(x)$, então podemos gerar um código BCH com símbolos de $\mathbb{Z}_q$ se escolhermos $g(x)$ como:

$$g(x) = mmc(M_{e_1}(x), M_{e_2}(x), \cdots, M_{e_j}(x)), \tag{3.29}$$

onde $M_{e_i}(x)$ é o polinômio minimal de β^{e_i}. Além disso,

$$\overline{g}(x) = R_p(g(x)) = mmc(m_{e_1(x)}, m_{e_2(x)}, \cdots, m_{e_j(x)}), \tag{3.30}$$

onde $m_{e_i(x)}$ é o polinômio minimal de $R_p(\beta^{e_i})$, gera um código BCH em $GF(p)$.

Portanto, a construção de códigos BCH cíclicos sobre o anel $\mathbb{Z}_q$ reduz-se à escolha de elementos do subgrupo cíclico G_n para serem raízes do polinômio gerador $g(x)$.

Observação 3.2.1 *[3] O método sistemático para o cálculo do mínimo múltiplo comum de um conjunto de polinômios $p_1(x), p_2(x), \cdots, p_n(x)$ é computar o máximo divisor comum, mdc, através do algoritmo de Euclides e então utilizar a seguinte relação:*

$$mmc = (p_1(x), p_2(x), \cdots, p_n(x)) = \frac{\prod_{i=1}^{n} p_1(x)}{mdc(p_1(x), p_2(x), \cdots, p_n(x)}. \tag{3.31}$$

Os dois próximos teoremas estabelecem um limitante inferior para a distância de Hamming do código BCH construído:

Teorema 3.19 *Seja $g(x)$ o polinômio gerador de um código cíclico de comprimento n com símbolos $\mathbb{Z}_q$ e sejam também $\beta^{e_1}, \beta^{e_2}, \cdots, \beta^{e_j}$ as raízes de $g(x)$ em G_n, onde β tem ordem n. Então, a distância mínima do código é maior que o número máximo de inteiros consecutivos módulo n no conjunto $e_1, e_2, \cdots, e_j$.*

Podemos notar que os polinômios geradores dos códigos BCH cíclicos são construídos de forma a respeitar o limitante para a distância mínima indicado no Teorema 3.19.

Apresentaremos o algoritmo de geração de códigos BCH sobre $GR(4, r)$ como estabelecido em [4] e [3]. Neste algoritmo é apresentado a construção de códigos BCH primitivos, sobre anel local $\mathbb{Z}_q$ de ordem $n = (p^r - 1)$, onde $q = p^k$, $p = k = 2$ e r é o grau da extensão de Galois. Se a ordem do corpo base, p, e o comprimento das palavras-código, n, são relativamente primos, isto é, $mdc(p, n) = 1$, então $x^n - 1$ não apresenta multiplicidade de raízes.

A seguir, mostraremos o algoritmo de identificação de sequências de DNA com as correspondentes palavras-código de códigos BCH sobre o anel de Galois $\mathbb{Z}_4$. Apesar do algoritmo ser utilizado na identificação de sequências de DNA sobre anéis e corpos, iremos considerar somente o caso de anéis residuais $\mathbb{Z}_4$. Dados de entrada: a) n=comprimento da sequência de DNA, e b) sequência de DNA.

Algoritmo de Identificação de Sequências de DNA

Passo 1 - Determinar todos os polinômios primitivos $p(x)$, relacionados à extensão de Galois;
Passo 2 - Determinar a extensão de Galois do anel $\mathbb{Z}_4$;
Passo 3 - Determinar o grupo das unidades para o código BCH primitivo, quando o comprimento da sequência de DNA for igual a $n = (2^r - 1)$, ou, determinar o subgrupo das unidades para o código BCH não primitivo, quando o comprimento da sequência de DNA for um submúltiplo de $n = (2^r - 1)$;
Passo 4 - Determinar os polinômios geradores $g(x)$ e $h(x)$;
1^o) Cálculo das raízes dos polinômios minimais;
2^o) Cálculo dos polinômios minimais $M_i(x)$, para todo $i = 1, 2, \cdots, n - 1$;
3^o) Cálculo dos polinômios geradores para todos os valores de t relacionados à distância de Hamming $d_H \leq 2t + 1$;
Passo 5 - Determinar as matrizes G e H e suas transpostas G^T e H^T;
Passo 6 - Rotular a sequência de DNA;
Passo 7 - Verificar se a sequência de DNA é palavra-código;

Passo 8 - Comparar todas as palavras-código armazenadas no Passo 7 com a sequência de DNA do NCBI e mostrar onde os erros ocorreram;
Passo 9 - Voltar para o Passo 4 e determinar outro $g(x)$;
Passo 10 - Repetir os Passos 4 ao Passo 7 para o $g(x)$ obtido no Passo 9, até que se esgotem todas as possibilidades de $g(x)$;
Passo 11 - Voltar para o Passo 1 e escolher outro $p(x)$, e, então, repetir os Passos 2 ao 9 até esgotar todos os $p(x)$ do Passo 1;
Passo 12 - Fim.

No caso de sequências de DNA que possuem comprimentos iguais ou submúltiplos de $n = (2^r + 2)$, a metionina da primeira posição ou *stop* da última posição podem ser desconsiderados, uma vez que a matriz geradora possui uma coluna com os mesmos elementos. O código BCH primitivo sobre a estrutura de anel com parâmetros (n, k, d_H) é capaz de identificar sequências de DNA com comprimento $n = (2^r - 1)$, e com uma única diferença de nucleotídeo da sequência de DNA do NCBI, onde r é o grau da extensão de Galois.

Descrição do algoritmo

Passo 1 - Determinar todos os polinômios primitivos $p(x)$, relacionados à extensão de Galois - Neste passo, os $p(x)$ relacionados ao grau da extensão de Galois, como por exemplo para $r = 6$, (Tabela 3.2), são informados. Em, [38] estes polinômios pode ser encontrados.

Tabela 3.2: Polinômios primitivos da extensão de Galois $r = 6$

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

Passo 2 - Determinar a extensão de Galois do anel $\mathbb{Z}_4$ - Considere o anel $GR(p^k, r) = GR(4, 6)$ como sendo dado pelo quociente do anel $\mathbb{Z}_4[x]$ (conjunto de todos os polinômios com coeficientes em $\mathbb{Z}_4$) pelo ideal gerado pelo mesmo $p(x)$ utilizado para realizar a extensão do corpo no **Passo 4**, isto é,

$$\frac{F_2[x]}{\langle p(x)\rangle} \cong \frac{F_2[x]}{\langle x^6 + x + 1\rangle} = \{b_0 + b_1x + b_2x^2 + \cdots b_5x^5 : b_i's \in \mathbb{Z}_4\}.$$

A seguir, determinaremos a ordem do grupo cíclico pertencente ao grupo das unidades. Sabemos que as operações em $GR^*(4, 6)$ são realizadas módulo $(x^6 + x + 1)$. Como α é uma raiz do polinômio usado tanto na extensão do corpo como na do anel, então $\alpha^6 = -\alpha - 1$. Como os coeficientes dos polinômios em $GR(4, 6)$ estão em $\mathbb{Z}_4$, então $\alpha^6 = 3\alpha + 3$.

Passo 3 - Determinar o grupo das unidades - Do **Passo 5**, resulta que f gera um grupo cíclico de ordem $n.d$ em $GR^*(4, 6)$, onde $d \geq 1 \in \mathbb{Z}$, e f^d gera um subgrupo cíclico cuja

ordem é 63 em $GR^*(4,6)$. Sendo assim, temos que $n.d = 63.d = 126$, implicando que $d = 2$. Consequentemente, $f^2 = (001000) = \alpha^2$ gera um subgrupo cíclico de ordem 63 em $GR^*(4,6)$. Logo $\beta = \alpha^2$ é o elemento primitivo que gera o subgrupo cíclico $G_n = G_{63}$. Esse elemento primitivo será utilizado na construção de um código BCH de comprimento $n = 63$ sobre $\mathbb{Z}_4$. Quando o comprimento n da palavra-código desejada for igual a cardinalidade de G_n, faremos então a construção de um **código BCH primitivo**, onde f gera um grupo cíclico de ordem $n \cdot 2$ em $GR^*(4,r)$.

Passo 4 - Determinar os polinômios geradores $g(x)$ e $h(x)$ - Neste passo, vamos calcular os polinômios geradores $g(x)$ das matrizes geradoras G dos códigos. Os polinômios geradores dos códigos de comprimento n, tem como raízes os elementos na sequência, $\{(\beta^i), (\beta^i)^p, (\beta^i)^{p^2}, (\beta^i)^{p^3}, \cdots, (\beta^i$ Estes polinômios são dados por

$$g(x) = mmc(M_1(x), M_2(x), \cdots, M_{n-1}(x)) \tag{3.32}$$

onde $M_i(x)$ é o polinômio minimal associado ao elemento $\beta_i, \{i = 1, 2, \cdots, n-1\}$ (β é um elemento primitivo em G_n). No caso da palavra-código em questão, cujo comprimento é $n = 63$, os valores de $1 \leq t \leq 31$ serão analisados. Já o polinômio gerador da matriz verificação de paridade H é obtido através da relação:

$$h(x) = \frac{x^n - 1}{g(x)} = \frac{x^{63} - 1}{x^6 + 2x^3 + 3x + 1} \tag{3.33}$$

$h(x) = x^{57} + 2x^{54} + x^{52} + 3x^{51} + x^{47} + 2x^{46} + x^{45} + 2x^{44} + 3x^{42} + x^{41} + 3x^{40} + 3x^{39} + x^{37} + 2x^{35} + 2x^{34} + x^{33} + x^{32} + 3x^{31} + 2x^{29} + x^{28} + 2x^{26} + 3x^{25} + 2x^{24} + 3x^{23} + x^{22} + 2x^{21} + 3x^{20} + x^{19} + x^{18} + 3x^{16} + 3x^{15} + 2x^{14} + 2x^{13} + x^{12} + 2x^{11} + x^9 + 2x^8 + 3x^7 + x^5 + 3x^4 + x^3 + 3x^2 + 3x + 3$ onde os coeficientes do polinômios $h(x)$ pertencem a $\mathbb{Z}_4$. Para cada valor de t, teremos uma distância equivalente e seus respectivos polinômios minimais envolvidos nos cálculos dos $g(x)$, da seguinte maneira:

1^o) **Cálculo das raízes dos polinômios minimais:** Para cada polinômio minimal $M_i(x) = M_i$, com $i = 1, 2, \cdots, 62$, temos:

$$M_1(x) = \{(\beta^1), (\beta^1)^2, \cdots, (\beta^1)^{2^{6-1}(mod 63)}\} \rightarrow M_1 = \{\beta, \beta^2, \beta^4, \beta^8, \beta^{16}, \beta^{32}\},$$
$$M_2(x) = \{(\beta^2), (\beta^2)^2, \cdots, (\beta^2)^{2^{6-1}(mod 63)}\} \rightarrow M_2 = \{\beta^2, \beta^4, \beta^8, \beta^{16}, \beta^{32}, \beta\},$$
$$\vdots = \vdots$$
$$M_{62}(x) = \{(\beta^{62}), (\beta^{62})^2, \cdots, (\beta^{62})^{2^{6-1}(mod 63)}\} \rightarrow M_{62} = \{\beta^{62}, \beta^{61}, \beta^{59}, \beta^{55}, \beta^{47}, \beta^{31}\}.$$

2^o) **Cálculo dos polinômios minimais $M_i(x)$, para todo $i = 1, 2, \cdots, 62$:** Os polinômios minimais são calculados da seguinte maneira:

$$M_1(x) = \{(x - \beta)(x - \beta^2)(x - \beta^4)(x - \beta^8)(x - \beta^{16})(x - \beta^{32})\} \tag{3.34}$$

$$M_1(x) = x^6 + 2x^3 + 3x + 1 \tag{3.35}$$

De maneira análoga, os demais polinômios minimais são determinados. Lembrando que as operações módulo 4 devem ser obedecidas nos cálculos dos polinômios minimais. Alguns polinômios minimais possuem as mesmas raízes. Portanto, estes polinômios minimais são iguais.

3^o) **Cálculo dos polinômios geradores para** $1 \leq t \leq 31 \leq$: O polinômio gerador para cada valor de t é dado por $g(x) = mmc\{M_1(x), M_2(x), \cdots, M_{n-1}(x)\}$, formado pelos polinômios minimais que são diferentes entre si e possuem raízes $\beta, \cdots, \beta^{2t}$. Considerando que a distância mínima do código seja $d_H = 3$, então o polinômio gerador do código é dado por $g_1(x) = x^6 + 2x^3 + 3x + 1$, que gera o código desejado e está relacionado com a matriz geradora G do código BCH sobre $\mathbb{Z}_4$ com parâmetros $(n, k, d_H) = (63, 57, 3)$. De maneira análoga, os demais polinômios geradores para outros valores de t correspondentes a outras distâncias são determinados.

Passo 5 - Determinar as matrizes G **e** H **e suas transpostas** G^T **e** H^T - O polinômio gerador $g_1(x) = x^6 + 2x^3 + 3x + 1$ está relacionado à matriz geradora G. Realizando os deslocamentos dos coeficientes do polinômios $g(x)$ da esquerda para a direita, obtendo uma matriz G com dimensão 57×63. A matriz G^T com dimensão 63×57 é determinada como sendo a troca da linha pela coluna. Determinado o polinômio $h(x)$ neste passo, realizamos os deslocamentos dos coeficientes do polinômio gerador $h(x)$ da direita para a esquerda e obtemos a matriz H com dimensão 6×63. A matriz H^T com dimensão 63×6 é determinada pela troca da linha pela coluna.

Passo 6 - Rotular a sequência de DNA - Este passo determina as 24 permutações entre o alfabeto do código genético $N = \{A, C, G, T/U\}$ e o alfabeto do código BCH $\mathbb{Z}_4 = \{0, 1, 2, 3\}$ da sequência de DNA a ser analisada. Uma vez que o mapeamento entre $N \to \mathbb{Z}_4$ não é conhecido, consideremos todas as permutações entre esses dois conjuntos. Cada uma das 24 permutações foi definida como um caso, mostrado na Tabela 3.3.

Tabela 3.3: Rotulamentos determinados pelas 24 permutações, encontrada em [3].

Caso	$N \longrightarrow Z_4$	Caso	$N \longrightarrow Z_4$	Caso	$N \longrightarrow Z_4$
caso 1	{A,C,G,T}={0,1,2,3}	caso 9	{A,C,G,T}={1,2,0,3}	caso 17	{A,C,G,T}={2,3,0,1}
caso 2	{A,C,G,T}={0,1,3,2}	caso 10	{A,C,G,T}={1,2,3,0}	caso 18	{A,C,G,T}={2,3,1,0}
caso 3	{A,C,G,T}={0,2,1,3}	caso 11	{A,C,G,T}={1,3,0,2}	caso 19	{A,C,G,T}={3,0,1,2}
caso 4	{A,C,G,T}={0,2,3,1}	caso 12	{A,C,G,T}={1,3,2,0}	caso 20	{A,C,G,T}={3,0,2,1}
caso 5	{A,C,G,T}={0,3,2,1}	caso 13	{A,C,G,T}={2,0,1,3}	caso 21	{A,C,G,T}={3,1,0,2}
caso 6	{A,C,G,T}={0,3,1,2}	caso 14	{A,C,G,T}={2,0,3,1}	caso 22	{A,C,G,T}={3,1,2,0}
caso 7	{A,C,G,T}={1,0,2,3}	caso 15	{A,C,G,T}={2,1,0,3}	caso 23	{A,C,G,T}={3,2,0,1}
caso 8	{A,C,G,T}={1,0,3,2}	caso 16	{A,C,G,T}={2,1,3,0}	caso 24	{A,C,G,T}={3,2,1,0}

Passo 7 - Verificar se a sequência de DNA é palavra-código - O procedimento usado para terminar quais das sequências são palavras-códigos do código $(63, k, d_H)$, é o seguinte: verifique se $v.H^T = 0$. Caso seja verdade, então declare v como palavra-código. Caso contrário, $v.H^T \neq 0$, considere todas as possibilidades de troca de nucleotídeos em cada posição. Aquelas em que $v.H^T = 0$ são armazendas.

Passo 8 - Comparar todas as palavras-código armazenadas no Passo 7 com a sequência de DNA do NCBI e mostrar onde os erros ocorreram - Neste passo, todas as palavras-código armazenadas no passo anterior estão royuladas na forma do alfabeto do código, $\mathbb{Z}_4 = \{0, 1, 2, 3\}$, e serão convertidas em nucleotídeos usando o rotulamento do alfabeto do código genético $N = \{A, C, G, T\}$. Após o rotulamento, as palavras-código são comparadas, uma-a-uma, com a sequência de DNA original mostrando onde os nucleotídeos diferem, e armazena os resultados.

Passo 9 - Voltar para o Passo 4 e determinar outro $g(x)$ - Neste passo, determinamos outro valor da distância mínima d_H, por exemplo $d_H = 5$, e utilizamos o mesmo procedimento, apresentado no **Passo 4**, para calcular o polinômio gerador relativo a esta distância.

Passo 10 - Repetir os Passos 4 ao Passo 7 para o $g(x)$ **obtido no Passo 9, até que se esgotem todas as possibilidades de** $g(x)$ - Neste passo, o algoritmo determina todas as palavras-código encontradas com nenhum, 1 e 2 nucleotídeos de diferença através de todos os polinômios geradores relativos à distância mínima $1 \leq d_H \leq n$, neste exemplo $1 \leq d_H \leq 63$, e armazena os resultados.

Passo 11 - Voltar para o Passo 1 e escolher outro $P(x)$**, e, então, repetir os Passos 2 ao 9 até esgotar todos os** $p(x)$ **do Passo 1**

Passo 12 - Fim.

Capítulo 4

Análise do Splicing Alternativo via CCE

Neste capítulo abordamos três temas que foram desenvolvidos ao longo da pesquisa. No primeiro, mostraremos que uma sequência de DNA (gene) é identificada como palavra-código de um código corretor de erros (BCH) sobre anel. Através dessa identificação é possível estabelecer (conjecturar) uma estrutura matemática associada aos éxons e íntrons, uma vez que os éxons são separados dos íntrons e justapostos de diferentes formas para a geração de proteínas. Por outro lado, o splicing alternativo vem sendo tema de várias pesquisas devido ao pouco conhecimento de um mecanismo de grande importância para a diversidade proteômica e responsável pelo aumento na capacidade de codificação de genes, sendo identificado em plantas, animais, alguns fungos e em quase todos os organismos eucarióticos.

O segundo tema trata de um modelo para gerar e reproduzir partes de um genoma, usando a matriz geradora de um código corretor de erros (BCH) sobre anel, visto que na biologia os genes são separados do restante do genoma para realizar funções biológicas. No terceiro tema serão usados os códigos de Varshamov-Tenengolts para reconstruir uma sequência em que houve uma única deleção de nucleotídeo, ou uma única inserção de nucleotídeo. Vamos mostrar que um RNA maduro pode ser identificado como uma palavra-código de um código (BCH), e do mesmo modo é identificado como palavra-código de um código de Varshamov-Tenengolts.

Consideramos nesta pesquisa o gene Trav7 localizado no cromossomo 14 do genoma humano, com 511 nucleotídeos, com dois éxons e um íntron. O cromossomo 14 está envolvido no processo biológico muito importante conhecido como telomerase. Em relação as doenças genética ele esta relacionado diretamente com a seguintes doenças: paraplegia espática, uma forma grave da síndrome de Usher e a doença de Niemann-Pick.

Também consideramos o gene Hint-1 do nematoide Caenorhabditis Elegans com 511 nucleotídeos e 3 éxons e 2 íntrons. Brenner em meados da década de 60 do século passado prôpos que o Caenorhabditis Elegans fossem um organismo modelo para pesquisa devido a alguns elementos presentes neste nematoide: primeiro por causa do seu ciclo de vida curto, em segundo pelo tamanho pequeno e pela facilidade de manutenção de grandes populações. Em terceiro, pela facilidade de cultivo em laboratório. Em quarto, pela existência de indivíduos hermafroditas protândricos, sendo a população constituída por $99,9\%$ de hermafroditas e $0,1\%$ de machos. Em

quinto, o Caenorhabditis Elegans selvagem pode ser congelados indefinidamente em nitrogênio líquido e recuperado posteriormente. Os genes são partes funcionais do DNA, oferecendo as informações básicas para a produção de todas as proteínas que o organismo necessita.

Para o modelo de geração de partes de um genoma usamos o genoma do plasmídeo Loclococcus Latis pcl 21 com comprimento 2047 nucleotídeos e separado em 9 regiões. Sendo os plasmídeos pequenos fragmentos de DNA bacteriano de forma circular. Eles podem se modificar com a adição de novos fragmentos de DNA e são facilmente inseridos em bactérias, sendo utilizados para o transporte de DNA para o interior de células alvo. O genoma contém toda a informação hereditária de um organismo que está codificada em seu DNA, incluindo genes e sequências não-codificadoras que são importantes para a regulação gênica.

4.1 Modelo para a Geração de Partes de uma Sequência

Para um melhor entendimento do procedimento utilizado na análise utilizada nesta pesquisa, vamos considerar um exemplo de uma sequência fictícia de DNA que foi identificada como uma palavra-código cuja matriz geradora tem dimensões menores do que as matrizes geradoras dos genes Trav7 e Hint-1. Na Subseção 4.1.1 vamos mostrar como localizar na matriz geradora as partes em que temos interesse, como gerar partes de uma sequência de informação e na Subseção 4.1.2 vamos usar o código de Varshamov-Tenengolts para corrigir uma inserção ou uma deleção.

4.1.1 Geração de partes de uma sequência de informação

Dado $g(x) = 1x^4 + 3x^3 + 2x^2 + 1$ considere a matriz geradora G de tamanho 12×15 mostrada abaixo, considere também a palavra-código v mostrada na Tabela 4.1.

$$G = \begin{bmatrix}
1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1
\end{bmatrix}$$

Tabela 4.1: Palavra-código v

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

Dada a matriz geradora G e a palavra-código v , é necessário determinar o vetor de informação u. Uma maneira de determinar o vetor u é como se segue: note que v=u.G, ou

equivalentemente,

$$(v_0 \quad v_1 \cdots v_{14}) = (u_0 \quad u_1 \cdots u_{11}). \begin{bmatrix} 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 \end{bmatrix}$$

Assim,

$$v_0 = u_0 \cdot 1 + u_1 \cdot 0 + u_2 \cdot 0 + u_3 \cdot 0 + u_4 \cdot 0 + u_5 \cdot 0 + u_6 \cdot 0 + u_7 \cdot 0 + u_8 \cdot 0 + u_9 \cdot 0 + u_{10} \cdot 0 + u_{11} \cdot 0$$

$$v_1 = 0 \cdot 2 + u_1 \cdot 1 + u_2 \cdot 0 + u_3 \cdot 0 + u_4 \cdot 0 + u_5 \cdot 0 + u_6 \cdot 0 + u_7 \cdot 0 + u_8 \cdot 0 + u_9 \cdot 0 + u_{10} \cdot 0 + u_{11} \cdot 0$$

$$v_2 = 0 \cdot 3 + 2 \cdot 2 + u_2 \cdot 1 + u_3 \cdot 0 + u_4 \cdot 0 + u_5 \cdot 0 + u_6 \cdot 0 + u_7 \cdot 0 + u_8 \cdot 0 + u_9 \cdot 0 + u_{10} \cdot 0 + u_{11} \cdot 0$$

$$v_3 = 0 \cdot 1 + 2 \cdot 3 + 1 \cdot 2 + u_3 \cdot 1 + u_4 \cdot 0 + u_5 \cdot 0 + u_6 \cdot 0 + u_7 \cdot 0 + u_8 \cdot 0 + u_9 \cdot 0 + u_{10} \cdot 0 + u_{11} \cdot 0$$

$$v_4 = 0 \cdot 0 + 2 \cdot 1 + 1 \cdot 3 + 3 \cdot 2 + u_4 \cdot 1 + u_5 \cdot 0 + u_6 \cdot 0 + u_7 \cdot 0 + u_8 \cdot 0 + u_9 \cdot 0 + u_{10} \cdot 0 + u_{11} \cdot 0$$

$$v_5 = 0 \cdot 0 + 2 \cdot 0 + 1 \cdot 1 + 3 \cdot 3 + 1 \cdot 2 + u_5 \cdot 1 + u_6 \cdot 0 + u_7 \cdot 0 + u_8 \cdot 0 + u_9 \cdot 0 + u_{10} \cdot 0 + u_{11} \cdot 0$$

$$v_6 = 0 \cdot 0 + 2 \cdot 0 + 1 \cdot 0 + 3 \cdot 1 + 1 \cdot 3 + 0 \cdot 2 + u_6 \cdot 1 + u_7 \cdot 0 + u_8 \cdot 0 + u_9 \cdot 0 + u_{10} \cdot 0 + u_{11} \cdot 0$$

$$v_7 = 0 \cdot 0 + 2 \cdot 0 + 1 \cdot 0 + 3 \cdot 0 + 1 \cdot 1 + 0 \cdot 3 + 3 \cdot 2 + u_7 \cdot 1 + u_8 \cdot 0 + u_9 \cdot 0 + u_{10} \cdot 0 + u_{11} \cdot 0$$

$$v_8 = 0 \cdot 0 + 2 \cdot 0 + 1 \cdot 0 + 3 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 3 \cdot 3 + 2 \cdot 2 + u_8 \cdot 1 + u_9 \cdot 0 + u_{10} \cdot 0 + u_{11} \cdot 0$$

$$v_9 = 0\cdot 0 + 2\cdot 0 + 1\cdot 0 + 3\cdot 0 + 1\cdot 0 + 0\cdot 0 + 3\cdot 1 + 2\cdot 3 + 1\cdot 2 + u_9\cdot 1 + u_{10}\cdot 0 + u_{11}\cdot 0$$

$$v_{10} = 0\cdot 0 + 2\cdot 0 + 1\cdot 0 + 3\cdot 0 + 1\cdot 0 + 0\cdot 0 + 3\cdot 0 + 1\cdot 2 + 1\cdot 3 + 0\cdot 2 + u_{10}\cdot 1 + u_{11}\cdot 0$$

$$v_{11} = 0\cdot 0 + 2\cdot 0 + 1\cdot 0 + 3\cdot 0 + 1\cdot 0 + 0\cdot 0 + 3\cdot 0 + 2\cdot 0 + 1\cdot 1 + 0\cdot 3 + 1\cdot 2 + u_{11}\cdot 1$$

Como $v = (0 \quad 2 \quad 1 \quad 3 \quad 0 \quad 0 \quad 1 \quad 1 \quad 2 \quad 3 \quad 2 \quad 2 \quad 1 \quad 2 \quad 3)$, então: $0 = u_0 \cdot 1$, $2 = u_1 \cdot 1$; $1 = u_2 \cdot 1$; $3 = 2 + 2 + u_3 \cdot 1$; $0 = 0 + 2 + 3 + 2 + u_4 \cdot 1$; $0 = 0 + 0 + 1 + 1 + 2 + u_5 \cdot 1$; $1 = 0+0+0+3+3+0+u_6\cdot 1$; $1 = 1+2+u_7\cdot 1$; $2 = 1+0+u_8\cdot 1$; $3 = 3+2+2+u_9\cdot 1$; $2 = 1+u_{10}\cdot 1$; $2 = 3 + u_{11}\cdot 1$. Assim, o vetor u é dado por $u = (0 \quad 2 \quad 1 \quad 3 \quad 1 \quad 0 \quad 3 \quad 2 \quad 1 \quad 0 \quad 1 \quad 3)$.

Sem perda de generalidade considere que a matriz G será subdividida em três partes, sendo parte 1 será da coluna 1 ao coluna 5, a parte 2 será do coluna 6 à coluna 10 e a parte 3 será da coluna 11 à coluna 15 da palavra-código v.

Para localizar estas partes na matriz geradora G podemos relacionar cada elemento da palavra-código com uma coluna da matriz, pois a matriz geradora G possui 15 colunas e a palavra-código possui 15 elementos. A Tabela 4.2 ilustra onde cada parte da informação está localizada na matriz geradora G. As correspondentes partes na palavra-código v é mostrada na Tabela 4.3, sendo os elemento em vermelho correspondente a parte 1, os elementos em azul correspondem a parte 2 e os elementos em verde correspondem a parte 3.

Tabela 4.2: Matriz geradora G separada em partes

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

Após identificar onde se localiza a informação de cada parte na matriz geradora, podemos notar que estas informações são submatrizes, neste caso as submatrizes da parte 1 e parte 3 tem tamanhos iguais e parte 2 tem tamanho diferente. Além disso notamos que os subespaços não são independentes portanto, não sendo uma soma direta. Quando encontramos as submatrizes

Tabela 4.3: Palavra-código v separada em partes

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

correspondentes a parte 1, parte 2 e parte 3 notamos que os deslocamentos ciclícos ficam divididos entre as partes como podemos visualizar na Tabela 4.2.

Na matriz geradora G e na palavra-código v é fácil identificar a localização dos éxons e íntrons, mas já no caso do vetor u é mais difícil identificar onde está cada uma destas três partes. Com isso vem a seguinte pergunta: como relacionar os 15 elementos da palavra-código v com os 12 elementos do vetor u? A relação da palavra-código v com o vetor u pode ser associada de duas formas, primeiro verificando o grau do polinômio gerador ou olhando diretamente na matriz e observando quais as linhas e colunas que contém a parte da informação que vamos gerar.

Observe que a informação da parte 1 está localizada na matriz geradora G da linha 1 até a linha 5 e da coluna 1 até a coluna 5. Na parte 1 a informação no vetor u esta localizada do elemento 1 ao elemento 5, efetuando a multiplicação da parte 1 do vetor u pela parte 1 da matriz geradora G obtemos a parte 1 na palavra-código v. Na parte 2 visualizamos que a informação na matriz geradora G esta localizada da linha 3 até a linha 10 e da coluna 6 até a coluna 10. Assim podemos notar que a informação no vetor u pode ser relacionada com as linhas da matriz geradora G que contém a informação, então a informação da parte 2 no vetor u esta localizada do elemento 3 até o elemento 10. Efetuando a multiplicação da parte 2 do vetor u pela parte 2 da matriz geradora G verificamos que esta multiplicação gera os elementos de 6 a 10 da palavra-código v.

Considerando o caso do polinômio gerador, note que este polinômio tem grau 3, como na parte 1 usamos os elementos de 1 a 5 do vetor u, pela lógica os elementos da parte 2 são os elemento de 6 a 10, mas fazendo os cálculos verificamos que não gera a parte 2 da palavra-código v. Assim percebemos que existe a necessidade de considerar o grau do polinômio que no caso é 3, então em vez de começarmos no elemento 6 começamos no elemento 3 do vetor u, assim consideramos os elementos de 3 a 10 e multiplicamos pela informação da parte 2 da matriz geradora G resultando na parte 2 da palavra-código v.

A informação da parte 3 na matriz geradora G está localizada da linha 8 até a linha 12 e da coluna 11 até a coluna 15, relacionando a matriz geradora G com o vetor u podemos visualizar que teremos que usar os elementos de 8 até 12 do vetor u. Usando os elementos de 8 até 12 do vetor u e efetuando a multiplicação pela informação da parte 3 da matriz geradora G, temos como resultado a parte 3 da palavra-código v.

Pela ótica do polinômio gerador podemos perceber que o polinômio que gera esta matriz é de grau 3. Como na parte 2 usamos até os elementos de 3 a 10 do vetor u, pela lógica os elementos da parte 3 são os elemento de 11 a 12, mas fazendo os cálculos verificamos que não gera a parte 3 da palavra-código v. Assim percebemos que existe a necessidade de considerar o grau do polinômio que no caso é 3, então em vez de começarmos no elemento 11 começamos no elemento 8 do vetor u, assim consideramos os elementos de 8 a 12 e multiplicamos pela

informação da parte 3 da matriz geradora G resulta na parte 3 da palavra-código v.

Tabela 4.4: Vetor u separado em partes

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

Podemos visualizar na Tabela 4.4 que no vetor u a primeira parte está localizada do elemento 1 até o elemento 5 em vermelho e azul, sendo que a primeira parte em azul é comum para a primeira parte e segunda parte. A segunda parte é localizada do elemento 3 ao elemento 10 em azul e negrito, sendo a segunda parte em azul comum a segunda parte e terceira parte. A parte 3 está localizada nos elementos de 8 a 12 nas cores azul e verde.

4.1.2 Utilização do código de Varshamov-Tenengolts

Vamos usar o código de Varshamov-Tenengolts para mostrar como é corrigida uma inserção ou uma deleção de informação. Sabemos que o código de Varshamov-Tenengolts não corrige uma combinação de inserção e deleção na mesma sequência. Dada a palavra-código v mostrada na Tabela 4.5 e seja $q = 4$ e $n = 15$.

Tabela 4.5: Palavra-código v

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

Usando o código de Varshamov-Tenengolts vamos determinar o vetor α, e seu primeiro elemento α_1 pode ser qualquer símbolo binário. Considere α_1=1, como $q = 4$ e $n = 15$ então α_i é dado pela Relação 4.1 mostrada abaixo, e o vetor binário resultante da Relação 4.1 é mostrado na Tabela 4.6.

$$\alpha_i = \begin{cases} 1 & \text{se } a_i \geq a_{i-1}; \\ 0 & \text{se } a_i < a_{i-1}. \end{cases} \tag{4.1}$$

Tabela 4.6: Vetor α resultante da Palavra-código v

1	1	0	1	0	1	1	1	1	1	0	1	0	1	1

Uma vez que conhecemos a palavra-código v, Tabela 4.5, e o correspondente vetor α mostrado na Tabela 4.6, podemos calcular os parâmetros β e γ, dados por:

$$\sum_{i=1}^{n} a_i \equiv \beta (mod\, q) \tag{4.2}$$

$$\sum_{i=1}^{n}(i-1)\alpha_i \equiv \gamma(mod\, n) \tag{4.3}$$

Fazendo os cálculos encontramos:

$$\beta \equiv (0+2+1+3+0+0+1+1+2+3+2+2+1+2+3)\, mod\, 4 \equiv 3\, mod\, 4 \tag{4.4}$$

$$\gamma \equiv (0+1+0+3+0+5+6+7+8+9+0+11+0+13+14)\, mod\, 15 \equiv 2\, mod\, 15 \tag{4.5}$$

Após encontrar estes parâmetros, simulamos uma deleção na palavra-código, ou seja, deletamos o elemento 3 na posição 4, mostrado em vermelho na Tabela 4.5. Com isso, criamos um vetor v', mostrado na Tabela 4.7. Através do vetor v' encontramos o vetor α' mostrado na Tabela 4.8, de posse destes valores encontramos os parâmetros S_1, S_2 e W, necessários para a reconstrução da sequência original, permitindo uma única decodificação. S_1 é igual ao valor do símbolo perdido, W é o peso (número de símbolos diferentes de zero) da sequência α' e S_1, S_2 são os menores resíduos não negativos das congruências. Os cálculos e os vetores v' e α' são mostrados a seguir.

Tabela 4.7: Vetor v' resultante da deleção de um elemento da palavra-código v

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

Tabela 4.8: Vetor α' resultante do vetor v'

1	1	0	0	1	1	1	1	1	0	1	0	1	1

$$S_1 \equiv \beta - \sum_{i=1}^{n} a'_i (mod\, q) \tag{4.6}$$

$$S_2 \equiv \gamma - \sum_{i=1}^{n}(i-1)\alpha'_i (mod\, n) \tag{4.7}$$

$$S_1 \equiv 3 - (0+2+1+0+0+1+1+2+3+2+2+1+2+3)\, mod\, 4 \equiv 3\, mod\, 4 \tag{4.8}$$

$$S_2 \equiv 2 - (0+1+0+0+4+5+6+7+8+0+10+0+12+13)\, mod\, 15 \equiv 11\, mod\, 15 \tag{4.9}$$

é o peso (número de símbolos diferentes de zero) da sequência α'

Para encontrar o valor do peso W verificamos na sequência α' a quantidade de símbolos diferentes de zero, assim encontramos W = 10, como $S_2 \geq$ W , portanto inserimos o símbolo 1 na sequência α' de modo que o número de zeros do lado esquerdo de onde o símbolo será inserido seja igual a S_2 - W, neste caso como S_2=11 e W = 10, assim S_2 - W=11-10=1, então temos que colocar o símbolo 1 a direita do primeiro símbolo 0. Como o primeiro símbolo 0 esta na posição 3, então inserimos o símbolo 1 na posição 4, este novo vetor aqui chamado de α'_1 é mostrado na Tabela 4.9.

Tabela 4.9: Vetor α'_1

1	1	0	1	0	1	1	1	1	1	0	1	0	1	1

Como S_1=3, então concluímos que o símbolo que foi excluído é o símbolo 3, assim a única possibilidade é colocar o símbolo 3 na posição 4 da sequência, na Tabela 4.10 é mostrada a sequência corrigida.

Tabela 4.10: Sequência reconstruida

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

Concluímos que a sequência corrigida é igual a sequência enviada.

Uma outra abordagem é que o código de Varshamov-Tenengolts permite corrigir uma única inserção de informação, neste exemplo vamos fazer uma inserção de informação na palavra-código v mostrada na Tabela 4.5. Usando o código de Varshamov-Tenengolts vamos determinar o vetor α, e seu primeiro elemento α_1 pode ser qualquer símbolo binário. Considere α_1=1,seja $q = 4$ e $n = 15$, então α_i é dado pela Relação 4.1 mostrada anteriormente, e o vetor binário resultante da relação 4.1 é mostrado na Tabela 4.6.

Uma vez que conhecemos a palavra-código v, Tabela 4.5, e o correspondente vetor α mostrado na Tabela 4.6, logo podemos calcular os parâmetros β e γ, dados pela relação 4.3, mostrada anteriormente.

Fazendo os cálculos encontramos:

$$\beta \equiv (0+2+1+3+0+0+1+1+2+3+2+2+1+2+3)\, mod\, 4 \equiv 3\, mod\, 4 \tag{4.10}$$

e

$$\gamma \equiv (0+1+0+3+0+5+6+7+8+9+0+11+0+13+14)\, mod\, 15 \equiv 2\, mod\, 15 \tag{4.11}$$

Agora vamos fazer uma inserção de informação na palavra-código v. Iremos inserir o número 2 na posição 11 sendo mostrado na Tabela 4.11, e depois vamos reconstruir esta sequência usando o código de Varshamov-Tenengolts.

Tabela 4.11: Vetor v'

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

Tabela 4.12: Vetor α'

1	1	0	1	0	1	1	1	1	1	0	1	1	0	1	1

Após encontrar o vetor v' vamos encontrar o seu vetor correspondente α', mostrado na Tabela 4.12 obtido através da fórmula mostrada na relação 4.1:

Para reconstruir a sequência enviada é necessário encontrar os parametros S_1, S_2 e W, que são mostrados a seguir com suas respectivas fórmulas:

$$S_1 \equiv \sum_{i=1}^{n} a'_i - \beta(mod\, q) \equiv 2\,(mod\, 4) \tag{4.12}$$

e

$$S_2 \equiv \sum_{i=1}^{n} (i-1)\alpha'_i - \gamma(mod\, n) \equiv 14\,(mod\, 15) \tag{4.13}$$

W = 12

Para reconstruir a sequência olhamos o S_2 e o W. Neste caso como $S_2 >$ W-1 então descartamos qualquer símbolo 1 de modo que o número de zeros no lado direito deste símbolo seja igual a $n - S_2$. Como $n = 15$ e $S_2 = 14$, então $n - S_2 = 15 - 14 = 1$, assim temos que localizar o último zero e excluir o 1 anterior a ele. O último zero se encontra na posição 14, e o 1 anterior a ele esta na posição 13, então excluimos o elemento 13 da vetor α' mostrado na Tabela 4.12, a nova sequência α'_1 é mostrada abaixo na Tabela 4.13.

Tabela 4.13: Vetor α'_1

1	1	0	1	0	1	1	1	1	1	0	1	0	1	1	

Como S_1=2, então concluímos que o símbolo que foi incluindo é o 2, assim a única possibilidade de decodificação é excluir o símbolo 2 na posição 11 da sequência, assim concluímos que a sequência corrigida é igual a sequência enviada.

4.2 Modelo para a geração de éxons e íntrons

4.2.1 Gene Trav7

O gene Trav7 postado no banco de dados biológicos NCBI é identificado pelo "geneID"de número **28686** mostrado na Figura 4.1. Os nucleotídeos em verde mostram o éxon 1 com

atggagaagatgcggagacctgtcctaattatattttgtctatgtcttggctgtaagttgagggttctaagaactgggga
ccccaggagacatttattcaagtccttttggggagatggggatgtagtctggacttacttgtcattgcttgtttgagattaa
gaaataaaattatgaaaggtctaaattaaaatgtacatattgtacctgatgtctttctgaatagggcaaatggagaaa
accaggtggagcacagccctcattttctgggaccccagcagggagacgttgcctccatgagctgcacgtactctgtcag
tcgttttaacaatttgcagtggtacaggcaaaatacagggatgggtcccaaacacctattatccatgtattcagctggat
atgagaagcagaaaggaagactaaatgctacattactgaagaatggaagcagcttgtacattacagccgtgcagcct
gaagattcagccacctatttctgtgctgtagatg

Figura 4.1: Sequência em nucleotídeos do gene Trav7.

comprimento 52, os nucleotídeos em azul mostram o íntron 1 com comprimento 174 e os nucleotídeos em vermelho mostram o éxon 2 com comprimento 285, totalizando o comprimento de 511 nucleotídeos.

Dada esta sequência, o objetivo é identificá-la com uma palavra-código, via o algoritmo de geração de sequências de DNA usando códigos BCH sobre anéis proposto em [4]- [3]- [65] e [66]. Identificada a palavra-código obtemos através dos procedimentos delineados no Capítulo 3 o correspondente polinômio gerador dado por $g(x) = 1x^9 + 3x^8 + 2x^7 + 2x^6 + 1x^5 + 1x^4 + 2x^2 + 3$ e rótulo caso 3, Tabela 3.3 bem como a matriz geradora com 502 linhas e 511 colunas mostrada abaixo:

$$G = \begin{bmatrix}
3 & 0 & 2 & 0 & 1 & 1 & 2 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3 & 0 & 2 & 0 & 1 & 1 & 2 & 2 & 3 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 3 & 0 & 2 & 0 & 1 & 1 & 2 & 2 & 3 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 3 & 0 & 2 & 0 & 1 & 1 & 2 & 2 & 3 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 3 & 0 & 2 & 0 & 1 & 1 & 2 & 2 & 3 & 1 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 3 & 0 & 2 & 0 & 1 & 1 & 2 & 2 & 3 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 3 & 0 & 2 & 0 & 1 & 1 & 2 & 2 & 3 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 3 & 0 & 2 & 0 & 1 & 1 & 2 & 2 & 3 & 1
\end{bmatrix}$$

A Tabela 4.14 ilustra a correspondente palavra-código v_1.

O próximo passo é determinar o vetor informação u_1 que multiplicado pela matriz geradora G resulta na palavra-código v_1 ou equivaletemente $(u_0 \quad u_1 \quad u_2 \cdots u_{501}).G = (v_0 \quad v_1 \quad v_2 \cdots v_{510})$. Considere o processo pelo qual foi obtido o vetor u mostrado no início deste capítulo, assim temos que os elementos do vetor u_1 são determinados da seguinte forma:

Tabela 4.14: Palavra-código v_1 gene Trav7

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

$$v_0 = u_0 \cdot 3 + u_1 \cdot 0 + u_2 \cdot 0 + \cdots + u_{501} \cdot 0 + u_{501} \cdot 0$$

$$v_1 = 0 \cdot 0 + u_1 \cdot 3 + u_2 \cdot 0 + \cdots + u_{501} \cdot 0 + u_{501} \cdot 0$$

$$v_2 = 0 \cdot 0 + 1 \cdot 0 + u_2 \cdot 3 + \cdots + u_{501} \cdot 0 + u_{501} \cdot 0$$

$$\vdots$$

$$v_{510} = 0 \cdot 0 + 1 \cdot 0 + 3 \cdot 0 + \cdots + 1 \cdot u_{501}$$

Fazendo todos os cálculos para os 502 elementos do vetor u_1, encontramos todos os componentes do vetor que são mostrados na Tabela 4.15. Depois de encontrado o vetor u_1 fizemos u_1.G para verificar a igualdade com a palavra-código v_1, em seguida fizemos mais alguns testes para verificação dos dados como, a multiplicação módulo 4 da matriz geradora pela sua correspondente H transposta, multiplicamos módulo 4 a palavra-código pela H transposta para verificar se a síndrome era 0.

Tabela 4.15: Vetor u_1 referente ao gene Trav7

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

Neste trabalho nossa motivação era gerar éxons e íntrons separadamente usando a estrutura dos códigos corretores de erros visto que, no splicing alternativo os éxons e íntrons são separados por macromoléculas chamadas spliceossomos e depois justapostos de diferentes formas. Após encontrarmos o vetor u_1, tínhamos os dados para começar a tentar fazer um modelo que fizesse a geração de cada parte do gene. No caso da palavra-código é trivial encontrar em qual lugar se encontra a informação do éxon 1, íntron 1 e éxon 2, pois temos 511 nucleotídeos e a palavra-código tem comprimento 511, cada éxon e íntron é facilmente identificado, sendo mostrado na Tabela 4.16, com o éxon 1 em roxo o íntron 1 em azul e o éxon 2 em vermelho.

Na matriz geradora podemos encontrar onde estão cada éxon e íntron olhando as colunas e relacionado com os nucleotídeos do gene, já que a matriz possui 511 colunas encontramos facilmente onde estão cada éxon e íntron. Após identificar onde se localiza a informação de cada éxon e de cada íntron na matriz geradora, podemos notar que estas informações são submatrizes, no caso do trav7 com tamanho diferentes. Além disso, notamos que os subespaços não são independentes não sendo soma direta. Quando encontramos as submatrizes correspondentes a éxons e íntrons percebemos que os deslocamentos ciclícos ficam divididos entre éxons e íntrons como podemos visualizar na Tabela 4.2 no exemplo mostrado no começo do capítulo.

Após identificarmos onde estão cada éxon e íntron na palavra-código e na matriz geradora faltava fazer a mesma identificação no vetor u_1, como este vetor tem 502 elementos ficou um pouco mais complicado para sabermos onde estavam localizados éxons e íntrons, pois, temos

Tabela 4.16: Palavra-código separada em éxons e íntrons do gene Trav7

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

um gene de comprimento 511, como iremos relacionar estes números diferentes, a partir disso pensamos em como gerar o primeiro éxon, neste caso ele tem comprimento 52, então pegamos os 52 primeiros elementos do vetor u_1 e multiplicamos módulo 4 pela parte da matriz geradora onde está localizado o éxon 1, sendo uma submatriz de tamanho 52×52, localizada da linha 1 até a linha 52 e da coluna 1 até a coluna 52 da matriz geradora, na qual obtemos como resultado os 52 primeiros elementos da palavra-código v_1 que resultou no éxon 1.

A partir da reprodução do éxon 1 era necessário agora reproduzir o íntron 1, olhando a informação referente ao íntron 1 na matriz geradora, podemos visualizar uma submatriz 183×174 que esta localizada da linha 44 até a linha 226 e da coluna 53 até a coluna 226. Vislumbrando a submatriz referente a informação do íntron 1, podemos perceber que o número de linhas desta submatriz será o mesmo número de elementos vetor u_1 usado para gerar o íntron 1, como na matriz geradora estas informações estão compreendidas da linha 44 até a linha 226, então os elementos do vetor u_1 que seram multiplicados módulo 4 pela submatriz seram os elementos do vetor u_1 de 44 até 226. Efetuando a multiplicação dos elementos do vetor u_1 referentes ao íntron 1 (elemento 44 ao elemento 226) pela submatriz referente ao íntron 1 obtemos como resposta os 174 elementos da palavra-código v_1 referente ao íntron 1.

Uma outra abordagem é pela ótica do polinômio gerador, assim verificamos que a parte do vetor u_1 que contém a informação do íntron 1 está relacionada com o grau do polinômio gerador, sabíamos que uma parte do vetor u_1 que contém o íntron 1 era do elemento 53 até o elemento

226, notamos que para reproduzir o íntron 1 tínhamos que acrescentar mais nove elementos que é exatamente o grau do polinômio gerador. A partir destes fatos em vez de usar os elementos de 53 até 226 do vetor u_1 usamos estes elementos combinados com os nove elementos anteriores a estes, assim usamos os elementos de 44 até 226 do vetor u_1. Efetuamos a multiplicação módulo 4 da parte do vetor com a submatriz de tamanho 183×174 correspondente ao íntron 1 e assim geramos então o íntron 1, verificando a sua igualdade com os elementos da palavra-código v_1.

Para encontrar o éxon 2 o processo foi análogo ao íntron 1, encontramos na matriz geradora a parte onde se localizava a informação do éxon 2, neste caso da linha 218 até a linha 502, da coluna 227 até a coluna 511, sendo uma submatriz de tamanho 285×285. Após encontrar a parte do éxon 2 na matriz geradora, podemos perceber que o número de linhas da submatriz será igual ao número de elementos do vetor u_1 referente ao éxon 2. Assim multiplicamos módulo 4 os elementos de 218 a 502 do vetor u_1 pela submatriz referente ao éxon 2 e obtemos como resultado os elementos da palavra-código v_1 referentes ao éxon 2.

Podemos encontrar o éxon 2 usando o grau do polinômio, sabemos que do elemento 227 até o elemento 502 tinha uma parte da informação do éxon 2 no vetor u_1 mas, ainda faltava uma parte de informação para que pudéssemos gerar o éxon 2, então acrescentamos os nove dígitos anteriores ao elemento 227 do vetor u_1, que exatamente o grau do polinômio gerador, assim a parte do éxon 2 no vetor u_1 se encontra do elemento 218 ao 502 sendo multiplicado módulo 4 pela submatriz com a informação do éxon 2 gerando assim a informação do éxon 2 na palavra-código v_1.

Na Tabela 4.17 podemos visualizar onde se encontra os 2 éxons e o íntron no vetor u_1, as partes em verde são comuns aos éxons e ao íntron, a parte em roxo mostra a localização do éxon 1 a primeira parte em verde completa a informação do éxon 1, a informação do íntron 1 esta compreendida nas duas partes em verde e na parte em azul, já a informação do éxon 2 está localizada na segunda parte em verde e na parte em vermelho, com isso podemos mostrar no vetor u_1 como que os éxons e o íntron estão localizados.

Sob o ponto de vista do vetor sinalização (vetor u), notamos que existem componentes deste vetor que são comuns tanto a éxons como a íntrons, mostrando uma forte ligação na região de fronteira. Uma intrepretação biológica que fazemos do vetor sinalização u_1 é a de realizar a localização/identificação no DNA da sequência precursora do RNA, pré-RNA. O próximo passo é a obtenção do mRNA associado ao correspondente gene. Para isso, é necessário que o mecanismo de splicing do pré-mRNA entre em ação. Isto por sua vez implica que a maquinaria de splicing deve reconhecer três regiões na molécula precursora do RNA: a região de splicing 5', a região de splicing 3' e o ponto da forquilha na sequência do íntron que forma a base do fragmento em laço a ser excisado. Cada um desses três sítios tem uma sequência nucleotídica consenso, que é similar entre os íntrons e que fornece a posição onde deve ocorrer o splicing.

Olhando a questão do splicing alternativo no caso do gene Trav7, podemos visualizar que o éxon 1 começa com o códon de inicialização **ATG** e termina com alanina **GCT**, como o éxon 1 não tem o códon de finalização, sozinho não gera proteína. O íntron 1 começa com valina **GTA** e termina com um códon de finalização **TAG**, logo a união do éxon 1 com o íntron 1 é possível. O éxon 2 começa com uma glicina **GGG** e termina com metionina **ATG**, mas antes deste códon temos o códon de finalização **TAG**. Do ponto de vista biológico temos duas possibilidade de

Tabela 4.17: Vetor u_1 separado em éxons e íntrons do gene Trav7

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

splicing alternativo sendo a primeira possibilidade: o éxon 1 com o íntron 1, sendo mostrada a concatenação dos vetores na Tabela 4.18 e a outra possibilidade é o éxon 1 com o éxon 2, sendo mostrada a concatenação dos vetores na Tabela 4.19.

Tabela 4.18: Primeiro caso de splicing alternativo do gene Trav7

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

No éxon 1 temos comprimento 52, na formação dos códons os nucleotídeos são agrupados 3 a 3, neste caso quando houver este agrupamento irá sobrar um nucleotídeo, que pode ser deletado

Tabela 4.19: Segundo caso de splicing alternativo do gene Trav7

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

no processo para que ocorra a formação dos códons. Se no splicing alternativo houver a união de éxon 1 com íntron 1, continuará sobrando um nucleotídeo, o qual poderá ser deletado. No caso da união do éxon 1 com o éxon 2, irá sobrar um nucleotídeo, que poderá ser deletado para que ocorra a formação dos códons necessários para a geração da proteína.

Podemos observar que após a localização de éxons e íntrons na matriz geradora G do gene Trav7, parte dos deslocamentos cíclicos ficam no éxon e a outra parte fica no íntron, isso define uma dependência entre éxons e íntrons e a existência de uma memória unitária parcial presente neste processo. Esta memória unitária parcial é descrita por Lauer em [67], como sendo o k_0 n-upla e denotado por a_t o sub-bloco de informações no instante t, com $t = 0, 1, \cdots$, em (n_0, k_0) do código convolucional binário. Seja n_0 uma n-upla binária denotada por b_t, o sub-bloco codificado no tempo t. Assim a equação de codificação pode ser escrita da seguinte forma:

$$b_t = a_t G_0 + a_{t-1} G_1 + \cdots + a_{t-m} G_m \tag{4.14}$$

Onde cada G_i é uma matriz binária $n_0 \times k_0$ com $G_m \neq 0$, onde M é a memória do codificador, onde, por meio de convenção, $a_t = 0$, para $t < 0$. Segundo [68], pode ser mostrado que $(n' = Mn_0, k' = Mk_0)$ com um codificador convolucional $M' = 1$ sendo definido por:

$$G'_0 = \begin{bmatrix} G_0 & G_1 & \cdots & G_{M-1} \\ 0 & G_0 & \cdots & G_{M-2} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & G_0 \end{bmatrix} \qquad G'_1 = \begin{bmatrix} G_M & 0 & \cdots & 0 \\ G_{M-1} & G_M & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ G_1 & G_2 & \cdots & G_M \end{bmatrix} \tag{4.15}$$

O código mostrado na Relação 4.14 é equivalente no sentido de que a mesma sequência binária semi-infinita codificada está associada com a mesma sequência semi-infinita de entrada. De acordo com [68], estes códigos são denominados códigos de memória unitária, com distancia máxima livre (d_{free}) para um dado k' e $R = k'/n'$.

Outro detalhe que podemos notar é que quanto maior o comprimento do gene maior é a dependência, pois terá um maior número de deslocamentos cíclicos com uma parte no éxon e outra parte no íntron, onde podemos inferir que existe uma dependência dos éxons e íntrons no splicing alternativo. Esta dependência de éxons e íntrons é mais forte entre os éxons vizinhos ao íntron. Um éxon depende mais de um íntron vizinho do que de um íntron que não seja seu vizinho, sendo a influência deste íntron não vizinho bem menor. Podemos notar ainda que o íntron tem um papel fundamental na relação de informação entre éxons.

Sob o ponto de vista da matriz geradora G, o espaço vetorial gerado tem dimensão 502. Todavia, a dimensão de cada subespaço correspondente ao éxon 1, íntron 1 e éxon 2 apresenta os seguintes valores 52, 183, 285. Note que a soma dessas dimensões vale 520, portanto ultrapassando o valor 502. Isso implica que o espaço total não é uma soma direta dos correspondentes subespaços. Mais ainda, estabelece uma dependência entre os subespaços vizinhos. Essa dependência entre subespaços vizinhos nada mais é que uma memória associada. Biologicamente podemos inferir que um íntron estabelece um processo de "amarramento"entre os éxons subsequentes e que se mostram importantes tanto no aspecto da realização do splicing alternativo como no da confiabilidade. Ambos processos de vital importância para a conservação da espécie.

4.2.2 Gene Hint-1

O gene Hint-1 do nematoide Caenorhabditis Elegans postado no banco de dados biológicos NCBI é identificado pelo "geneID"de número **184760** sendo mostrado na Figura 4.2. Os nucleotídeos em vermelho fazem parte do éxon 1 com comprimento 123, os nucleotídeos em verde fazem parte do íntron 1 com comprimento 44, a parte em azul são os nucleotídeos do éxon 2 com comprimento 138, a parte em negrito são os nucleotídeos do íntron 2 com comprimento 74, a parte em roxo são os nucleotídeos do éxon 3 com comprimento 132, totalizando um comprimento de 511 nucleotídeos para o gene Hint-1.

atgtcggaagtagataaagcccacttggcggcaattaacaaagatgttcaagccaacgacactcttttcggaaaaataattcga
aaagagattccagcgaaaatcatttttgaagatgatgaggtatgtaagatcaggcaaacgatcacataaaatatttatttaaggct
ctcgcattccatgatgtctctccacaagctccaattcattttcttgtgatccctaagcgtcgcattgatatgctcgagaatgccgttg
attcggatgctgcgcttattggaaagcttatggttactgcttcaaaggtaattataaatgagtaaaaacgaatttgaaaatccagaa
atctccattatattactcttaaataaaattccaggttgcaaagcagctcggcatggccaatggataccgtgttgttgtgaacaatgg
aaaagatggagctcaatcagttttccatcttcatctccacgttttgggaggacgtcagctccaatggccacctggataa

Figura 4.2: Sequência em nucleotídeos do gene Hint-1

Dada esta sequência, identificamos esta sequência como uma palavra-código, via o algoritmo de geração de sequências de DNA usando códigos BCH sobre anéis proposto por [4]- [3]- [65] e [66]. Identificada a palavra-código obtemos através dos procedimentos delineados no Capítulo 3 o correspondente polinômio gerador dado por $g(x) = 1x^9 + 2x^7 + 1x^5 + 3$, rótulo caso 1, bem como a matriz geradora com 502 linhas e 511 colunas mostrada abaixo:

$$G = \begin{bmatrix}
3 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 1 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 1
\end{bmatrix}$$

A Tabela 4.20 ilustra a correspondente palavra-código w_1.

Tabela 4.20: Palavra-código w_1 do gene Hint-1

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

Na matriz geradora podemos encontrar onde estão cada éxon e íntron olhando as colunas e relacionado com os nucleotídeos do gene, já que a matriz possui 511 colunas encontramos facilmente onde estão cada éxon e íntron. Após identificar onde se localiza a informação de cada éxon e de cada íntron na matriz geradora, podemos notar que estas informações são submatrizes, no caso do gene Hint-1 com tamanho diferentes. Além disso notamos que os subespaços não são independentes e portanto não sendo uma soma direta de subespaços. Quando encontramos as submatrizes correspondentes a éxons e íntrons percebemos que os deslocamentos ciclícos ficam divididos entre éxons e íntrons como podemos visualizar na Tabela 4.2 no exemplo mostrado no começo do capítulo.

Na palavra-código w_1 mostrada na Tabela 4.21 temos a localização de onde se encontra a informação de cada éxon e íntron, podemos visualizar que o éxon 1 são os elementos na cor vermelha, o íntron 1 são os elementos na cor verde, o éxon 2 são os elementos na cor roxa, o íntron 2 são os elementos na cor azul e o éxon 3 são os elementos em negrito.

Tabela 4.21: Palavra-código w_1 separada em éxons e íntrons do gene Hint-1

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

O próximo passo é determinar o vetor informação y_1 que multiplicado pela matriz geradora G resulta na palavra-código w_1 ou equivaletemente $(y_0 \quad y_1 \quad y_2 \cdots y_{501}).G = (w_0 \quad w_1 \quad w_2 \cdots w_{510})$. Considere os processos pelos quais foram obtidos os vetores u e u_1 mostrados anteriormente, assim temos que os elementos do vetor y_1 são determinados da seguinte forma:

$$w_0 = y_0 \cdot 3 + y_1 \cdot 0 + y_2 \cdot 0 + \cdots + y_{501} \cdot 0 + y_{501} \cdot 0$$

$$w_1 = 0 \cdot 0 + y_1 \cdot 3 + y_2 \cdot 0 + \cdots + y_{501} \cdot 0 + y_{501} \cdot 0$$

$$w_2 = 0 \cdot 0 + 1 \cdot 0 + y_2 \cdot 3 + \cdots + y_{501} \cdot 0 + y_{501} \cdot 0$$

$$\vdots$$

$$w_{510} = 0 \cdot 0 + 1 \cdot 0 + 2 \cdot 0 + \cdots + 1 \cdot y_{501}$$

Fazendo todos os cálculos para os 502 elementos do vetor y_1, encontramos todos os componentes do vetor que são mostrados na Tabela 4.22. Depois de encontrado o vetor y_1 fizemos y_1.G para verificar a igualdade com a palavra-código w_1, em seguida fizemos mais alguns testes para verificação dos dados como, a multiplicação módulo 4 da matriz geradora pela sua correspondente H transposta, multiplicamos módulo 4 a palavra-código pela H transposta para verificar se a síndrome era 0.

Tabela 4.22: Vetor y_1 referente ao gene Hint-1

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

Encontrado o vetor y_1 correspondente a palavra-código w_1 para o gene Hint-1 o processo para determinar éxons e íntrons é análogo ao realizado para o gene Trav7, a diferença do gene Hint-1 para o gene Trav7 é o número de éxons e íntrons. Para determinar o éxon 1, separamos o 123 primeiros elementos do vetor y_1 e multiplicamos módulo 4 pela informação do éxon 1 na matriz geradora sendo uma submatriz de tamanho 123×123, localizada da linha 1 até a linha 123 e da coluna 1 até a coluna 123, encontrando como resultado os valores referente ao éxon 1 na palavra-código w_1.

Para determinar onde se encontra a informação do íntron 1 no vetor y_1, usamos o processo análogo ao gene analisado anteriormente, olhamos o grau do polinômio gerador que no caso é 9, então a informação referente ao íntron 1 começa 9 elementos antes do elemento 124. Neste

caso a informação do íntron 1 no vetor y_1 começa no elemento 115 e vai até o elemento 167. Fazendo a multiplicação módulo 4 da parte do vetor y_1 pela informação do íntron 1 contida na matriz geradora, sendo uma submatriz de tamanho 53×44, localizada da linha 115 até a linha 167 e da coluna 124 até a coluna 167, obtendo com o resultado os valores referentes ao íntron 1 na palavra-código w_1.

Fazendo uma outra abordagem podemos utilizar a submatriz de tamanho 53×44 referente ao íntron 1, localizada da linha 115 até a linha 167 e da coluna 124 até a coluna 167, para encontrar a parte do íntron 1 no vetor y_1. Verificamos quais são as linhas que fazem parte do íntron 1 na matriz geradora, assim utilizamos o mesmo número de elementos do vetor y_1, neste caso devemos utilizar os elementos do vetor y_1 de 115 até 167. Efetuando a multiplicação módulo 4 dos elementos correspondentes ao íntron 1 do vetor y_1 pela submatriz referente ao íntron 1, obtemos como resultado a informação do íntron 1 na palavra-código w_1.

No caso do éxon 2, a informação no vetor y_1 pode ser obtida olhando o grau do polinômio gerador, que neste caso é 9, sabemos que uma parte da informação do y_1 está entre os elementos 168 a 305, considerando o grau do polinômio gerador, então a informação do éxon 2 no vetor y_1 começa 9 elementos antes do 168, e termina no elemento 305, assim a parte é composta do elemento 159 ao elemento 305. Efetuamos a multiplicação módulo 4 destes elementos do vetor y_1 pela parte da matriz geradora que contém a informação do éxon 2, sendo a submatriz de tamanho 147×138, localizada da linha 159 até a linha 305 e da coluna 168 até coluna 305 da matriz geradora, encontrando como resultado os valores referentes ao éxon 2 na palavra-código w_1.

Podemos gerar o éxon 2 usando as informações da submatriz correspondente ao éxon 2, neste caso esta submatriz está localizada da linha 159 até a linha 305 e da coluna 168 até coluna 305 na matriz geradora. Olhando a quantidade de linhas que fazem parte desta submatriz podemos perceber que serão a mesma quantidade de elementos do vetor y_1, então os elementos do vetor y_1 correspondentes ao éxon 2 são os elementos de 159 ao elemento 305, efetuando a multiplicação módulo 4 destes elementos pela submatriz correspondente ao éxon 2, obtemos como resultado a informação do éxon 2 contida na palavra-código w_1.

No caso do íntron 2, a informação no vetor y_1 pode ser obtida olhando o grau do polinômio gerador, que neste caso é 9, sabemos que uma parte da informação do y_1 esta entre os elementos 306 a 379, considerando o grau do polinômio gerador, então a informação do íntron 2 no vetor y_1 começa 9 elementos antes do 306, e termina no elemento 305, assim a parte é composta do elemento 297 ao elemento 379. Efetuamos a multiplicação módulo 4 destes elementos do vetor y_1 pela parte da matriz geradora que contém a informação do íntron 2, sendo a submatriz de tamanho 83×74, localizada da linha 297 até a linha 379 e da coluna 306 até coluna 379 da matriz geradora, encontrando como resultado os valores referentes ao íntron 2 na palavra-código w_1.

Podemos gerar o íntron 2 usando as informações da submatriz correspondente ao íntron 2, neste caso esta submatriz está localizada da linha 297 até a linha 379 e da coluna 306 até coluna 379 na matriz geradora. Olhando a quantidade de linhas que fazem parte desta submatriz podemos perceber que serão a mesma quantidade de elementos do vetor y_1, então os elementos do vetor y_1 correspondentes ao íntron 2 são os elementos de 297 ao elemento 379, efetuando a

multiplicação módulo 4 destes elementos pela submatriz correspondente ao íntron 2, obtemos como resultado a informação do íntron 2 contida na palavra-código w_1.

No caso do éxon 3, a informação no vetor y_1 pode ser obtida olhando o grau do polinômio gerador, que neste caso é 9, sabemos que uma parte da informação do y_1 esta entre os elementos 380 a 502, considerando o grau do polinômio gerador, então a informação do éxon 3 no vetor y_1 começa 9 elementos antes do 380, e termina no elemento 502, assim a parte é composta do elemento 371 ao elemento 502. Efetuamos a multiplicação módulo 4 destes elementos do vetor y_1 pela parte da matriz geradora que contém a informação do éxon 3, sendo a submatriz de tamanho 132×132, localizada da linha 371 até a linha 502 e da coluna 380 até coluna 511 da matriz geradora, encontrando como resultado os valores referentes ao éxon 3 na palavra-código w_1.

Podemos gerar o éxon 3 usando as informações da submatriz correspondente ao éxon 3, neste caso esta submatriz está localizada da linha 371 até a linha 502 e da coluna 380 até coluna 511 na matriz geradora. Olhando a quantidade de linhas que fazem parte desta submatriz podemos perceber que serão a mesma quantidade de elementos do vetor y_1, então os elementos do vetor y_1 correspondentes ao éxon 3 são os elementos de 371 ao elemento 502, efetuando a multiplicação módulo 4 destes elementos pela submatriz correspondente ao éxon 3, obtemos como resultado a informação do éxon 3 contida na palavra-código w_1.

O detalhamento de éxons e íntrons no vetor y_1 é mostrado na Tabela 4.23, em que os elementos em preto são partes comuns de éxons e íntrons, a parte em vermelho combinada com a parte preta comum ao íntron 1 são os elementos que geram o éxon 1, a parte em verde combinada com a parte preta comum ao éxon 1 e a parte preta comum ao éxon 2 geram o íntron 1, a parte em roxo combinada com a parte preta comum ao íntron 1 e com a parte preta comum ao íntron 2 geram o éxon 2, a parte em azul combinada com a parte preta comum ao éxon 2 e com a parte preta comum ao éxon 3 geram o íntron 2, a parte em negrito combinada com a parte preta comum ao íntron 2 geram o éxon 3.

Sob o ponto de vista do vetor sinalização (vetor u), notamos que existem componentes deste vetor que são comuns tanto a éxons como a íntrons, mostrando uma forte ligação na região de fronteira. Uma intrepretação biológica que fazemos do vetor sinalização y_1 é a de realizar a localização/identificação no DNA da sequência precursora do RNA, pré-RNA. O próximo passo é a obtenção do mRNA associado ao correspondente gene. Para isso, é necessário que o mecanismo de splicing do pré-mRNA entre em ação. Isto por sua vez implica que a maquinaria de splicing deve reconhecer três regiões na molécula precursora do RNA: a região de splicing 5', a região de splicing 3' e o ponto da forquilha na sequência do íntron que forma a base do fragmento em laço a ser excisado. Cada um desses três sítios tem uma sequência nucleotídica consenso, que é similar entre os íntrons e que fornece a posição onde deve ocorrer o splicing.

Fazendo uma análise dos possiveis casos de splicing alternativo notamos que o éxon 1 começa com **ATG** e termina com **GAG**, assim ele possui o start códon porém não possui o stop códon assim, o éxon 1 sozinho não é capaz de gerar proteína. O íntron 1 começa com o códon **GTA** e termina com o códon **AAG**, assim o íntron 1 não possui start códon nem stop códon. O éxon 2 começa com o códon **GCT** e termina com o códon **AAG**, assim o éxon 2 não possui start códon nem stop códon portanto não gera proteínas. O íntron 2 começa com o códon **GTA** e

Tabela 4.23: Vetor y_1 separado em éxons e íntrons do gene Hint-1

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

termina com o códon **CAG**, assim o íntron 2 não possui start códon nem stop códon. O éxon 3 começa com o códon **GTT** e termina com o códon **TAA**, o éxon 3 não possui start códon mas, possui stop códon.

Como o start códon está no éxon 1 e o stop códon esta no éxon 3 podemos ter as seguintes combinações entre éxons e íntrons: a primeira possibilidade de geração de proteínas é a composição do éxon 1, éxon 2, íntron 2, e éxon 3. A segunda possibilidade é a composição de todos os éxons e íntrons. A terceira possibilidade é a composição de éxon 1 com éxon 3. A quarta possibilidade é a composição de éxon 1, éxon 2 e éxon 3 e a quinta possibilidade é a composição de éxon 1, íntron 1, éxon 2 e éxon 3. Do ponto de vista biológico as demais combinações não são possiveis. Considerando as possibilidades de splicing alternativo do gene Hint-1 podemos observar que a partir de único gene é possível gerar cinco proteínas diferentes, isso explica em parte a enorme diferença entre o tamanho modesto do conjunto de genes do Caenorhabditis Elegans e a elevada capacidade proteomica.

Após gerar os éxons e íntrons do gene Hint-1 e verificar as possibilidades de composição do splicing alternativo podemos realizar matematicamente cada um deste casos. Cada éxon e íntron é identificado por um vetor, assim cada uma destas combinações podem ser feitas através de uma concatenação de vetores. Podemos assim fazer uma possível modelagem matemática para o splicing aternativo no gene Hint-1, sendo a primeira possibilidade mostrada na Tabela 4.24, a segunda possibilidade mostrada na Tabela 4.25, a terceira possibilidade mostrada na Tabela

4.26, a quarta possibilidade mostrada na Tabela 4.27 e a quinta possibilidade é mostrada na Tabela 4.28.

Tabela 4.24: Primeiro caso de splicing alternativo do gene Hint-1

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

Tabela 4.25: Segundo caso de splicing alternativo do gene Hint-1

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

Tabela 4.26: Terceiro caso de splicing alternativo do gene Hint-1

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

Tabela 4.27: Quarto caso de splicing alternativo do gene Hint-1

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

Tabela 4.28: Quinto caso de splicing alternativo do gene Hint-1

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

Podemos observar que após a localização de éxons e íntrons na matriz geradora G do gene Hint-1, parte dos deslocamentos cíclicos ficam no éxon e a outra parte fica no íntron, isso define uma dependência entre éxons e íntrons e a existência de um código de memória unitária parcial no processo, sendo este código convolucional de memória parcial unitária descrito na Subseção

4.2.1.

Outro detalhe que podemos notar é que quanto maior o comprimento do gene maior será a dependência, pois terá o maior número de deslocamentos cíclicos com uma parte no éxon e outra parte no íntron, o que pode no direcionar a pensar que existe uma dependência dos éxons e íntrons no splicing alternativo. Esta dependência de éxons e íntrons é mais forte em um íntron vizinho do éxon. Um éxon depende mais de um íntron vizinho do que de um íntron que não seja seu vizinho, sendo a influência deste íntron não vizinho bem menor. Podemos notar ainda que o íntron tem um papel fundamental na relação de informação entre éxons.

Sob o ponto de vista da matriz geradora G, o espaço vetorial gerado tem dimensão 502. Todavia, as dimensões dos subespaços correspondentes aos éxon 1, íntron 1, éxon 2, íntron 2, e éxon 3 apresentam os seguintes valores 123, 53, 147, 83, 132. Note que a soma dessas dimensões vale 538, portanto ultrapassando o valor 502. Isso implica que o espaço total não é uma soma direta dos correspondentes subespaços. Mais ainda, estabelece uma dependência entre os subespaços vizinhos. Essa dependência entre subespaços vizinhos nada mais é que uma memória associada. Biologicamente podemos inferir que um íntron estabelece um processo de "amarramento"entre os éxons subsequentes e que se mostram importantes tanto no aspecto da realização do splicing alternativo como no da confiabilidade. Ambos processos de vital importância para a conservação da espécie.

4.3 Modelo para a geração de partes de um genoma

Como no caso dos genes foi gerado os éxons e íntrons separadamente, resolvemos tentar a geração de partes de um genoma. Assim escolhemos o genoma do Plasmídeo contido no banco dados biológicos NCBI, com o GI número **118213250**. Este Genoma possui 2047 nucleotídeos, obedecendo ao comprimento do código $n = 2^r$*-1*, sendo dividido em nove regiões e mostradas na Figura 4.3.

As partes da sequência genômica que estão na cor vermelho escuro, são sequências cuja funcionalidade biológica ainda é desconhecida, mas, já são conhecidos o início e o fim destas sequências, sendo elas: a primeira região é composta por 715 nucleotídeos, a terceira região é composta por 104 nucleotídeos, a quinta região é composta por 12 nucleotídeos e a nona região é composta por 113 nucleotídeos. A segunda região na cor verde claro é composta por 168 nucleotídeos onde é identificada pela origem da replicação de uma fita do DNA, a quarta região na cor azul claro é composta por 129 nucleotídeos sendo identificada pela origem da replicação da fita dupla do DNA, a sexta região na cor roxa é composta por 138 nucleotídeos sendo identificada pelo gene "Cob G". A sétima região na cor verde escuro é composta de 76 nucleotídeos sendo identificada por um RNA, a oitava região na cor laranja é composta por 612 nucleotídeos sendo identificada pelo gene "Rep B".

Uma importante observação é feita na parte da sequência em que os nucleotídeos estão na cor preta, pois são locais que pertencem a duas partes diferentes no genoma, os nucleotídeos **aa** pertencem tanto a sexta região como a sétima região do genoma, e os nucleotídeos **atgacagaa-aaaaaacta** fazem parte da sétima e oitava região.

Dada a sequência genômica identificamos esta sequência como uma palavra-código, via algo-

cctacatttttttattgctctgctatgattgtttatcgatagtttttatacagataagcgtgcgacgcttgctctttccgaggaggaagt
catgctgacaagcacggcagagcctccgcatgaaatgctctcaatgaaattgccggcggagctttttgagcttgtgccacttg
cgaaaaaaacaagaacaaaagagacaggaaactgtctttttttgcttgcttggggattggggcaacgccccaaaaataaaaag
aatcgtctgaaacgaggaacaaactaaaatgtaaatttagttgttaccgagtggaagatgaatacttttaacctatgtgtataca
cacatagtaagctcgctataatactttataacgtttttatttacatgagcaaagcgagtttttccaacacgtttaatctaaaatattggc
aatttataccatgattttcatggtatgtaagtgcgcccttaggaaaataatttgaatatatttcagattttcaatctgactgctcctgtca
tcgagcagaccgatgaggaaaacaaaaagaggactaaacaaaaaagtttagtcctcttttgttttgaatagttctagaacgtcat
attttgcgttttaagcaattttgactaactaggcggggatttttacttagaaattattcaaaacgtctgtaaagtgcttaaaatcgtttct
aagagcttttagcgtttatttcgtttagttatcggcataatcgttaaaacaggcgttatcgtagcggaaaagcccttgagcgtagcg
tggctttgcagtgaagatgttgtctgttagattatgaaagccgataactgaatgaaataataagcgtagcgcccctttatttcggtcg
gaggaggctcaagggagtttgagggaatgaaattccctcatggttttaaaattgcttgcaattttgccgagcggtagcgctggaa
aatttttgaaaaaaatttggaatttggaaaaatgggggggtactacgacccccccctatgtggtaatttggtaacttggtcaaaatt
gatactaatatatattaaaacagcacaaaacagaatcttatgatataataagatatactgaaatttgaaggagtaaaaaatggcag
aagagaaaaaaaagagttttgctaactttgtcgttggacaaagcagaagaattagaaactatatcaaaagaaatgggaattagtaa
atctgctcttgttagtttatggattgcggaaaattctagaaaataaaaaaagagccacggcgaatggctctagtatatttacggtta
ggaatattatagcatatgacagaaaaaaaactagaaaaaatgacccagttagaaactggagttgggttgtttatccagagtctg
ctcctgaaaattggagaacattgttagacgaaactggagaaaaatggattgagagtccgttgcatgataaagatattaacgaaa
caacaaacgaaccgaaaaaggcacattggcatataataatttcttttcaaataaaaaaagttataagcaagtattaaaaatttctg
aaatgttaaatgcaccagagcctgtaaaaacaaaaaatttacaagggtcagttcaatatttgtggcacagaaacaatcctgaaaa
atatcagtataataaaagcgatgttgttgctcataatgggtttaaatatagacaatatttaacagatattggagttgatactgattctat
tttacaagaagttatagaatggataaaagaaactggatgttctgaatatagagatttagtcgattatgcagtatcagaacgtttcga
tgattggtttcctacagtcagaagtcaaaccatatttttaaattcttatttacgctcaaatcgtcatagtcagaaaaaatataatccag
aaacaggagaggtgttatgaaagttgaaattatagctagtgtttttagtgaaaaatcagttcagaaaaaagtaaataatttattgat
tatttaaatgacaataattttgaagtattggaagttcaatatagg

Figura 4.3: Sequência em nucleotídeos do genoma do Plasmídeo

ritmo de geração de sequências de genômicas usando códigos BCH sobre anéis proposto por [4]-[3]- [65] e [66]. Identificado a palavra-código obtemos através dos procedimentos delineados no Capítulo 3 o correspondente polinômio gerador dado por $g(x) = 1x^{11} + 3x^{10} + 2x^9 + 1x^7 + 2x^6 + 2x^5 + 3x^2 + 2x^1 + 3$ sendo rotulada no caso 1, bem como a matriz geradora com 2036 linhas e 2047 colunas mostrada abaixo:

$$G = \begin{bmatrix}
3 & 2 & 3 & 0 & 0 & 2 & 2 & 1 & 0 & 2 & 3 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3 & 2 & 3 & 0 & 0 & 2 & 2 & 1 & 0 & 2 & 3 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 3 & 2 & 3 & 0 & 0 & 2 & 2 & 1 & 0 & 2 & 3 & 1 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 3 & 2 & 3 & 0 & 0 & 2 & 2 & 1 & 0 & 2 & 3 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 3 & 2 & 3 & 0 & 0 & 2 & 2 & 1 & 0 & 2 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 3 & 0 & 0 & 2 & 2 & 1 & 0 & 2 & 3 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 2 & 3 & 0 & 0 & 2 & 2 & 1 & 0 & 2 & 3 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 3 & 2 & 3 & 0 & 0 & 2 & 2 & 1 & 0 & 2 & 3 & 1
\end{bmatrix}$$

As Tabelas 4.29 e 4.30 ilustram a correspondente palavra-código d_1.

O próximo passo é determinar o vetor informação c_1 que multiplicado pela matriz geradora G resulta na palavra-código d_1 ou equivaletemente $(c_0 \quad c_1 \quad c_2 \cdots c_{2036}).G = (d_0 \quad d_1 \quad d_2 \cdots d_{2047})$. Considere os processos pelos quais foram obtidos os vetores u, u_1 e y_1 mostrados anteriormente, assim temos que os elementos do vetor c_1 são determinados da seguinte forma:

$$d_0 = c_0 \cdot 3 + c_1 \cdot 0 + c_2 \cdot 0 + \cdots + c_{2035} \cdot 0 + c_{2035} \cdot 0$$

Tabela 4.29: Parte 1 da palavra-código d_1 do Genoma do Plasmídeo

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

$$d_1 = 3 \cdot 2 + c_1 \cdot 3 + c_2 \cdot 0 + \cdots + c_{2035} \cdot 0 + c_{2035} \cdot 0$$

$$d_2 = 3 \cdot 3 + 2 \cdot 1 + c_2 \cdot 3 + \cdots + c_{2035} \cdot 0 + c_{2035} \cdot 0$$

$$\vdots$$

Tabela 4.30: Parte 2 da palavra-código d_1 rótulo caso 1 do Genoma do Plasmídeo

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

$$d_{2046} = 3 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 + \cdots + 1 \cdot c_{2035}$$

Fazendo todos os cálculos para os 2036 elementos do vetor c_1, encontramos todos os componentes do vetor que são mostrados nas Tabelas 4.31 e 4.32. Depois de encontrado o vetor c_1 fizemos c_1.G para verificar a igualdade com a palavra-código d_1, em seguida fizemos mais alguns

testes para verificação dos dados como, a multiplicação módulo 4 da matriz geradora pela sua correspondente H transposta, multiplicamos módulo 4 a palavra-código pela H transposta para verificar se a síndrome era 0.

Tabela 4.31: Parte 1 vetor c_1 do Genoma do Plasmídeo

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

Depois de encontrados todos os dados referentes ao genoma do Plasmídeo fizemos os mesmos cálculos já feitos anteriormente para genes, para verificação do seu comportamento em relação a gerar partes deste genoma. Nossa motivação era mostrar que podemos gerar partes

Tabela 4.32: Parte 2 vetor c_1 do Genoma do Plasmídeo

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

de um genoma da mesma forma em que geramos éxons e íntrons. A primeira região do genoma é composta por 715 nucleotídeos, então separamos os 715 primeiros elementos no vetor c_1 e multiplicamos módulo 4 pela submatriz oriunda da matriz geradora de tamanho 715x715, referente a primeira região, como resultado encontramos a informação referente a primeira região do genoma na palavra-código d_1.

A segunda região é composta por 168 nucleotídeos, para gerar a segunda região, encontramos a parte referente a segunda região no vetor c_1 como o polinômio gerador tem grau 12, então a

informação esta localizada 12 elementos antes do 716, sendo do elemento 704 até o elemento 883 e multiplicamos módulo 4 pela submatriz de tamanho 180×168 oriunda da matriz geradora, referente a segunda região, assim encontramos com resultado a informação referente a segunda região do genoma na palavra-código d_1. A terceira região é composta por 104 nucleotídeos, para encontramos esta região é necessário encontrar a parte referente a terceira região no vetor c_1, sendo a informação iniciada no elemento 872 e terminando no elemento 987, assim fazemos a multiplicação módulo 4 pela submatriz referente a terceira região de tamanho 116×104, oriunda da matriz geradora, resultando na informação da terceira região contida na palavra-código d_1.

A quarta região é composta por 129 nucleotídeos, a parte referente a quarta região no vetor c_1 começa no elemento 976 até o elemento 1116, identificada esta parte fazemos a multiplicação módulo 4 pela submatriz oriunda da matriz geradora, referente a quarta região com tamanho 141×129, resultando na informação da quarta região contida na palavra-código d_1. A quinta região é composta por 12 nucleotídeos, a parte referente a quinta região no vetor c_1 começa no elemento 1105 até o elemento 1128, localizada esta parte fazemos a multiplicação módulo 4 pela submatriz oriunda da matriz geradora, referente a quinta região com tamanho 23×12, obtemos como resultado a informação contida na quinta região da palavra-código d_1.

A sexta região é composta por 138 nucleotídeos, a parte referente a sexta região no vetor c_1 começa no elemento 1117 até o elemento 1266, encontrada esta parte fazemos a multiplicação módulo 4 pela submatriz oriunda da matriz geradora, referente a sexta região com tamanho 150×138, assim obtemos como resultado a informação contida na sexta região da palavra-código d_1. A sétima região é composta por 76 nucleotídeos, esta informação no vetor c_1 começa no elemento 1253 até o elemento 1340, localizada esta informação fazemos a multiplicação módulo 4 pela submatriz oriunda da matriz geradora, referente a sétima região com tamanho 88×76, resultando na informação contida na sétima região da palavra-código d_1.

A oitava região é composta por 612 nucleotídeos, a parte referente a oitava região no vetor c_1 começa no elemento 1311 até o elemento 1934, encontrada esta informação efetuamos a multiplicação módulo 4 pela submatriz oriunda da matriz geradora, referente a oitava região de tamanho 624×612, obtendo como resultado a informação contida na oitava região da palavra-código d_1. A nona região do genoma é composta por 113 nucleotídeos, a parte referente a nona região no vetor c_1 começa no elemento 1923 até o elemento 2047, encontrada esta informação multiplicamos módulo 4 pela submatriz oriunda da matriz geradora, referente a nona região de tamanho 125×113, assim obtemos como resultado a informação contida na nona região da palavra-código d_1.

4.4 Uso do código de Varshamov-Tenengolts

O código Varshamov-Tenengolts é usado para reconstruir sequências de dados em que ocorre uma única deleção, ou uma única inserção. Este código não é capaz de corrigir a combinação de deleção e inserção ao mesmo tempo, para reconstruir a sequência é necessário conhecer a sequência original em que houve a deleção ou a inserção. Neste caso consideremos os genes usados anteriormente, o gene Trav7 e Hint-1, onde realiazamos inserção/deleção de nucleotídeos em ambos os genes.

4.4.1 Gene Trav7

O gene Trav7 tem 511 nucleotídeos com dois éxons e um íntron, sendo este gene identificado como palavra-código de um código BCH, portanto sabemos que existe uma estrutura matemática associada ao gene Trav7. No presente trabalho esta sequência genética será identificada como palavra-código de um código de Varshamov-Tenengolts. Assim é possível mostrar que além da sequência genética estar associada a estrutura matemática do código BCH, esta sequência genética está associada a outros códigos, e um dos códigos que podemos associar a sequências genéticas é o código de Varshamov-Tenengolts.

A partir da identificação do gene Trav7 como palavra-código de um código BCH, vamos fazer uma deleção de informação nesta palavra-código e depois reconstruir esta informação usando o código de Varshamov-Tenengolts. Considere a palavra-código mostrada na Tabela 4.14 aqui chamado de vetor A. Para reconstruir uma informação em que houve uma deleção/inserção devemos calcular alguns parâmetros. O primeiro parâmetro que vamos determinar é o vetor α, e seu primeiro elemento α_1 pode ser qualquer símbolo binário. Consideramos $\alpha_1 = 1$,seja $q = 4$ e $n = 511$, então α_i é dado pela Relação 4.1 e o vetor binário resultante da Relação 4.1 é mostrado na Tabela 4.33.

Tabela 4.33: Vetor α obtido através do vetor A referente ao gene Trav7

1	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1
1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	0	1
1	1	0	1	0	1	1	1	1	0	0	1	1	1	1	1	0	1	0	1	1	0	1	1	1
0	1	1	1	0	1	1	1	1	0	1	1	0	1	0	1	0	1	1	1	0	1	1	0	0
1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1
0	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	1	1
0	1	1	1	0	0	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1	1	1	0	1
0	0	1	1	1	1	1	0	1	0	1	1	1	1	0	1	1	1	1	0	1	0	1	0	1
0	1	1	0	1	0	1	1	1	0	0	1	0	1	0	1	1	1	0	1	0	0	1	1	0
1	1	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	1	1	0	1
0	1	1	0	1	0	1	1	1	1	1	0	0	1	1	1	1	0	1	0	1	1	0	1	1
1	1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0
0	1	1	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	0	1	1	0	0	1	1
1	1	0	1	1	0	1	1	1	1	0	1	0	1	1	0	1	1	0	1	0	1	1	1	0
1	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1	1	0	1	1	1	0	1
1	1	0	1	1	0	1	0	1	0	1	0	1	0	1	1	0	0	1	1	1	0	1	0	1
0	1	0	0	1	0	1	1	1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1	1
1	0	1	1	0	1	0	1	1	0	1	1	0	0	1	1	0	1	1	0	1	0	1	1	1
0	1	1	1	1	0	1	0	1	0	1	1	0	1	0	1	1	1	0	1	0	1	0	1	1
1	1	0	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0
1	0	1	1	0	1	0	1	0	1	0														

Conhecidos o vetor A e o correspondente vetor α, logo podemos calcular os parâmetros β e γ, dados pela Relação 4.3.

Fazendo os cálculos encontramos:

$$\beta \equiv 2 \, mod \, 4 \tag{4.16}$$

e

$$\gamma \equiv 449 \, mod \, 511 \tag{4.17}$$

Determinados os parâmetros, β e γ simulamos uma deleção de um nucleotídeo qualquer, assim deletamos o nucleotídeo na posição 150, sendo um T com o rotulamento no caso 3, assim o número deletado foi 3, então criamos um vetor A'. Através do vetor A' encontramos o vetor α', de posse destes valores encontramos os parâmetros S_1, S_2 e W, necessários para a reconstrução da sequência original, permitindo uma única decodificação. S_1 é igual ao valor do símbolo perdido, W é o peso (número de símbolos diferentes de zero) da sequência α' e S_1, S_2 são os menores resíduos não negativos das congruências, o vetor A' é mostrado na Tabela 4.34 e o vetor α' é mostrado na Tabela 4.35.

Tabela 4.34: Vetor A' obtido após a deleção de informação referente ao gene Trav7

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

$$S_1 \equiv \beta - \sum_{i=1}^{n} a_i'(mod \, q) \equiv 3 \, (mod \, 4) \tag{4.18}$$

Tabela 4.35: Vetor α' obtido através do vetor A' referente ao gene Trav7

1	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1
1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	0	1
1	1	0	1	0	1	1	1	1	0	0	1	1	1	1	1	0	1	0	1	1	0	1	1	1
0	1	1	1	0	1	1	1	1	0	1	1	0	1	0	1	0	1	1	1	0	1	1	0	0
1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1
0	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	1	0
1	1	1	0	0	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1	1	1	0	1	0
0	1	1	1	1	1	0	1	0	1	1	1	1	0	1	1	1	1	0	1	0	1	0	1	0
1	1	0	1	0	1	1	1	0	0	1	0	1	0	1	1	1	0	1	0	0	1	1	0	1
1	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	1	1	0	1	0
1	1	0	1	0	1	1	1	1	1	0	0	1	1	1	1	0	1	0	1	1	0	1	1	1
1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0	0
1	1	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	0	1	1	0	0	1	1	1
1	0	1	1	0	1	1	1	1	0	1	0	1	1	0	1	1	0	1	0	1	1	1	0	1
1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1	1	0	1	1	1	0	1	1
1	0	1	1	0	1	0	1	0	1	0	1	0	1	1	0	0	1	1	1	0	1	0	1	0
1	0	0	1	0	1	1	1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1	1	1
0	1	1	0	1	0	1	1	0	1	1	0	0	1	1	0	1	1	0	1	0	1	1	1	0
1	1	1	1	0	1	0	1	0	1	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1
1	0	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0	1
0	1	1	0	1	0	1	0	1	0															

e

$$S_2 \equiv \gamma - \sum_{i=1}^{n}(i-1)\alpha'_i(mod\, n) \equiv 383\,(mod\, 511) \tag{4.19}$$

Fazendo o cálculo de W encontramos que W = 333 , assim $S_2 >$ W , portanto inserimos o símbolo 1 na sequência α' de modo que o número de zeros do lado esquerdo de onde o símbolo foi inserido seja igual a $S_2 - W$, neste caso como S_2=383 então após o quinquagésimo 0 acrescentamos o símbolo 1. Como o quinquagésimo 0 esta na posição 149, então inserimos o símbolo 1 na posição 150, este novo vetor aqui chamado de α'_1 é mostrado na Tabela 4.36.

Como S_1=3, então concluímos que o símbolo que foi excluído é o símbolo 3. Fazendo o procedimento descrito pelo código de Varshamov-Tenengolts é necessário colocar um símbolo 1 na posição 150 da sequência, assim temos uma única possibilidade de decodificação, mostrada na Tabela 4.37.

Como podemos observar a mensagem corrigida é igual a mensagem enviada. Com este resultado é possível notar que além do código BCH as sequências genéticas podem ser associadas a outros códigos, assim notamos que existem outras estruturas matemáticas envolvidas nos processos da biologia molecular, neste caso é notório que a estrutura do código de Varshamov-Tenengolts pode ser associada a sequências genéticas. Com a identificação dos códigos que podem estar associados aos processos biológicos teremos um pouco mais de facilidade para

Tabela 4.36: Vetor α_1' referente ao gene Trav7

1	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1
1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	0	1
1	1	0	1	0	1	1	1	1	0	0	1	1	1	1	1	0	1	0	1	1	0	1	1	1
0	1	1	1	0	1	1	1	1	0	1	1	0	1	0	1	0	1	1	1	0	1	1	0	0
1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1
0	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	1	1
0	1	1	1	0	0	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1	1	1	0	1
0	0	1	1	1	1	1	0	1	0	1	1	1	1	0	1	1	1	1	0	1	0	1	0	1
0	1	1	0	1	0	1	1	1	0	0	1	0	1	0	1	1	1	0	1	0	0	1	1	0
1	1	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	1	1	0	1
0	1	1	0	1	0	1	1	1	1	1	0	0	1	1	1	1	0	1	0	1	1	0	1	1
1	1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0
0	1	1	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	0	1	1	0	0	1	1
1	1	0	1	1	0	1	1	1	1	0	1	0	1	1	0	1	1	0	1	0	1	1	1	0
1	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1	1	0	1	1	1	0	1
1	1	0	1	1	0	1	0	1	0	1	0	1	0	1	1	0	0	1	1	1	0	1	0	1
0	1	0	0	1	0	1	1	1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1	1
1	0	1	1	0	1	0	1	1	0	1	1	0	0	1	1	0	1	1	0	1	0	1	1	1
0	1	1	1	1	0	1	0	1	0	1	1	0	1	0	1	1	1	0	1	0	1	0	1	1
1	1	0	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0
1	0	1	1	0	1	0	1	0	1	0														

entender o que de fato ocorre em animais, plantas e nos seres humanos.

Ocorrendo o splicing alternativo, haverá a união dos dois éxons ficando com 337 nucleotídeos, na formação dos códons os nucleotídeos são agrupados 3 a 3, logo o número de nucleotídeos após splicing alternativo tem que ser divisível por 3. Neste caso 337, não é divisível por 3 então é bem possível que ocorrerá uma deleção de nucleotídeo, assim o mRNA tem o comprimento 336, sendo um número divisível por 3. A partir disso resolvemos simular uma deleção de nucleotídeo e verificar se os mesmos resultados obtidos com outros tipos de fonte informação se assemelham com a fonte de informação genética, logo simulamos a deleção do nucleotídeo na posição 337 e usamos o código de Varshamov-Tenengolts para reconstruir a sequência . Temos o vetor de comprimento 337 chamado aqui de A e mostrado na Tabela 4.38.

Usando o código de Varshamov-Tenengolts vamos determinar o vetor α, e seu primeiro elemento α_1 pode ser qualquer símbolo binário. Consideramos α_1=1,seja $q = 4$ e $n = 337$, então α_i é dado pela Relação 4.1 e o vetor binário resultante da Relação 4.1 é mostrado na Tabela 4.39.

Conhecidos o vetor A e o correspondente vetor α, logo podemos calcular os parâmetros β e γ, dados pela Relação 4.3.

Tabela 4.37: Vetor A' referente ao gene Trav7

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

Tabela 4.38: Vetor A originado durante o splicing alternativo referente ao gene Trav7

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

Fazendo os cálculos encontramos:

$$\beta \equiv 2 \, mod \, 4 \tag{4.20}$$

Tabela 4.39: Vetor α gerado a partir do vetor A referente ao gene Trav7

1	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1
1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	0	1
1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	1	1	0
1	0	1	1	0	1	0	1	1	1	1	1	0	0	1	1	1	1	0	1	0	1	1	0	1
1	1	1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	0	1	1	1	0	1	0	1
0	0	1	1	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	0	1	1	0	0	1
1	1	1	0	1	1	0	1	1	1	1	0	1	0	1	1	0	1	1	0	1	0	1	1	1
0	1	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1	1	0	1	1	1	0
1	1	1	0	1	1	0	1	0	1	0	1	0	1	0	1	1	0	0	1	1	1	0	1	0
1	0	1	0	0	1	0	1	1	1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1
1	1	0	1	1	0	1	0	1	1	0	1	1	0	0	1	1	0	1	1	0	1	0	1	1
1	0	1	1	1	1	0	1	0	1	0	1	1	0	1	0	1	1	1	0	1	0	1	0	1
1	1	1	0	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	1	1	1	0	1
0	1	0	1	1	0	1	0	1	0	1	0													

e

$$\gamma \equiv 91 \, mod \, 337 \tag{4.21}$$

Determinados os parâmetros, β e γ simulamos uma deleção de nucleotídeo de forma que os nucleotídeos possam formar os códons, assim deletamos o nucleotídeo na posição 337, sendo um G com o rotulamento no caso 3, assim o número deletado foi 1, então criamos um vetor A'. Através do vetor A' encontramos o vetor α', de posse destes valores encontramos os parâmetros S_1, S_2 e W, necessários para a reconstrução da sequência original, permitindo uma única decodificação. S_1 é igual ao valor do símbolo perdido, W é o peso (número de símbolos diferentes de zero) da sequência α' e S_1, S_2 são os menores resíduos não negativos das congruências, o vetor A' é mostrado na Tabela 4.40 e o vetor α' é mostrado na Tabela 4.41.

$$S_1 \equiv \beta - \sum_{i=1}^{n} a_i'(mod \, q) \equiv 1 \, (mod \, 4) \tag{4.22}$$

e

$$S_2 \equiv \gamma - \sum_{i=1}^{n} (i-1)\alpha_i'(mod \, n) \equiv 0 \, (mod \, 337) \tag{4.23}$$

Fazendo o cálculo de W encontramos que W = 217 , assim S_2 < W , portanto inserimos o símbolo 0 na sequência α' de modo que o número de uns do lado direito de onde o símbolo foi inserido seja igual a S_2, neste caso como S_2=0 então não podemos ter nenhum símbolo 1 a direita do símbolo. Como o último 1 esta na posição 336, então inserimos o símbolo 0 na posição 337, este novo vetor aqui chamado de α_1' é mostrado na Tabela 4.42.

Com S_1=1, então concluímos que o símbolo que foi excluído é o 1, assim a única possibilidade é colocar o símbolo 1 na posição 337 da sequência, é mostrada na Tabela 4.43 a sequência corrigida.

Concluímos que a sequência corrigida é igual a sequência enviada.

Tabela 4.40: Vetor A' referente ao gene Trav7

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

Tabela 4.41: Vetor α' gerado a partir do vetor A' referente ao gene Trav7

1	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1
1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	0	1
1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	1	1	0
1	0	1	1	0	1	0	1	1	1	1	1	0	0	1	1	1	1	0	1	0	1	1	0	1
1	1	1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	0	1	1	1	0	1	0	1
0	0	1	1	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	0	1	1	0	0	1
1	1	1	0	1	1	0	1	1	1	1	0	1	0	1	1	0	1	1	0	1	0	1	1	1
0	1	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1	1	0	1	1	1	0
1	1	1	0	1	1	0	1	0	1	0	1	0	1	0	1	1	0	0	1	1	1	0	1	0
1	0	1	0	0	1	0	1	1	1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1
1	1	0	1	1	0	1	0	1	1	0	1	1	0	0	1	1	0	1	1	0	1	0	1	1
1	0	1	1	1	1	0	1	0	1	0	1	1	0	1	0	1	1	1	0	1	0	1	0	1
1	1	1	0	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	1	1	1	0	1
0	1	0	1	1	0	1	0	1	0	1														

4.4.2 Gene Hint-1

No caso do gene Hint-1 usamos o código de Varshamov-Tenengolts para fazer uma deleção de nucleotídeo e uma inserção de nucleotídeo e verificar se os mesmos resultados obtidos com outros tipos de fonte informação se assemelham com a fonte de informação genética. O gene Hint-1 possui 3 éxons e 2 íntrons, fazendo uma análise biológica podemos perceber que no éxon 1 temos o start códon mas, não temos o stop códon, assim ele sozinho não gera proteína, o éxon 2 não possui start códon nem stop códon, assim ele também não é capaz de gerar proteína sozinho, o éxon 3 não possui start códon, mas possui stop códon, ele também não gera proteína

Tabela 4.42: Vetor α'_1 gerado a partir do vetor α' referente ao gene Trav7

1	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1
1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	0	1
1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	1	1	0
1	0	1	1	0	1	0	1	1	1	1	1	0	0	1	1	1	1	0	1	0	1	1	0	1
1	1	1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	0	1	1	1	0	1	0	1
0	0	1	1	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	0	1	1	0	0	1
1	1	1	0	1	1	0	1	1	1	1	0	1	0	1	1	0	1	1	0	1	0	1	1	1
0	1	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1	1	0	1	1	1	0
1	1	1	0	1	1	0	1	0	1	0	1	0	1	0	1	1	0	0	1	1	1	0	1	0
1	0	1	0	0	1	0	1	1	1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1
1	1	0	1	1	0	1	0	1	1	0	1	1	0	0	1	1	0	1	1	0	1	0	1	1
1	0	1	1	1	1	0	1	0	1	0	1	1	0	1	0	1	1	1	0	1	0	1	0	1
1	1	1	0	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	1	1	1	0	1
0	1	0	1	1	0	1	0	1	0	1	0													

Tabela 4.43: Vetor A' corrigido referente ao gene Trav7

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

sozinho. Considere a palavra-código w_1 mostrada na Tabela 4.20, chamamos ela aqui de A, seja $q = 4$ e $n = 511$, criamos o vetor α mostrado na Tabela 4.44 que é dado pela Relação 4.1.

Com o vetor A e o seu correspondente vetor α, podemos então calcular os parâmetros β e γ, que são dados pela Relação 4.3:

$$\beta \equiv 2 \, mod \, 4 \tag{4.24}$$

e

$$\gamma \equiv 228 \, mod \, 511 \tag{4.25}$$

Tabela 4.44: Vetor α gerado através do vetor A da tabela 4.20 referente ao gene Hint-1

1	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	0	1	1
1	0	1	0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	0
1	1	0	1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	0	1	1	0	1	1	1
1	1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	1	0	1	0	1	0	1	0
1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	1	1	1
0	1	0	1	0	1	1	0	1	0	0	1	1	0	0	1	1	1	1	0	1	0	0	1	0
1	0	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	0
0	1	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1
0	1	0	1	1	1	0	0	1	1	1	1	0	1	1	0	1	0	0	1	0	1	1	1	0
1	1	0	1	1	0	1	0	0	1	1	0	0	1	0	1	0	0	1	0	1	0	1	0	1
1	0	0	1	1	1	1	0	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1
0	1	1	0	1	0	1	1	1	0	1	1	0	1	0	1	1	1	0	1	1	0	0	1	1
0	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1	1	0	0	1	1	0	1	1	1
1	1	1	0	1	1	1	1	0	0	1	1	1	1	0	1	0	1	0	1	1	1	0	1	0
1	0	1	1	0	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	0	0	0	1	1	1	0	0	1	0	1	0	1	1	0	0	1	0
1	0	1	0	1	1	0	1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1	0	0
1	1	0	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0
0	1	1	1	1	1	0	1	0	1	0	1	1	0	1	1	0	1	0	1	0	1	1	1	1
1	1	0	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	1	0	1	0
1	0	1	1	1	0	1	0	1	0	1														

Depois de encontrado o vetor A e o seu correspondente vetor α, simulamos uma deleção de nucleotídeo na posição 167 que contém o símbolo 2, sendo rotulada no caso 1 (A,C,G,T)=(0,1,2,3), assim o símbolo 2 corresponde ao nucleotídeo G. A sequência com a deleção da posição 167 é chamada de A', sendo mostrada na Tabela 4.45 juntamente com o seu correspondente vetor α' mostrado na Tabela 4.46 .

Após encontrar os parâmetros β e γ iremos calcular os parametros S_1, S_2 e W, necessários para reconstruir a sequência em que houve a deleção, a seguir mostramos os cálculos:

$$S_1 \equiv \beta - \sum_{i=1}^{n} a'_i (mod\, q) \equiv 2\, (mod\, 4) \tag{4.26}$$

e

$$S_2 \equiv \gamma - \sum_{i=1}^{n} (i-1)\alpha'_i (mod\, n) \equiv 375\, (mod\, 511) \tag{4.27}$$

W = 316

Como $S_2 \geq$ W, então na sequência α' inserimos o símbolo 1 de modo que o número de zeros no lado esquerdo do símbolo vai ser igual a S_2 - W, assim temos que: S_2 - W = 375 - 316 = 59. Fazendo os cálculos verificamos que precisamos ter 59 zeros do lado esquerdo de onde vamos inserir o símbolo 1, assim contando os zeros, percebemos que os 59 zeros são alcançados na

Tabela 4.45: vetor A' oriundo do vetor A referente ao gene Hint-1

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

posição 167 do vetor α', assim acrescentamos o símbolo 1 na posição 168, a sequencia corrigida α'_1 é mostrada na Tabela 4.47.

Como S_1=2, então concluímos que o símbolo que foi excluído é o 2, assim a única possibilidade de decodificação é colocar o símbolo 2 na posição 167 da sequência, concluímos que a sequência corrigida é igual a sequência enviada.

Uma outra abordagem foi testar o caso de inserção de nucleotídeo no gene Hint-1, como podemos visualizar nos casos anteriores o comportamento da fonte informação genética é semelhante a uma outra fonte de informação, podendo ser reconstruida uma mensagem em que houve uma deleção. Considere o vetor w_1 da Tabela 4.20, aqui chamado de A, seja $q = 4$ e $n = 511$, assim geramos o vetor α que é mostrado na Tabela 4.44, com estes resultados podemos calcular os parâmetros β e γ, dados pela Relação 4.3:

$$\beta \equiv 2 \, mod \, 4 \tag{4.28}$$

e

$$\gamma \equiv 228 \, mod \, 511 \tag{4.29}$$

Depois de calculado os parâmetros β e γ o próximo passo é criar um novo vetor A com uma inserção de nucleotídeo, aqui chamado de A'. Neste caso simulamos a inserção do número 3 na posição 499, assim geramos o A' com 512 elementos e seu correspondente vetor α' sendo

Tabela 4.46: vetor α' oriundo do vetor A' referente ao gene Hint-1

1	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	0	1	1
1	0	1	0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	0
1	1	0	1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	0	1	1	0	1	1	1
1	1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	1	0	1	0	1	0	1	0
1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	1	1	1
0	1	0	1	0	1	1	0	1	0	0	1	1	0	0	1	1	1	1	0	1	0	0	1	0
1	0	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	0	1	0	1	0	1	0	0
1	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1	0
1	0	1	1	1	0	0	1	1	1	1	0	1	1	0	1	0	0	1	0	1	1	1	0	1
1	0	1	1	0	1	0	0	1	1	0	0	1	0	1	0	0	1	0	1	0	1	0	1	1
0	0	1	1	1	1	0	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0
1	1	0	1	0	1	1	1	0	1	1	0	1	0	1	1	1	0	1	1	0	0	1	1	0
0	1	1	1	1	1	0	1	1	1	0	1	0	1	1	1	0	0	1	1	0	1	1	1	1
1	1	0	1	1	1	1	0	0	1	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1
0	1	1	0	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1	1	1	0
1	0	1	1	1	1	0	0	0	1	1	1	0	0	1	0	1	0	1	1	0	0	1	0	1
0	1	0	1	1	0	1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1	0	0	1
1	0	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0	0
1	1	1	1	1	0	1	0	1	0	1	1	0	1	1	0	1	0	1	0	1	1	1	1	1
1	0	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	1	0	1	0	1
0	1	1	1	0	1	0	1	0	1															

mostrado na Tabela 4.46:

Agora podemos calcular os parâmetros S_1, S_2 e W necessários para reconstruir a sequência enviada.

$$S_1 \equiv \sum_{i=1}^{n} a'_i - \beta (mod\, q) \equiv 3\, (mod\, 4) \tag{4.30}$$

e

$$S_2 \equiv \sum_{i=1}^{n} (i-1)\alpha'_i - \gamma (mod\, n) \equiv 7\, (mod\, 511) \tag{4.31}$$

W = 317

Se 0<S_2<W-1, então jogamos fora qualquer zero de modo que o número de uns a direita deste símbolo na sequência α' seja igual a S_2, assim precisamos ter 310 uns a esquerda, então jogamos fora o zero da posição 501, a nova sequência aqui chamada de α'_1 é mostrada é mostrada na Tabela 4.50.

Como S_1=3, então concluímos que o símbolo que foi incluindo é o 3, assim a única possibilidade de decodificação é excluir o símbolo 3 na posição 499 da sequência, assim concluímos que a sequência corrigida é igual a sequência enviada.

Como os start códon esta no éxon 1 e o stop códon esta no éxon 3 podemos ter as seguintes combinações entre éxons e íntrons: a primeira possibilidade de geração de proteínas é a compo-

Tabela 4.47: Vetor α'_1 oriundo do vetor α' referente ao gene Hint-1

1	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	0	1	1
1	0	1	0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	0
1	1	0	1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	0	1	1	0	1	1	1
1	1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	1	0	1	0	1	0	1	0
1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	1	1	1
0	1	0	1	0	1	1	0	1	0	0	1	1	0	0	1	1	1	1	0	1	0	0	1	0
1	0	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	0
0	1	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1
0	1	0	1	1	1	0	0	1	1	1	1	0	1	1	0	1	0	0	1	0	1	1	1	0
1	1	0	1	1	0	1	0	0	1	1	0	0	1	0	1	0	0	1	0	1	0	1	0	1
1	0	0	1	1	1	1	0	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1
0	1	1	0	1	0	1	1	1	0	1	1	0	1	0	1	1	1	0	1	1	0	0	1	1
0	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1	1	0	0	1	1	0	1	1	1
1	1	1	0	1	1	1	1	0	0	1	1	1	1	0	1	0	1	0	1	1	1	0	1	0
1	0	1	1	0	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	0	0	0	1	1	1	0	0	1	0	1	0	1	1	0	0	1	0
1	0	1	0	1	1	0	1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1	0	0
1	1	0	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0
0	1	1	1	1	1	0	1	0	1	0	1	1	0	1	1	0	1	0	1	0	1	1	1	1
1	1	0	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	1	0	1	0
1	0	1	1	1	0	1	0	1	0	1														

sição de todos os éxons e íntrons, a segunda possibilidade é a composição de éxon 1, íntron 1, éxon 2 e éxon 3, a terceira possibilidade é a composição de éxon 1, éxon 2, íntron 2, e éxon 3, a quarta possibilidade é a composição de éxon 1, éxon 2 e éxon 3 e a quinta possibilidade é a composição de éxon 1 com éxon 3, do ponto de vista biológico as demais combinações não são possiveis. Dado que a combinação do éxon 1 com o éxon 3 atende a restrição de comprimento do código BCH sobre anel $2^r - 1$, tendo 255 nucleotídeos, resolvemos verificar se este RNA maduro era palavra-código de um BCH e se era palavra-código de um Varshamov-Tenengolts. Nosso primeiro passo é mostrar que este RNA maduro é palavra-código de um BCH, através dos procedimentos delineados no Capítulo 3 o correspondente polinômio gerador dado por $g(x) = 1x^8 + 3x^5 + 1x^3 + 2x^2 + 3x^1 + 1$, rótulo caso 1 de acordo com a Tabela 3.3 bem como sua matriz geradora.

A Tabela 4.51 ilustra a palavra-código do RNA maduro referente ao gene Hint-1.

Após identificarmos o RNA maduro como palavra-código, vamos mostrar que este RNA maduro é identificado como uma palavra-código de um Varshamov-Tenengolts, assim faremos uma deleção de nucleotídeo. Considere a palavra-código na Tabela 4.51 aqui chamada de A e seja $q = 4$ e $n = 255$, vamos determinar o vetor α mostrado na Tabela 4.52 dado pela Relação 4.1:

Determinado o vetor A e seu correspondente vetor α podemos então determinar os parâme-

Tabela 4.48: Vetor A' oriundo do A após uma inserção referente ao gene Hint-1

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

tros β e γ necessários para reconstruir uma sequência em que houve uma deleção dados pela Relação 4.3.

$$\beta \equiv 2 \, mod \, 4 \tag{4.32}$$

e

$$\gamma \equiv 111 \, mod \, 255 \tag{4.33}$$

Para verificar se o RNA maduro é palavra-código de um código Varshamov-Tenengolts, vamos simular uma deleção de nucleotídeo na posição 123, sendo o símbolo deletado o número 2, assim geramos um novo A, aqui chamado de A' mostrado na Tabela 4.53 e seu correspondente α' mostrado na Tabela 4.54.

Gerado o A' e seu correspondente α', vamos calcular o S_1, S_2 e W o peso de α' (número de uns na sequência α').

$$S_1 \equiv \beta - \sum_{i=1}^{n} a'_i (mod \, q) \equiv 2 \, (mod \, 4) \tag{4.34}$$

e

$$S_2 \equiv \gamma - \sum_{i=1}^{n} (i-1) \alpha'_i (mod \, n) \equiv 201 \, (mod \, 255) \tag{4.35}$$

Tabela 4.49: Vetor α' oriundo do vetor A' referente ao gene Hint-1

1	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	0	1	1
1	0	1	0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	0
1	1	0	1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	0	1	1	0	1	1	1
1	1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	1	0	1	0	1	0	1	0
1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	1	1	1
0	1	0	1	0	1	1	0	1	0	0	1	1	0	0	1	1	1	1	0	1	0	0	1	0
1	0	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	0
0	1	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1
0	1	0	1	1	1	0	0	1	1	1	1	0	1	1	0	1	0	0	1	0	1	1	1	0
1	1	0	1	1	0	1	0	0	1	1	0	0	1	0	1	0	0	1	0	1	0	1	0	1
1	0	0	1	1	1	1	0	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1
0	1	1	0	1	0	1	1	1	0	1	1	0	1	0	1	1	1	0	1	1	0	0	1	1
0	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1	1	0	0	1	1	0	1	1	1
1	1	1	0	1	1	1	1	0	0	1	1	1	1	0	1	0	1	0	1	1	1	0	1	0
1	0	1	1	0	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	0	0	0	1	1	1	0	0	1	0	1	0	1	1	0	0	1	0
1	0	1	0	1	1	0	1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1	0	0
1	1	0	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0
0	1	1	1	1	1	0	1	0	1	0	1	1	0	1	1	0	1	0	1	0	1	1	1	1
1	1	0	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	1	0	1	0
0	1	0	1	1	1	0	1	0	1	0	1													

W = 158

Como $S_2 \geq$W, então na sequência α' mostrada na Tabela 4.54 inserimos o símbolo 1 de modo que o número de zeros do lado esquerdo do qual o símbolo vai inserido seja igual a S_2-W=201-158=43, então devemos ter 43 zeros do lado esquerdo de onde o símbolo será inserido. Fazendo a contagem verificamos que os 43 zeros são encontrados na posição 125, assim devemos colocar o 1 nesta posição e chegar na nova sequência α'_1 mostrada na Tabela 4.55.

Como S_1=2, então concluímos que o símbolo que foi excluído é o 2, assim a única possibilidade de decodificação é inserir o símbolo 2 na posição 123 da sequência, assim concluímos que a sequência corrigida é igual a sequência enviada. Portanto este mRNA maduro é uma palavra-código de um código BCH e também é palavra-código de um código de Varshamov-Tenengolts, assim podemos perceber que uma sequência de informação biológica pode ser identifica por mais de um código.

Tabela 4.50: Vetor α'_1 oriundo do vetor α' referente ao gene Hint-1

1	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	0	1	1
1	0	1	0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	0
1	1	0	1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	0	1	1	0	1	1	1
1	1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	1	0	1	0	1	0	1	0
1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	1	1	1
0	1	0	1	0	1	1	0	1	0	0	1	1	0	0	1	1	1	1	0	1	0	0	1	0
1	0	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	0
0	1	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1
0	1	0	1	1	1	0	0	1	1	1	1	0	1	1	0	1	0	0	1	0	1	1	1	0
1	1	0	1	1	0	1	0	0	1	1	0	0	1	0	1	0	0	1	0	1	0	1	0	1
1	0	0	1	1	1	1	0	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1
0	1	1	0	1	0	1	1	1	0	1	1	0	1	0	1	1	1	0	1	1	0	0	1	1
0	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1	1	0	0	1	1	0	1	1	1
1	1	1	0	1	1	1	1	0	0	1	1	1	1	0	1	0	1	0	1	1	1	0	1	0
1	0	1	1	0	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	0	0	0	1	1	1	0	0	1	0	1	0	1	1	0	0	1	0
1	0	1	0	1	1	0	1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1	0	0
1	1	0	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0
0	1	1	1	1	1	0	1	0	1	0	1	1	0	1	1	0	1	0	1	0	1	1	1	1
1	1	0	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	1	0	1	0
1	0	1	1	1	0	1	0	1	0	1														

Tabela 4.51: Palavra-código de um RNA maduro referente ao gene Hint-1

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

Tabela 4.52: Vetor α correspondente palavra-código do RNA maduro referente ao gene Hint-1

1	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	0	1	1
1	0	1	0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	0
1	1	0	1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	0	1	1	0	1	1	1
1	1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	1	0	1	0	1	0	1	0
1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	1	1	1
1	0	0	0	1	1	1	0	0	1	0	1	0	1	1	0	0	1	0	1	1	0	0	1	1
0	1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1	0	0	1	1	0	1	1	0
1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0	0	1	1	1	1	1
0	1	0	1	0	1	1	0	0	1	0	1	0	1	0	1	1	1	1	1	1	0	1	1	0
1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	1	1	0
1	0	1	0	1																			

Tabela 4.53: Vetor A' oriuindo do vetor A referente ao gene Hint-1

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

Tabela 4.54: Vetor α' corresponde do vetor A' referente ao gene Hint-1

1	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	0	1	1
1	0	1	0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	0
1	1	0	1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	0	1	1	0	1	1	1
1	1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	1	0	1	0	1	0	1	0
1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	1	1	1
0	0	0	1	1	1	0	0	1	0	1	0	1	1	0	0	1	0	1	1	0	0	1	1	0
1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1	0	0	1	1	0	1	1	0	1
0	1	1	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0	0	1	1	1	1	1	0
1	0	1	0	1	1	0	0	1	0	1	0	1	0	1	1	1	1	1	1	0	1	1	0	1
1	0	1	1	1	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	1	1	0	1
0	1	0	1																				

Tabela 4.55: Vetor α'_1 referente ao gene Hint-1

1	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	0	1	1
1	0	1	0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	0
1	1	0	1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	0	1	1	0	1	1	1
1	1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	1	0	1	0	1	0	1	0
1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	1	1	1
1	0	0	0	1	1	1	0	0	1	0	1	0	1	1	0	0	1	0	1	1	0	0	1	1
0	1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1	0	0	1	1	0	1	1	0
1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0	0	1	1	1	1	1
0	1	0	1	0	1	1	0	0	1	0	1	0	1	0	1	1	1	1	1	1	0	1	1	0
1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	1	1	0

Capítulo 5

Conclusões e Sugestões de Trabalhos Futuros

Este trabalho de pesquisa teve como objetivo estabelecer propriedades matemáticas associadas ao splicing alternativo. Para isso, consideramos os genes TRAV7 e Hint-1 pela importância biológica associada. Identificamos estes genes com palavras-código de códigos cíclicos (BCH). O gene Trav7 do *genoma humano* (identificado por "geneID"número **28686**, no NCBI) é composto por 2 éxons e 1 íntron com os seguintes comprimentos (em termos de nucleotídeos): éxon 1 com 52, íntron 1 com 174, éxon 2 com 285, totalizando 511 nucleotídeos. Este gene foi identificado com uma palavra-código de um código BCH com parâmetros $(511, 502, 3)$ sobre o anel $\mathbb{Z}_4$ com polinômio gerador $g(x) = x^9 + 3x^8 + 2x^7 + 2x^6 + x^5 + x^4 + 2x^2 + 3$, rotulado no caso 3 $\{A, C, G, T\} = \{0, 2, 1, 3\}$, via o algoritmo de geração de sequências de DNA proposto em [4]- [3]- [65] e [66]. A matriz geradora G tem dimensão 502×511.

O gene Hint-1 do *nematoide C. elegans* (identificado por "geneID"número **184760**, no NCBI) é composto por 3 éxons e 2 íntrons com os seguintes comprimentos (em termos de nucleotídeos): éxon 1 com 123, íntron 1 com 44, éxon 2 com 138, íntron 2 com 74 e éxon 3 com 132, totalizando 511 nucleotídeos. Este gene foi identificado com uma palavra-código de um código BCH com parâmetros $(511, 502, 3)$ sobre o anel $\mathbb{Z}_4$ com polinômio gerador $g(x) = x^9+2x^7+x^5+3$ rotulada no caso 1 $\{A, C, G, T\} = \{0, 1, 2, 3\}$, via o algoritmo de geração de sequências de DNA proposto em [4]- [3]- [65] e [66]. A matriz geradora G tem dimensão 502×511. Por outro lado, sabemos do processo de codificação que a palavra-código v (sequência do gene Trav7) resulta da seguinte operação $v = u.G$.

Após a identificação dos genes Trav7 e Hint-1 como palavra-código determinamos quem era o vetor u que multiplicado pela matriz geradora G resulta na palavra-código v. As análises matemáticas tiveram como ponto de partida a identificação da localização dos éxons e íntrons no vetor u, na matriz geradora G e na palavra-código v, após encontrar a localização de cada éxon e íntron, verificamos se era possível gerar cada um dos éxons e cada um dos íntrons, visto que no splicing alternativo, cada éxon e íntron é separado por um processo conhecido como clivagem e depois justapostos de diferentes formas.

Sob o ponto de vista do vetor sinalização u, notamos que existem componentes deste vetor que são comuns tanto a éxons como a íntrons, mostrando uma forte ligação na região de fronteira. Uma interpretação biológica que fazemos do vetor sinalização u é a de realizar a localização/identificação no DNA da sequência precursora do RNA, pré-RNA. Para isso, é ne-

cessário que o mecanismo de splicing do pré-mRNA entre em ação. Isto por sua vez implica que a maquinaria de splicing deve reconhecer três regiões na molécula precursora do RNA: a região de splicing 5', a região de splicing 3' e o ponto da forquilha na sequência do íntron que forma a base do fragmento em laço a ser excisado. Cada um desses três sítios tem uma sequência nucleotídica consenso, que é similar entre os íntrons e que fornece a posição onde deve ocorrer o splicing.

Dado o gene Trav7, sob o ponto de vista da matriz geradora G, podemos notar que o espaço vetorial gerado tem dimensão 502. Todavia, as dimensões dos subespaços correspondentes aos éxon 1, íntron 1, éxon 2, apresentam os seguintes valores 52, 183, 285. Note que a soma dessas dimensões vale 520, portanto ultrapassando o valor 502. Isso implica que o espaço total não é uma soma direta dos correspondentes subespaços. Mais ainda, estabelece uma dependência entre os subespaços vizinhos. Essa dependência entre subespaços vizinhos nada mais é que uma memória associada.

Sob o ponto de vista da matriz geradora G do gene Hint-1 podemos notar que o espaço vetorial gerado tem dimensão 502. Todavia, as dimensões dos subespaços correspondentes aos éxon 1, íntron 1, éxon 2, íntron 2, e éxon 3 apresentam os seguintes valores 123, 53, 147, 83, 132. Note que a soma dessas dimensões vale 538, portanto ultrapassando o valor 502. Isso implica que o espaço total não é uma soma direta dos correspondentes subespaços. Mais ainda, estabelece uma dependência entre os subespaços vizinhos. Essa dependência entre subespaços vizinhos nada mais é que uma memória associada. Biologicamente podemos inferir que um íntron estabelece um processo de "amarramento"entre os éxons subsequentes e que se mostram importantes tanto no aspecto da realização do splicing alternativo como no da confiabilidade. Ambos processos de vital importância para a conservação da espécie.

Com este trabalho mostramos que é possível gerar cada éxon e íntron separadamente, assim podemos fazer a justaposição de éxons e íntrons de acordo com as restrições biológicas. Esta justaposição de éxons e íntrons é feita através de uma concatenação de vetores. No capítulo de resultados podemos observar que o gene TRAV7 tem duas possibilidades de splicing alternativo de acordo com as restrições biológicas e o gene Hint-1 tem cinco possibilidades de splicing alternativo de acordo com as restrições biológicas, portanto podemos expressar matematicamente o splicing alternativo no caso dos genes TRAV7 e Hint-1.

Nesta pesquisa também analisamos a geração de partes de um genoma, visto que este processamento ocorre nas células. Para tal, escolhemos o genoma de um plasmídeo contendo nove regiões. Fazendo uso da mesma análise realizada para éxons e íntrons mostramos que é possível gerar estas nove regiões do genoma.

Por fim, um outro objetivo era verificar se podemos corrigir erros de deleção e inserção em sequências genéticas, visto que estes tipos de erros ocorrem nos genes. Dentro desta classe de códigos escolhemos o código de Varshamov-Tenengolts para alfabetos não binários já que o alfabeto genético é composto por 4 bases. Podemos visualizar nos resultados que o código de Varshamov-Tenengolts é capaz de corrigir erros de deleção e inserção em sequências genéticas. Além disso, podemos observar que além da estrutura matemática dos códigos BCH estarem presentes no processamento da informação genética, a estrutura matemática dos códigos de Varshamov-Tenengolts se fazem presentes também neste processamento. Isso nos leva a crer

que existem indícios que outras estruturas matemáticas podem ser associadas ao processamento da informação genética.

Sugestões de Trabalhos Futuros

Diante dos resultados apresentados e do encaminhamento da pesquisa sugerimos os seguintes temas para trabalhos futuros:

1. Proposta de um procedimento para a determinação das regiões codantes em uma sequência de DNA via teoria de informação.

2. Proposta de códigos corretores de erros de múltiplas inserções e deleções.

3. Modelagem do canal de informação biológica.

4. Construção de códigos de fontes de informação biológica.

Trabalhos Publicados

Franco L. A. L., R. Palazzo Jr. Analysis of Mathematical Properties Associated with Alternative Splicing Through The Identification of the Correspondent Generating Matrix of na Error Correcting Code. Advanced Topics In Genomics And Cell Biology. Universidade Estadual de Campinas-Unicamp. Campinas-SP, Março de 2013.

Franco L. A. L., R. Palazzo Jr. Análise do Splicing Alternativo do Gene Hint-1 Através do Código BCH Associado. Congresso Nacional de Matemática Aplicada e Computacional (XXXV CNMAC). Natal-RN, Setembro de 2014.

Bibliografia

[1] B. Alberts, *Biologia Molecular da Célula*, 5th ed. Artes Medicas, 2010.

[2] Reddy ASN., "Alternative splicing of pre-messenger rnas in plants in the genomic era," *Annu. Rev. Plant Biol*, vol. 58, pp. 267–94, 2007.

[3] Faria L. C. B., R. Palazzo Jr., "Existências de códigos corretores de erros e protocolos de comunicação em sequências de dna," Ph.D. dissertation, FEEC-UNICAMP, Julho 2011.

[4] A.S.L. Rocha, R. Palazzo Jr. e M.C. Silva-Filho, "Modelo de sistema de comunicações digital para o mecanismo de importação de proteínas mitocondriais através de códigos corretores de erros," Ph.D. dissertation, DT-FEEC-UNICAMP, Fevereiro 2010.

[5] Marieb E. N., Hoehn K., *Anatomia e Fisiologia*. Artmed, 2009.

[6] Jurica MS, Moore MJ., "Pre-mrna splicing: Awash in a sea of proteins," *Molecular Cell*, vol. 12, pp. 5–14, 2003.

[7] Nilsen, T. W., "The spliceosome: The most complex macromolecular machine in the cell?" *Bioessays*, vol. 25, pp. 1147–1149, 2003.

[8] Zhou Z., and Licklider L.J., and Gygi S.P., and Reed R., "Comprehensive proteomic analysis of the human spliceosome," *Nature*, vol. 419, pp. 182–185, 2002.

[9] J. J.M., C. J., G.-E. P., K. Z., L. P.M., A. C.D., S. R., S. E.E., S. R., and S. D.D., "Genome-wide survey of human alternative pre-mRNA splicing with exon junction microarrays," *Science*, vol. 302, pp. 2141–2144, 2003.

[10] G. Ast, "How did alternative splicing evolve?" *Nature Review Genetics*, vol. 5(10), pp. 773–782, 2004.

[11] B. S., L. I., and B. P., " Alternative splicing and evolution," *Bioessays*, vol. 25(11), pp. 1031–1034, 2003.

[12] L. L.F., G. R.E., B. R.S., and B. S.E., "The evolving roles of alternative splicing," *Current Opinion in Structural Biology*, vol. 14(3), pp. 273–282, 2004.

[13] W. E.T., S. R., L. S., K. I., Z. L., M. C., K. S. F., S. G.P., and B. C.B., "Alternative isoform regulation in human tissue transcriptomes," *Nature*, vol. 456(7221), pp. 470–476, 2008.

[14] R. ASN., " Nuclear pre-mRNA splicing in plants," *Crit. Rev. Plant Sci.*, vol. 20, pp. 523–71, 2001.

[15] B. V. Wang BB, " Genomewide comparative analysis of alternative splic- spliced. ing in plants," *Proc. Natl. Acad. Sci. USA*, vol. 103, pp. 7175–80, 2006.

[16] S. C. Brown JW, " Splice site selection in plant pre-mRNA splicing," *Annu. Rev. Plant Physiol. Plant Mol. Biol.*, vol. 49, pp. 77–95, 1998.

[17] F. W. Simpson GG, "Splicing of precursors to mRNA in higher plants:mechanism, regulation, and subnuclear organization of the spliceosomal machinery." *Plant Mol. Biol.*, vol. 32, pp. 1–41, 1996.

[18] A. NN, T. ME, B. VV, T. T, F. RB, and F. KA., "Features of Arabidopsis genes and genome discovered using full-length cDNAs," *Plant Mol. Biol.*, vol. 60, pp. 69–85, 2006.

[19] Z. W., S. SD, and B. V., "Refined annotation of the Arabidopsis genome by complete expressed sequence tag mapping," *Plant Physiol*, vol. 132, pp. 469–84, 2003.

[20] W. Z., Burge, and C. B., "Splicing regulation: From a parts list of regulatory elements to an integrated splicing code," *A Publication Of The RNA Society.*, vol. 14, pp. 802–813, 2012.

[21] L. ZJ, L. R, F. C, and B. A., "Evolutionary conservation of minor u12-type spliceosome between plants and humans."

[22] B. V. Wang BB, "The ASRG database: identification and survey of Arabidopsis thaliana genes involved in premRNA splicing," *Genome Biol.*, vol. 5, p. R102, 2004.

[23] R. Hamming, "Error Detecting and Error Correcting Codes," *The Bell System Technical Journal*, vol. 29, pp. 379–423, 623–656, 1948.

[24] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, 1948.

[25] M. Golay, "Notes on digital coding," *Proc. IEEE*, vol. 37, p. 657, 1949.

[26] F. F. Sellers, "Bit loss and gain correction codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 35–38, 1962.

[27] H. M. [et al.], "A Survey of Error-Correcting Codes for Channels With Symbol Synchronization Errors," *IEEE Communications Surveys & Tutorials*, vol. 12, pp. 87–96, 2010.

[28] K. A. S. Immink, *Codes for Mass Data Storage Systems*, 2nd ed. Shannon Foundation Publishers, 2004.

[29] D. Kinniment, *Synchronization and Arbitration in Digital Systems*. John Wiley & Sons, 2008.

[30] B. Sklar, *Digital Communications: Fundamentals and Applications*, 2nd ed. Prentice Hall Communications Engineering and Emerging Technologies Series, 2001.

[31] S. W. Golomb, J. R. Davey, I. S. Reed, H. L. V. Trees, and J. J. Stiffler, "Synchronization," *IEEE Trans. Commun. Syst.*, vol. 11, no. 4, pp. 481–491, 1963.

[32] D. Sankoff and E. J. Kruskal, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. CSLI Publications, 1999.

[33] A. Orlitsky, "Interactive communication of balanced distributions and of correlated files," *SIAM J. Discrete Mathematics*, vol. 6, no. 4, pp. 548–564, 1993.

[34] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Cambridge University Press, 1998.

[35] G. Cormode, M. Paterson, S. Sahinalp, and U. Vishkin, "Communication complexity of document exchange," *in Proc. eleventh ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 197–206, 2000.

[36] S. Agarwal, D. Starobinski, and A. Trachtenberg, "On the scalability of data synchronization protocols for PDAs and mobile devices," *IEEE Netw*, pp. 22–28, 2002.

[37] Shu Lin and D.J. Costello Jr, *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, Inc., Englewood Clis, NJ, 1983.

[38] W.W.Peterson and E. Jr., *Error-Correcting Codes*, 2nd ed. MIT Press, 1972.

[39] F. McWillians and N. Sloane, *The Theory of Error Correcting Codes*. North-Holland Publishing Company, 1977.

[40] R. R. Varshamov and G. M. Tenengolts, "Codes which correct single asymmetric errors," *Automation and Remote Control*, vol. 26, no. 2, pp. 286–290, 1965.

[41] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics-Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[42] J. D. Ullman, "Near-optimal, single-synchronization-error-correcting code," *IEEE Trans. Inf. Theory*, vol. 12, no. 4, pp. 418–424, 1966.

[43] G. M. Tenengolts, "Class of codes correcting bit loss and errors in the preceding bit," *Automation and Remote Control*, vol. 37, pp. 797–802, 1976.

[44] A. S. J. Helberg and H. C. Ferreira, "On multiple insertion/deletion correcting codes," *IEEE Trans. Inf. Theory*, vol. 48, no. 1, pp. 305–308, 2002.

[45] L. Calabi and W. E. Hartnett, "Some general results of coding theory with applications to the study of codes for the correction of synchronization errors," *Information and Control*, vol. 15, no. 3, pp. 235–249, 1969.

[46] E. Tanaka and T. Kasai, "Synchronization and substitution errorcorrecting codes for the Levenshtein metric," *IEEE Trans. Inf. Theory*, vol. 22, no. 2, pp. 156–162, 1976.

[47] G. Tenengolts, "Nonbinary codes, correcting single deletion or insertion," *IEEE Trans. Inf. Theory*, vol. 30, no. 5, pp. 766–769, 1984.

[48] C. J. Colbourn and E. J. H. Dinitz, *Handbook of Combinatorial Designs*, 2nd ed. Chapman & Hall/CRC, 2006.

[49] D. Tonien and R. Safavi-Naini, "Construction of deletion correcting codes using generalized Reed-Solomon codes and their subcodes," *Designs, Codes and Cryptography*, vol. 42, pp. 227–237, 2007.

[50] L. McAven and R. Safavi-Naini, "Classification of the deletioncorrecting capabilities of Reed-Solomon codes of dimension 2 over prime fields," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2280–2294, 2007.

[51] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1757–1767, 2007.

[52] V. I. Levenshtein, "Asymptotically optimum binary codes with correction for losses of one or two adjacent bits," *Systems Theory Research*, vol. 19, pp. 298–304, 1970.

[53] I. Iizuka, M. Kasahara, and T. Namekawa, "Block codes capable of correcting both additive and timing errors," *IEEE Trans. Inf. Theory*, vol. 26, no. 4, pp. 393–400, 1980.

[54] K. Iwamura and H. Imai, "A code to correct synchronization errors," *Electronics and Communications in Japan, part 3*, vol. 76, no. 6, pp. 60–71, 1993.

[55] S. W. Golomb and L. R. Welch, "Comma-free codes," *Candian Journal of Mathematics*, vol. 10, pp. 202–209, 1958.

[56] N. H. Lam, "Completing Comma-free codes," *Theoretical Computer Science*, vol. 301, pp. 399–415, 2003.

[57] P. A. H. Bours, "Codes for correcting insertion and deletion errors," Ph.D. dissertation, Eindhoven University of Technology, June 1994.

[58] I. Herstein, *Topics in Algebra*. John Wiley and Sons, New York, 1975.

[59] J. Fraleigh, *A First Course in Asbtract Algebra*. Addison-Welwy Publishing Co, 1982.

[60] P.R. Barbosa, R. Palazzo Jr, "Construção de códigos $\mathbb{Z}_{2^k}$-pseudolineares através de aplicações isométricas e extensões de galois sobre anéis locais," Ph.D. dissertation, DT-FEEC-UNICAMP, Junho 2000.

[61] J.C. Interlando, R. Palazzo Jr., "Uma contribuição à construção e decodificação de códigos lineares sobre grupos abelianos via concatenação de códigos sobre anéis de inteiros residuais," Ph.D. dissertation, DT-FEEC-UNICAMP, Dezembro 1994.

[62] B. McDonald, *Finite Rings with Identity*. Marcel Dekker, New York, 1974.

[63] J. Interlando, R. P. Jr., J. Gerônimo, A. Andrade, O. Favareto, and T. da Nóbrega Neto, "Códigos Corretores de Erros sobre Estruturas de Corpos, Anéis e Grupos," *DT-FEEC-UNICAMP*, 1998.

[64] P. Shankar, "On BCH codes over arbitrary integer rings," *IEEE Trans. Inform. Theory*, vol. 25, pp. 480–483, 1979.

[65] A. Rocha, L. Faria, J. Kleinschmidt, J. Palazzo, R., and M. Silva-Filho, "Dna sequences generated by $\mathbb{Z}_4$-linear codes," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, June 2010, pp. 1320–1324.

[66] Faria L. C. B., and Rocha A. S. L., and Kleinschmidt J. H., and Silva-Filho M. C., and Bim E., and Herai R. H., and Yamagishi M. E. B., and Palazzo Jr. R., "Is a Genome a Codeword of an Error-Correcting Code?" *Plos ONE*, vol. 7, no. 5, p. e36644, 2012.

[67] G. S. Lauer, "Some optimal partial unit-memory codes," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 240–243, 1979.

[68] L. N. Lee, "Short unit-memory, byte-oriented, binary convolutional codes having maximal free distance," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 349–352, 1976.