

Universidade Federal de Alfenas

**Abner Saimont Teixeira**

***Implementação do algoritmo soma  $\mathbb{Z}_{64}$  na  
análise de sequências de proteínas***

Alfenas/MG

2024

**Abner Saimont Teixeira**

***Implementação do algoritmo soma  $\mathbb{Z}_{64}$  na  
análise de seqüências de proteínas***

Trabalho de Conclusão de Curso apresentado como parte dos requisitos para obtenção do título de Licenciado em Matemática, pelo Instituto de Ciências Exatas, da Universidade Federal de Alfenas. Área de concentração: Matemática Aplicada. Orientação: Anderson José de Oliveira.

Alfenas/MG

2024

**Abner Saimont Teixeira**

***Implementação do algoritmo Soma Z64 na análise de sequências de proteínas***

A banca examinadora abaixo-assinada, aprova a Monografia apresentada como parte dos requisitos para obtenção do título de Licenciado em Matemática, pelo Instituto de Ciências Exatas, da Universidade Federal de Alfenas. Área de concentração: Matemática Aplicada.

Aprovado em: 26/06/2024

Banca examinadora:

 Documento assinado digitalmente  
**ANDERSON JOSE DE OLIVEIRA**  
Data: 10/07/2024 13:36:42-0300  
Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Anderson José de Oliveira  
Instituto de Ciências Exatas  
Orientador

 Documento assinado digitalmente  
**CÁTIA REGINA DE OLIVEIRA QUILLES QUEIROZ**  
Data: 10/07/2024 11:23:16-0300  
Verifique em <https://validar.iti.gov.br>

---

Profa. Dra. Cátia Regina de Oliveira Quilles Queiroz  
Instituto de Ciências Exatas  
Avaliador 1

 Documento assinado digitalmente  
**MARCELO MOREIRA DA SILVA**  
Data: 10/07/2024 11:35:10-0300  
Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Marcelo Moreira da Silva  
Instituto de Ciências Exatas  
Avaliador 2

---

Profa. Dra. Angela Leite Moreno  
Instituto de Ciências Exatas  
Suplente

# Agradecimentos

Agradeço primeiramente a Deus, por me conceder força, sabedoria e saúde.

Ao meu pai, minha mãe e minha irmã, pelo amor incondicional e pelo apoio em TODOS os momentos. Amo vocês <3.

Ao meu orientador, Anderson Oliveira, pela orientação, dedicação e, principalmente, pela paciência ao longo do desenvolvimento deste trabalho.

# Resumo

O processo relacionado à modelagem e conseqüente análise do código genético é algo que diversos pesquisadores vêm realizando, a fim de identificar características e possíveis implicações em diversos fenômenos, como no caso de estudos de mutações genéticas. Existem algoritmos utilizados para realizar a soma entre códons, como o algoritmo SMG, o soma com transportes e o soma  $\mathbb{Z}_{64}$ . Este trabalho apresenta a implementação e a análise do algoritmo soma  $\mathbb{Z}_{64}$  na modelagem matemática de seqüências de proteínas. O objetivo principal é o de investigar a aplicabilidade desse algoritmo na análise de comportamentos genéticos e na possível previsão de mutações. Utilizando a linguagem de programação *Python*, o algoritmo foi desenvolvido e testado com diversas seqüências genéticas fictícias, proporcionando uma validação de suas respostas. Os resultados demonstraram a eficácia do algoritmo, destacando seu potencial como ferramenta em estudos de Biologia Computacional.

**Palavras-chave:** Código genético; Somas entre códons; Modelagem matemática; *Python*.

# Abstract

The process related to the modeling and subsequent analysis of the genetic code is something that several researchers have been carrying out, in order to identify characteristics and possible implications in different characteristics, as in the case of studies of genetic mutations. There are algorithms used to perform the sum among codons, such as the SMG algorithm, the sum with transports and the sum  $\mathbb{Z}_{64}$ . This work presents the implementation and analysis of the sum  $\mathbb{Z}_{64}$  algorithm in the mathematical modeling of protein sequences. The main objective is to investigate the applicability of this algorithm in the analysis of genetic behaviors and the possible prediction of mutations. Using the *Python* programming language, the algorithm was developed and tested with several fictitious genetic sequences, providing validation of its answers. The results demonstrated the effectiveness of the algorithm, highlighting its potential as a tool in Computational Biology studies.

**Keywords:** Genetic code; Sums among codons; Mathematical modeling; *Python*.

# Lista de ilustrações

Figura 1 – Exemplo de célula procarionte. . . . .	11
Figura 2 – Exemplo de célula eucarionte. . . . .	11
Figura 3 – Estrutura dos ácidos nucleicos. . . . .	12
Figura 4 – Estrutura do DNA. . . . .	13
Figura 5 – Replicação do DNA. . . . .	14
Figura 6 – Estrutura dos aminoácidos. . . . .	15
Figura 7 – Código genético. . . . .	16
Figura 8 – Rotulamentos do código genético. . . . .	23
Figura 9 – Fluxograma do código. . . . .	32

# Sumário

	<b>Lista de ilustrações</b> . . . . .	<b>6</b>
<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>8</b>
<b>2</b>	<b>REVISÃO DE CONCEITOS</b> . . . . .	<b>10</b>
<b>2.1</b>	<b>Elementos de Biologia</b> . . . . .	<b>10</b>
2.1.1	A célula . . . . .	10
2.1.2	Nucleotídeos e ácidos nucleicos . . . . .	11
2.1.3	Estrutura do DNA . . . . .	12
2.1.4	Síntese proteica . . . . .	13
2.1.5	Aminoácidos . . . . .	14
2.1.6	Código genético . . . . .	15
2.1.7	Mutações genéticas . . . . .	16
<b>2.2</b>	<b>Elementos de Álgebra</b> . . . . .	<b>17</b>
2.2.1	Operações binárias . . . . .	17
2.2.2	Grupos . . . . .	18
2.2.3	Anéis . . . . .	19
2.2.4	Corpos . . . . .	21
<b>2.3</b>	<b>Modelagem do Código Genético</b> . . . . .	<b>22</b>
2.3.1	Rotulamentos . . . . .	22
2.3.2	Algoritmo SMG . . . . .	24
2.3.3	Algoritmo soma com transporte . . . . .	25
<b>2.4</b>	<b>Algoritmo Soma <math>\mathbb{Z}_{64}</math></b> . . . . .	<b>26</b>
<b>3</b>	<b>IMPLEMENTAÇÃO E APLICAÇÕES DO ALGORITMO SOMA <math>\mathbb{Z}_{64}</math></b>	<b>30</b>
<b>3.1</b>	<b>Código Utilizado</b> . . . . .	<b>30</b>
<b>3.2</b>	<b>Aplicações do Algoritmo Soma <math>\mathbb{Z}_{64}</math> envolvendo os Rotulamentos do Código Genético</b> . . . . .	<b>33</b>
3.2.1	Testes com o método de soma códon a códon . . . . .	33
3.2.2	Testes com o método de soma com a sequência completa . . . . .	35
<b>4</b>	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	<b>39</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>40</b>

# 1 **INTRODUÇÃO**

A análise do código genético e sua modelagem matemática têm sido fundamentais para caracterizar suas propriedades, essenciais para avanços em áreas como Biotecnologia e Medicina. Este campo interdisciplinar, situado na interseção entre Biologia, Matemática e Computação, visa analisar e prever comportamentos genéticos associados a diversos processos biológicos, incluindo a síntese de proteínas e possíveis respostas a mutações genéticas.

Como mencionado em [1], o código genético é um sistema bioquímico essencial para a transmissão das informações contidas nas moléculas de DNA e RNA em sequências de proteínas. Neste contexto, [2] compara o código genético à teoria de comunicações, destacando a importância de transmitir informações com a mínima taxa de erros possível, para evitar mutações que podem resultar em consequências significativas para o organismo.

Diversas modelagens têm sido propostas para essa análise. Entre elas, destacam-se a construção de reticulados booleanos e diagramas de Hasse, a utilização das distâncias de Hamming, além da aplicação de algoritmos envolvendo operações algébricas no código genético, como o Algoritmo da Soma com Transporte, Algoritmo SMG e o Algoritmo Soma  $\mathbb{Z}_{64}$ . As modelagens mencionadas foram apresentadas em [3], [4] e [5] e esses métodos permitem explorar diferentes aspectos relacionados a problemas biológicos, em particular, no estudo do código genético e das mutações associadas [3].

Embora vários algoritmos tenham sido propostos com o intuito de caracterizar possíveis fenômenos associados a sequências de DNA, este trabalho concentra-se exclusivamente na implementação e análise do Algoritmo Soma  $\mathbb{Z}_{64}$ , também estudado por [3] e proposto inicialmente por [6].

A implementação deste algoritmo ilustra a aplicação prática de conceitos matemáticos em problemas reais, evidenciando sua aplicabilidade em diversas áreas, como Biologia e Computação e sua essencialidade no desenvolvimento de novas tecnologias.

Este estudo justifica-se pela necessidade de métodos mais eficientes e precisos no processo de modelagem do código genético, e essa implementação proporciona uma nova abordagem ao fornecer uma ferramenta que pode facilitar a detecção e a compreensão de padrões genéticos complexos.

Este trabalho adota uma abordagem quantitativa e são utilizadas técnicas de modelagem algébrica para implementar o algoritmo, utilizando a linguagem de programação *Python*. As etapas principais incluem a revisão da literatura, o desenvolvimento e implementação do algoritmo em *Python*, e testes com sequências genéticas fictícias. A análise

dos resultados visa validar o desempenho do algoritmo proposto.

Este trabalho está estruturado da seguinte maneira: no capítulo 2, serão explorados os conceitos essenciais que fundamentam este trabalho. Devido à natureza interdisciplinar do estudo, serão abordados conceitos de Biologia e Álgebra, além da modelagem do código genético, onde serão discutidos os métodos e modelos utilizados para mapear o código genético, incluindo detalhes sobre os rotulamentos e o algoritmos de soma entre códons. No capítulo 3, será descrita detalhadamente a implementação do algoritmo em *Python*, bem como exemplos de sua aplicação em dados genéticos. Por fim, no capítulo 4, será apresentada uma reflexão sobre os resultados obtidos, as contribuições do trabalho, além de sugestões para futuras pesquisas que possam expandir ou aprofundar os estudos iniciados aqui.

## 2 REVISÃO DE CONCEITOS

Neste capítulo serão apresentados os conceitos fundamentais utilizados para a construção deste trabalho, com elementos fundamentais de Biologia, Álgebra e sobre a modelagem do código genético, com o detalhamento dos rotulamentos e de três algoritmos envolvendo a soma entre códon (algoritmo SMG, algoritmo soma com transporte e algoritmo soma  $\mathbb{Z}_{64}$ ).

### 2.1 Elementos de Biologia

Nesta seção serão apresentados os principais elementos associados à Biologia utilizados neste trabalho. As principais referências utilizadas, onde informações adicionais poderão ser encontradas foram: [7], [8], [9], [10].

#### 2.1.1 A célula

As células são as unidades fundamentais da vida, presentes em todos os organismos vivos. Elas surgiram inicialmente como entidades de vida livre, evoluindo para formar comunidades multicelulares complexas. Isso permitiu o desenvolvimento de organismos multicelulares, nos quais as células desempenham funções especializadas, como a obtenção de nutrientes, a produção de energia, a reprodução e produção de respostas a estímulos ambientais. Além disso elas se distribuem em tecidos e órgãos, permitindo a organização eficiente dos sistemas biológicos e a manutenção da vida.

Existem duas principais classificações de células: procariontes e eucariontes. As células procariontes, como as bactérias, são mais simples e não possuem um núcleo definido; seu material genético está disperso no citoplasma.

Já as células eucariontes, encontradas em plantas, animais e outros organismos complexos, possuem um núcleo bem definido, onde o material genético está separado do citoplasma por uma membrana nuclear. Além disso, as células eucariontes possuem diversas organelas membranosas, como as mitocôndrias e o retículo endoplasmático, que desempenham funções específicas.

Um exemplo de célula procarionte é apresentado na Figura 1 e um exemplo de célula eucarionte é apresentado na Figura 2.

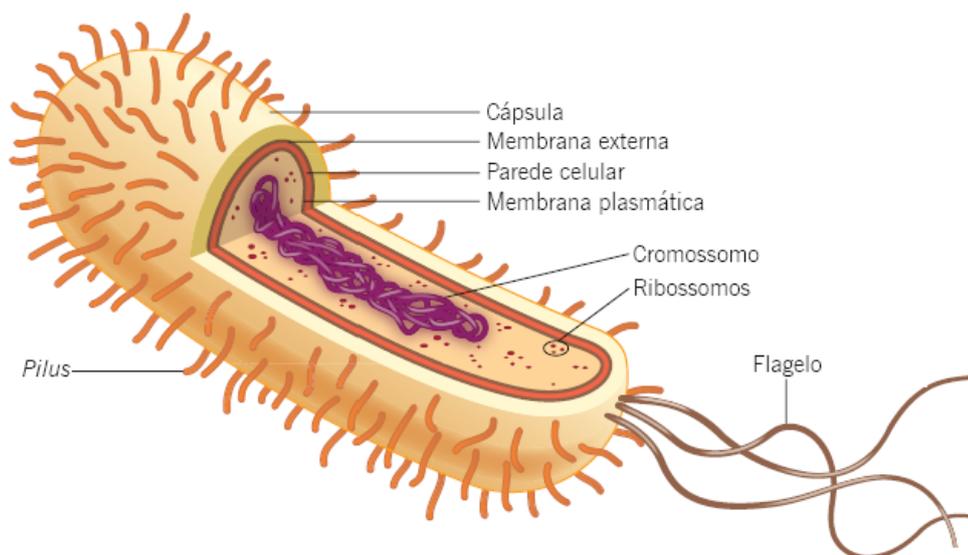


Figura 1 – Exemplo de célula procariote.

Fonte: [9].

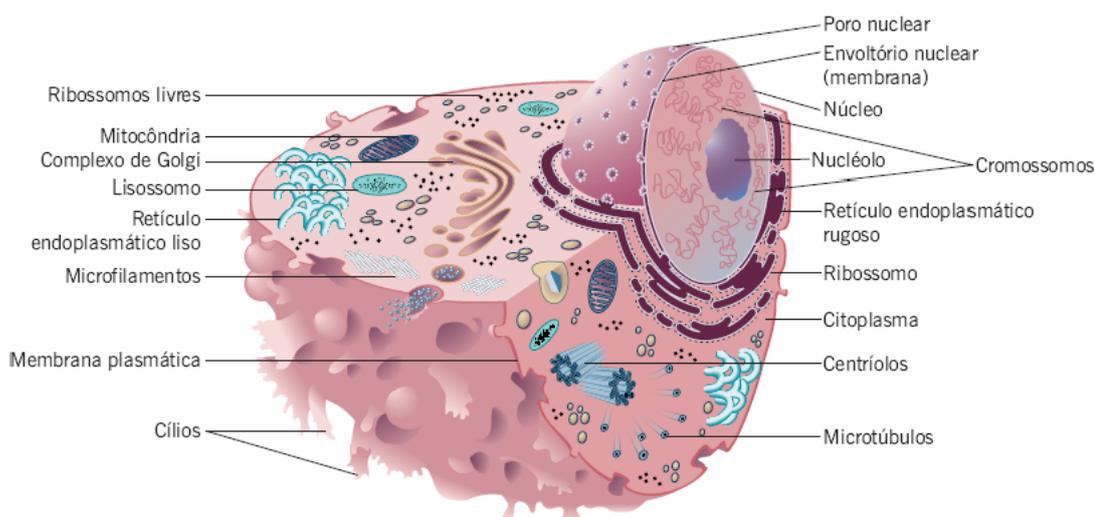


Figura 2 – Exemplo de célula eucariote.

Fonte: [9].

### 2.1.2 Nucleotídeos e ácidos nucleicos

Os ácidos nucleicos, DNA e RNA, são compostos por unidades chamadas nucleotídeos, sendo responsáveis por armazenar e codificar a informação genética. Cada nucleotídeo é constituído por três componentes principais: um grupo fosfato, uma pentose (açúcar de cinco carbonos) e uma base nitrogenada. No DNA, que geralmente tem uma estrutura de dupla hélice, a pentose é a desoxirribose, enquanto no RNA, que é tipicamente uma única fita, é a ribose. As bases nitrogenadas podem ser purinas, que possuem estrutura de anel duplo e incluem adenina (A) e guanina (G), ou pirimidinas, que possuem estrutura de anel simples e incluem citosina (C), timina (T) no DNA e uracila (U) no RNA. A estrutura dos nucleotídeos de ambos os ácidos é representada na Figura 3.

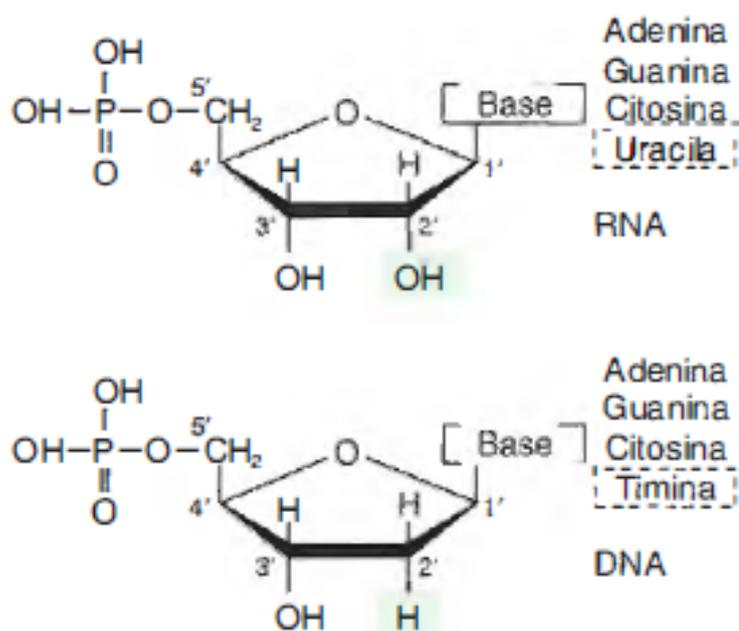


Figura 3 – Estrutura dos ácidos nucleicos.

Fonte: [10].

No DNA, a adenina forma duas pontes de hidrogênio com a timina, enquanto a guanina forma três pontes de hidrogênio com a citosina. Este emparelhamento específico é conhecido como a **regra de Chargaff** e é crucial para a estabilidade da estrutura de dupla hélice do DNA. No RNA, a adenina emparelha-se com a uracila em vez da timina, mas a guanina ainda emparelha-se com a citosina, seguindo a mesma lógica de complementaridade de bases.

Existem diferentes tipos de RNA, incluindo RNA mensageiro (RNAm), RNA transportador (RNAt) e RNA ribossômico (RNAr). Cada um desses tipos desempenha papéis distintos na célula, especialmente na síntese de proteínas.

### 2.1.3 Estrutura do DNA

A estrutura do DNA é composta por duas fitas de nucleotídeos que se enrolam em torno de um eixo comum, formando uma dupla hélice. Essas fitas são antiparalelas, significando que uma corre no sentido 5' para 3' e a outra no sentido 3' para 5'.

As ligações fosfodiéster em uma fita conectam o carbono 3' de um nucleotídeo ao carbono 5' do nucleotídeo adjacente, enquanto na fita complementar, as ligações seguem de um carbono 5' para um carbono 3'. A estabilização da dupla hélice é mantida pelas ligações de hidrogênio entre as bases nitrogenadas complementares e pelas interações hidrofóbicas entre as bases empilhadas no interior da hélice. Essa direção oposta dos filamentos complementares é essencial para os processos de replicação, transcrição e recombinação do DNA.

A Figura 4 apresenta a estrutura básica do DNA.

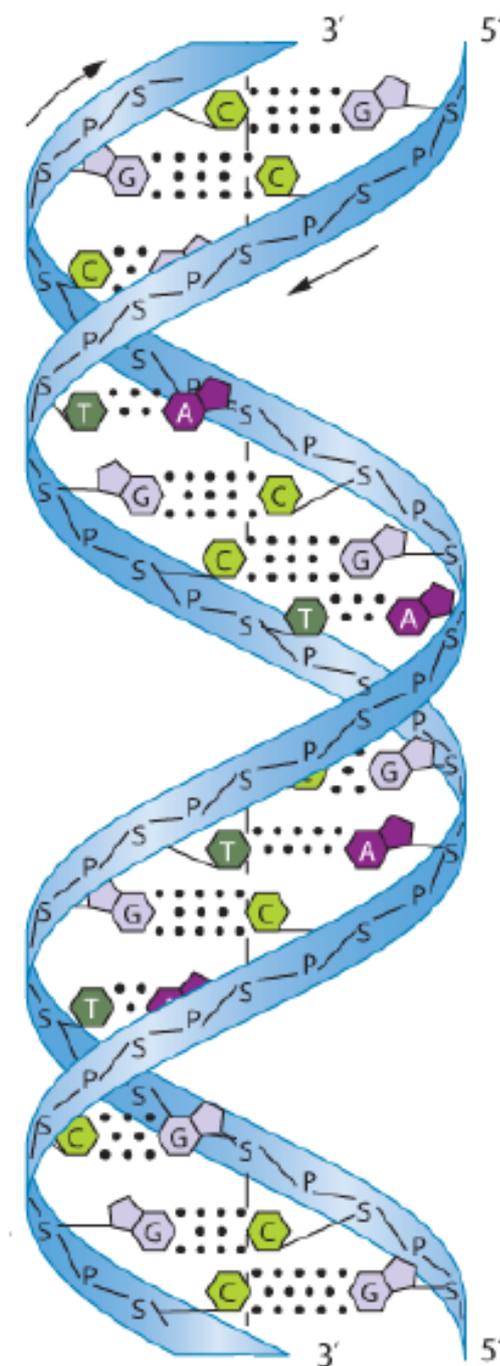


Figura 4 – Estrutura do DNA.  
Fonte: [9].

#### 2.1.4 Síntese proteica

A replicação do DNA é um processo que se inicia através da enzima helicase, que quebra as pontes de hidrogênio entre as bases nitrogenadas, separando as duas fitas complementares de DNA. Em seguida, a DNA polimerase adiciona nucleotídeos

complementares a cada fita molde. Esse processo é denominado semi-conservativo porque cada nova molécula de DNA é composta por uma fita original e uma fita recém-sintetizada. Como resultado, duas moléculas de DNA idênticas são formadas, cada uma contendo uma fita antiga e uma nova, conforme exemplificado na Figura 5.

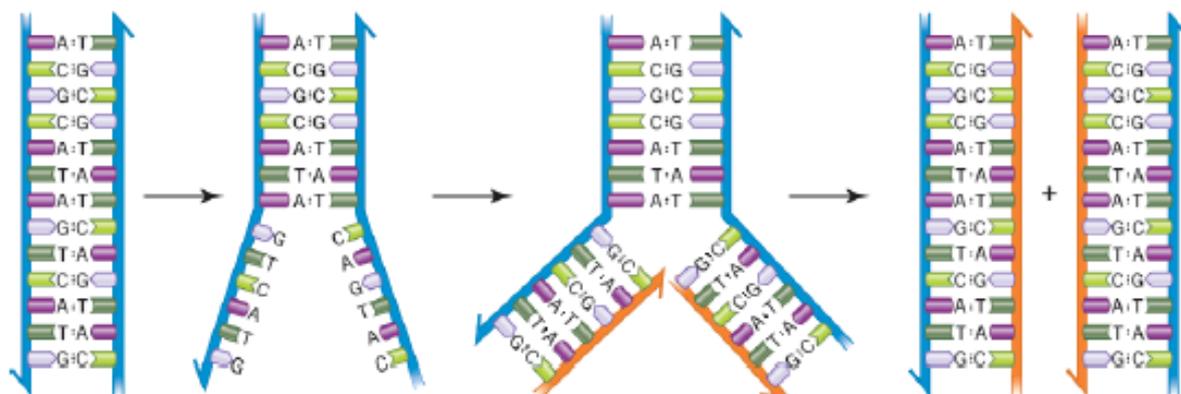


Figura 5 – Replicação do DNA.  
Fonte: [9].

A síntese de proteínas envolve duas etapas principais: transcrição e tradução. Na transcrição, uma região do DNA é usada como molde para produzir uma molécula de RNA mensageiro (RNAm). A enzima RNA polimerase liga nucleotídeos de RNA de acordo com a sequência do DNA, formando uma molécula de RNAm, que então sai do núcleo e vai para o citoplasma, onde os ribossomos a utilizam como guia para sintetizar as proteínas.

Durante a tradução, os ribossomos lêem a sequência de nucleotídeos do RNAm em grupos de três bases nitrogenadas, chamados códon. Cada códon especifica um aminoácido. O RNA transportador (RNAt) traz os aminoácidos corretos para o ribossomo, onde eles são ligados em uma cadeia crescente, formando uma proteína. Este processo garante que a informação genética do DNA seja corretamente expressa em proteínas, que desempenham inúmeras funções nas células.

### 2.1.5 Aminoácidos

Os aminoácidos são as unidades monoméricas que constituem as proteínas, cada um composto por um grupo amino ( $-\text{NH}_2$ ), um grupo carboxila ( $-\text{COOH}$ ), um átomo de hidrogênio e uma cadeia lateral específica (grupo R) ligada a um carbono central (carbono alfa). A variedade nas cadeias laterais é o que diferencia os 20 tipos de aminoácidos encontrados nas proteínas.

Os aminoácidos podem ser classificados de várias maneiras, incluindo sua estrutura, polaridade e essencialidade. Estruturalmente, os aminoácidos são divididos em dois grupos principais: alifáticos (cadeia lateral aberta) e aromáticos (contêm um anel benzênico). Quanto à polaridade, os aminoácidos podem ser hidrofílicos ou hidrofóbicos. Em termos

de essencialidade, eles são divididos em dois grupos: os essenciais, que não podem ser sintetizados pelo organismo e devem ser obtidos através da dieta, e não essenciais, que podem ser sintetizados pelo organismo humano.

A Figura 6 apresenta a estrutura básica dos aminoácidos.

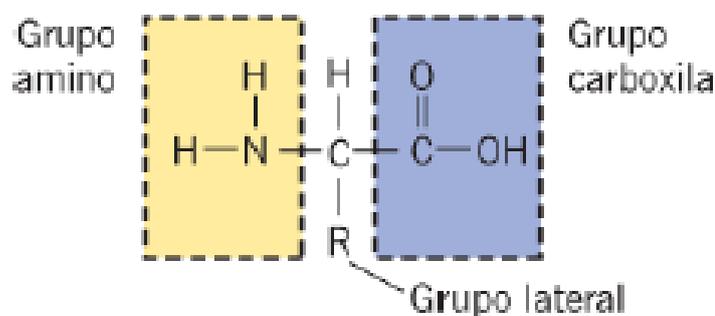


Figura 6 – Estrutura dos aminoácidos.

Fonte: [9].

Na síntese proteica, a importância dos aminoácidos não pode ser subestimada, pois a sequência e a composição dos aminoácidos determinam a estrutura final da proteína, o que, por sua vez, determina sua função biológica. Proteínas corretamente dobradas são essenciais para funções enzimáticas, sinalização celular, estrutura celular, dentre outras funções. Erros na sequência de aminoácidos podem levar a proteínas disfuncionais, que podem resultar em doenças ou disfunções celulares.

### 2.1.6 Código genético

O código genético é o conjunto de regras que determina como a sequência de nucleotídeos é traduzida em uma sequência de aminoácidos durante a síntese proteica. Cada grupo de três nucleotídeos no RNAm, conhecido como códon, especifica um aminoácido particular ou um sinal de término da tradução.

Existem 64 combinações possíveis de códon, resultantes das quatro bases nitrogenadas (adenina, uracila, citosina e guanina) que se combinam em grupos de três ( $4^3 = 64$ ).

Dos 64 códon, 61 codificam os 20 aminoácidos diferentes utilizados na síntese de proteínas, enquanto os três códon restantes (UAA, UAG e UGA) servem como sinais de parada, representados por STOP, indicando o fim da tradução. Essa redundância no código genético, onde múltiplos códon podem especificar o mesmo aminoácido, é uma característica que ajuda a proteger contra mutações.

A Figura 7 apresenta a representação do código genético.

		Segunda letra				
		U	C	A	G	
Primeira letra (5')	U	UUU } Phe (F)	UCU }	UAU } Tyr (Y)	UGU } Cys (C)	U
		UUC }	UCC } Ser (S)	UAC }	UGC }	C
		UUA } Leu (L)	UCA }	UAA Parada (terminador)	UGA Parada (terminador)	A
		UUG }	UCG }	UAG Parada (terminador)	UGG Trp (W)	G
	C	CUU }	CCU }	CAU } His (H)	CGU }	U
		CUC } Leu (L)	CCC } Pro (P)	CAC }	CGC }	C
		CUA }	CCA }	CAA } Gln (Q)	CGA }	A
		CUG }	CCG }	CAG }	CGG }	G
	A	AUU }	ACU }	AAU } Asn (N)	AGU } Ser (S)	U
		AUC } Ile (I)	ACC } Thr (T)	AAC }	AGC }	C
		AUA }	ACA }	AAA } Lys (K)	AGA }	A
		AUG Met (M) (iniciador)	ACG }	AAG }	AGG }	G
	G	GUU }	GCU }	GAU } Asp (D)	GGU }	U
		GUC } Val (V)	GCC } Ala (A)	GAC }	GGC }	C
		GUA }	GCA }	GAA } Glu (E)	GGA }	A
		GUG }	GCG }	GAG }	GGG }	G
						Terceira letra (3')

Figura 7 – Código genético.  
Fonte: [9].

### 2.1.7 Mutações genéticas

As mutações genéticas são alterações na sequência de nucleotídeos do DNA, que podem ocorrer de maneira espontânea ou ser induzidas por fatores externos como radiação e substâncias químicas. Essas mudanças podem ter variados efeitos sobre a célula e o organismo como um todo.

As mutações podem ser categorizadas em escalas: pequena escala, pontuais e de grande escala. Mutações de pequena escala afetam apenas um ou poucos nucleotídeos. Mutações pontuais são um tipo de mutação de pequena escala que envolvem a substituição de uma única base. Mutações de grande escala envolvem grandes segmentos de DNA e podem resultar em mudanças significativas na estrutura e função dos cromossomos, como deleções, duplicações, inversões e translocações.

Além disso, as mutações ainda podem ser classificadas de acordo com sua natureza e seus efeitos:

1. Substituições de base: uma base nitrogenada é substituída por outra. Estas podem ser:
  - Transições: substituição de uma purina por outra purina ( $A \leftrightarrow G$ ) ou de uma pirimidina por outra pirimidina ( $C \leftrightarrow T$ ).
  - Transversões: substituição de uma purina por uma pirimidina ou vice-versa ( $A \leftrightarrow C, A \leftrightarrow T, G \leftrightarrow C, G \leftrightarrow T$ ).
2. Inserções e deleções: adição ou remoção de nucleotídeos no DNA, o que pode causar uma alteração no momento da leitura, resultando em uma sequência completamente diferente de aminoácidos na proteína produzida.

O impacto das mutações genéticas em um organismo pode ser variado. Algumas mutações são neutras e não afetam significativamente a função biológica. Outras podem ser benéficas, conferindo uma vantagem evolutiva, enquanto algumas podem ser prejudiciais ou até mesmo letais. Mutações prejudiciais podem levar a doenças genéticas, afetar o desenvolvimento e a função celular, ou aumentar a chance de certos tipos de câncer.

## 2.2 Elementos de Álgebra

Nesta seção serão apresentados os principais conceitos, propriedades e informações acerca dos elementos fundamentais de Álgebra utilizados neste trabalho. As principais referências utilizadas, onde informações adicionais poderão ser consultadas foram: [11], [12] e [13].

### 2.2.1 Operações binárias

**Definição 1.** *Uma operação binária em um conjunto  $S$  é uma função  $*$  :  $S \times S \rightarrow S$ .*

Normalmente, uma operação binária é denotada por  $*(x, y) = x * y$ . Em outras palavras, uma operação binária sobre  $S$  deve associar a cada par ordenado  $(a, b)$  um elemento  $a * b$  que está também em  $S$ . Este requerimento de que o elemento associado deva estar em  $S$  é também conhecido como a condição de fechamento, isto é, exige-se que  $S$  seja fechado sob a operação binária.

Uma operação binária sobre um conjunto  $S$  é dita:

- comutativa, se  $a * b = b * a$ , para todo  $a, b \in S$ ;
- associativa, se  $(a * b) * c = a * (b * c)$ , para quaisquer que sejam  $a, b, c \in S$ ;

- ter elemento neutro  $e$ , se existe um elemento  $e$  em  $S$  tal que  $e * g = g = g * e$ , para todo  $g \in S$ ;
- simetrizável, se para cada  $g$  em  $S$ , existe um elemento  $g'$  em  $S$  tal que  $g' * g = g * g' = e$  (o elemento  $g'$  é um inverso de  $g$  com respeito à operação  $*$ ).

### 2.2.2 Grupos

**Definição 2.** Um grupo  $(G, *)$  é um conjunto não vazio  $G$ , junto com uma operação  $*$  sobre  $G$ , que satisfaz as seguintes propriedades:

- é associativa;
- tem elemento neutro  $e$ ;
- todo elemento é simetrizável.

Além disso, se a operação  $*$  for comutativa, isto é, se  $a * b = b * a$  para todos os elementos  $a$  e  $b \in G$ , então  $G$  é classificado como um grupo abeliano ou comutativo.

**Teorema 1.** O elemento neutro de um grupo  $G$  é único.

*Demonstração:* Sejam  $a \in G$  e  $e, e'$  elementos neutros de  $G$ . Assim,

$$e = e * e' = e'.$$

□

**Teorema 2.** O elemento inverso em um grupo  $G$  é único.

*Demonstração:* Sejam  $a \in G$  e  $a', a''$  elementos simétricos de  $a$ . Assim,

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

□

**Definição 3.** Se um subconjunto  $H$  de um grupo  $G$  é fechado sob a operação binária de  $G$  e se  $H$  forma um grupo, então diz-se que  $H$  é um subgrupo de  $G$  e denota-se isto por  $H \leq G$ .

**Definição 4.** Seja  $G$  um grupo finito. A ordem de  $G$  é o número de elementos de  $G$ . A cardinalidade de  $G$  é denotada por  $|G|$ .

A tábua de um grupo finito  $(G, *)$  é a tábua da operação considerada em  $G$ .

**Exemplo 1.**  $G = \{-1, 1\}$  é um grupo finito de ordem 2 em relação à multiplicação, que pode ser representada na seguinte tábua da Tabela 1 :

*	-1	1
-1	1	-1
1	-1	1

Tabela 1 – Tábua multiplicação.

Fonte: Adaptado de [11].

**Definição 5.** Dizemos que o inteiro  $r$  é a soma dos inteiros  $s$  e  $t$  módulo  $n$  se  $r$  for o resto da divisão de  $s + t$  por  $n$ . Como consequência, temos que  $\mathbb{Z}_n = \{\bar{0}, \bar{1}, \bar{2}, \dots, \overline{n-1}\}$ ,  $\forall n \geq 2$ . De forma análoga, define-se a operação produto módulo  $n$ ;  $\forall n \in \mathbb{N}$ ,  $n \geq 2$ .

**Teorema 3.** O conjunto  $\mathbb{Z}_n = \{\bar{0}, \bar{1}, \bar{2}, \dots, \overline{n-1}\}$  é um grupo sob a operação adição módulo  $n$ .

*Demonstração:* Sejam:  $\bar{0}$  : todos os inteiros cômugros a 0 mod  $n$ ,  $\bar{1}$  : todos os inteiros cômugros a 1 mod  $n$ ,  $\dots$ . Temos:

$$\bar{a} + \bar{b} = \overline{a + b} \text{ (vale a associatividade e a comutatividade);}$$

$$\bar{a} + \bar{0} = \overline{a + 0} = \bar{a} \text{ (}\bar{0} \text{ é o elemento neutro);}$$

$\overline{n-a}$  é o oposto de  $\bar{a} \in \mathbb{Z}_n$ , pois  $\bar{a} + \overline{n-a} = \overline{a + (n-a)} = \overline{n} = \bar{0}$ , pois  $n \cong 0 \text{ mod } n$ . Então:  $-\bar{a} = \overline{n-a}$ .

Portanto,  $(\mathbb{Z}_n, +)$  é um grupo comutativo,  $\forall n > 1 \in \mathbb{Z}$ , o grupo aditivo da classe de restos módulo  $n$ . □

**Exemplo 2.** Considere  $\mathbb{Z}_4 = \{\bar{0}, \bar{1}, \bar{2}, \bar{3}\}$ , um grupo finito de ordem 4 em relação à adição, cuja tábua é apresentada na Tabela 2.

+	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$
$\bar{0}$	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$
$\bar{1}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{0}$
$\bar{2}$	$\bar{2}$	$\bar{3}$	$\bar{0}$	$\bar{1}$
$\bar{3}$	$\bar{3}$	$\bar{0}$	$\bar{1}$	$\bar{2}$

Tabela 2 – Tábua adição em  $\mathbb{Z}_4 = \{\bar{0}, \bar{1}, \bar{2}, \bar{3}\}$ .

Fonte: Adaptado de [11].

### 2.2.3 Anéis

**Definição 6.** Um anel  $\langle A, +, \cdot \rangle$  é um conjunto  $A$  junto com duas operações binárias  $+$  e  $\cdot$  (as quais chamamos adição e multiplicação) tal que as seguintes propriedades são satisfeitas:

- $\langle A, + \rangle$  é um grupo abeliano;

- *A multiplicação é associativa;*
- *Para todos  $a, b, c \in A$ , valem a lei distributiva à esquerda,  $a(b + c) = (ab) + (ac)$  e a lei distributiva à direita,  $(a + b)c = (ac) + (bc)$ .*

Se além disso,  $A$  satisfizer:

- *Existe um elemento  $1 \in A$  (com  $1 \neq 0$ ) tal que  $a \cdot 1 = 1 \cdot a = a$  para todo  $a \in A$ , então  $\langle A, +, \cdot \rangle$  é um anel com unidade.*
- *$a \cdot b = b \cdot a$  para todos  $a, b \in A$ , então  $\langle A, +, \cdot \rangle$  é um anel comutativo.*
- *$a \cdot b = 0$  implica  $a = 0$  ou  $b = 0$ , então  $\langle A, +, \cdot \rangle$  é um anel sem divisores de zero.*

Se  $\langle A, +, \cdot \rangle$  é um anel comutativo, com unidade e sem divisores de zero, chamamos  $\langle A, +, \cdot \rangle$  de domínio de integridade.

**Teorema 4.** *Em um anel  $A$ , o elemento neutro da adição  $0_A$  é único.*

*Demonstração:* Suponhamos que existam dois elementos neutros  $0_A, 0'_A$ . Então:

$$0_A = 0_A + 0'_A = 0'_A, \text{ ou seja, } 0_A = 0'_A.$$

Logo, existe um único elemento neutro em  $A$ . □

**Teorema 5.** *Para cada elemento  $a \in A$ , o elemento oposto é único.*

*Demonstração:* Sejam  $a$  e  $a'$  os inversos de  $a \in A$ . Então:

$$a + (-a) = 0 = (-a) + a \text{ e } a' + (-a) = 0 = (-a) + a'.$$

Logo

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

Portanto:  $a = a'$ . □

**Exemplo 3.** *São anéis: o conjunto dos números inteiros sob adição e multiplicação; o conjunto dos números racionais sob adição e multiplicação; o conjunto  $\{\overline{0}, \overline{1}, \dots, \overline{n-1}\}$  sob as operações de soma e produto módulo  $n$ , etc.*

**Definição 7.** *Diz-se que  $N$  é um subanel de um anel  $A$  se  $N \subseteq A$  e  $N$  também forma um anel com as operações  $+$  e  $\cdot$  herdadas de  $A$ .*

## 2.2.4 Corpos

**Definição 8.** *Seja  $K$  um conjunto não vazio munido de duas operações, adição  $+$  e multiplicação  $\cdot$ , que associam cada par de elementos  $a, b \in K$  a um outro elemento do conjunto. Dizemos que  $K$  forma um corpo quando satisfaz os seguintes axiomas:*

- *Com relação à adição,  $K$  constitui um grupo abeliano, o que implica nas propriedades:*
  - *Associativa: para quaisquer  $a, b, c \in K$ , a soma  $(a + b) + c$  é idêntica a  $a + (b + c)$ .*
  - *Comutativa: a ordem dos termos não altera o resultado da soma, ou seja,  $a + b = b + a$  para todos  $a, b \in K$ .*
  - *Elemento neutro: existe um elemento  $0_K \in K$  tal que  $a + 0_K = 0_K + a = a$  para qualquer  $a \in K$ .*
  - *Elemento Inverso: para cada  $a \in K$ , há um elemento  $-a \in K$  que satisfaz  $a + (-a) = (-a) + a = 0_K$ .*
- *Com relação à multiplicação,  $K$  constitui um grupo abeliano sem o elemento  $0_K$ , o que implica:*
  - *Associativa: para quaisquer  $a, b, c \in K$ , o produto  $a \cdot (b \cdot c)$  é idêntico a  $(a \cdot b) \cdot c$ .*
  - *Comutativa: a ordem dos fatores não altera o produto, ou seja,  $a \cdot b = b \cdot a$  para todos  $a, b \in K$ .*
  - *Elemento neutro: existe um elemento  $1_K \in K$  tal que  $a \cdot 1_K = 1_K \cdot a = a$  para qualquer  $a \in K$ .*
  - *Elemento inverso: para cada  $a \in K$ ,  $a \neq 0_K$ , há um elemento  $a^{-1} \in K$  que satisfaz  $a \cdot a^{-1} = 1_K$ .*
- *Distributiva da multiplicação em relação à adição: para todos  $a, b, c \in K$ , vale que  $a \cdot (b + c) = a \cdot b + a \cdot c$ .*

## Corpos Finitos

**Definição 9.** *Um corpo finito, ou campo finito, contém um número finito de elementos. Também são conhecidos como corpos de Galois.*

Valem as seguintes propriedades e detalhes adicionais podem ser verificados em [13].

- **Existência e Unicidade:** Para cada número primo  $p$  e inteiro positivo  $n$ , existe exatamente um corpo finito com  $p^n$  elementos, denotado por  $\mathbb{F}_{p^n}$  ou  $GF(p^n)$ .
- **Estrutura Cíclica:** O grupo multiplicativo de um corpo finito  $\mathbb{F}_{p^n}$  (ou seja,  $\mathbb{F}_{p^n}$  sem o elemento 0) é sempre cíclico.

## 2.3 Modelagem do Código Genético

Nesta seção serão apresentados os conceitos fundamentais sobre os rotulamentos associados ao mapeamento do código genético e os algoritmos utilizados como referência para o desenvolvimento deste trabalho. As referências utilizadas, onde informações adicionais poderão ser encontradas foram: [2], [3], [6], [14] e [15].

### 2.3.1 Rotulamentos

O mapeamento do código genético pode ser compreendido por meio de rotulamentos específicos, que estabelecem uma relação entre as estruturas biológicas e algébricas analisadas. A utilização dos rotulamentos facilita a construção de modelos algébricos que podem ser utilizados para prever o comportamento de sequências genéticas sob diversas condições, além de possibilitar a detecção de mutações e a análise da estabilidade genética, conforme apresentado em [2], [3] e [14]. Esta relação é essencial, pois permite identificar padrões e regularidades que não são visíveis apenas pela observação biológica direta.

Os rotulamentos do código genético foram inicialmente propostos por [14], como uma bijeção entre um alfabeto biológico  $N = \{A, C, G, T/U\}$ , representado pelas bases nitrogenadas (adenina - A, citosina - C, guanina - G, timina/uracila - T/U), e um alfabeto algébrico, representado pelo anel  $\mathbb{Z}_4 = \{0, 1, 2, 3\}$ . Esse mapeamento, assim como descrito na Figura 8, é composto por 24 permutações, divididas em três rotulamentos, denominados A, B e C, de acordo com a característica geométrica associada a cada um deles. Cada rotulamento possui características únicas que influenciam a modelagem das sequências genéticas.

Nos trabalhos propostos por [2] e [14], foram realizadas uma série de análises e interpretações de sequências de DNA, por meio das quais podem ser verificadas mutações e como as mesmas podem ser identificadas por meio da identificação do rotulamento associado no processo de geração de proteínas, onde sequências de diversos comprimentos foram testadas e validadas por meio da implementação de um algoritmo de geração de sequências de DNA. Percebe-se dessa forma uma interessante conexão entre elementos de Biologia e Álgebra, além de elementos de Engenharia, uma vez que essas sequências podem ser vistas como um processo de transmissão da informação em um sistema de comunicação genético.

No rotulamento A, a complementaridade algébrica (00-11/01-10) é diretamente associada à complementaridade biológica (A-T/C-G), resultando em um mapeamento denominado  $\mathbb{Z}_4$ -linear. Esse tipo de rotulamento é caracterizado por um mapeamento não-linear, em que um nucleotídeo precisa percorrer duas arestas para encontrar seu complementar biológico em uma configuração geométrica quadrática, apresentando a característica geométrica mencionada anteriormente [2].

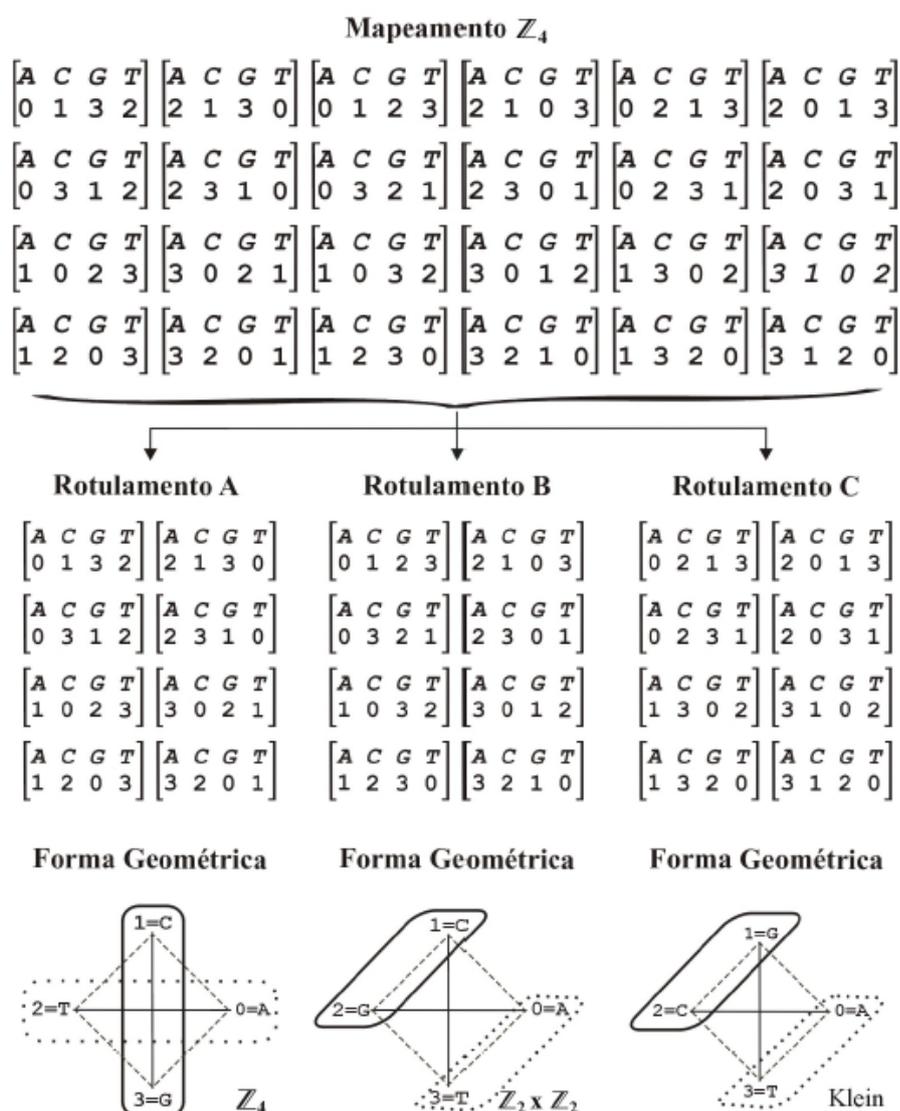


Figura 8 – Rotulamentos do código genético.  
 Fonte: [14].

Os rotulamentos B e C, por outro lado, não demonstram uma correspondência direta entre as complementaridades biológica e algébrica. O rotulamento B caracteriza-se pela união das bases A-G e C-T, configurando um mapeamento denominado  $\mathbb{Z}_2 \times \mathbb{Z}_2$ -linear [2].

O rotulamento C, seguindo uma lógica similar, associa as bases A-C e G-T, configurando um mapeamento denominado Klein-linear, em referência à garrafa de Klein, devido a uma torção existente no processo de mapeamento e complementaridade das bases. Nos rotulamentos B e C, as representações geométricas facilitam mapeamentos lineares, permitindo que os nucleotídeos encontrem seus complementares biológicos percorrendo apenas uma aresta [2].

Estes detalhes mencionados anteriormente foram utilizados nas modelagens propostas por [3], que também incorporou análises algébricas para explicar as relações entre os códons e suas propriedades de codificação e análise de características e propriedades físico-químicas dos aminoácidos do código genético [3].

A seguir serão apresentados os algoritmos SMG e soma com transporte. Ressalta-se que os casos analisados seguirão a ordem  $\{0, 1, 2, 3\}$ , onde para o rotulamento A considera-se  $\{G, U, C, A\}$ , para o rotulamento B  $\{A, C, G, U\}$  e para o rotulamento C  $\{U, A, G, C\}$ .

### 2.3.2 Algoritmo SMG

Este algoritmo proposto inicialmente por [6], foi utilizado em [3] para a realização de uma série de testes e que levaram à proposta de um novo algoritmo, o algoritmo soma com transporte. O algoritmo SMG, conforme detalhado em [3], segue os seguintes passos:

1. As bases correspondentes à terceira posição são adicionadas de acordo com a Tabela 3, dependendo do rotulamento.
2. Se a base resultante da operação soma é anterior à base adicionada (a ordem no conjunto de bases), então o novo valor é escrito e a base  $U$ ,  $C$  ou  $A$  é adicionada à próxima posição, respectivamente para os rotulamentos A, B e C.
3. As outras bases são adicionadas de acordo com a Tabela 3, passo 2, indo da primeira para a segunda base.

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

Tabela 3 – Tabelas soma: rotulamentos A, B e C, respectivamente.  
Fonte: Adaptado de [3].

**Exemplo 4.** Considere os códons  $CGU$  e  $GCA$ . Efetue a soma desses códons, utilizando as permutações  $3201$ ,  $0123$  e  $1320$ , referente aos rotulamentos A, B e C, respectivamente.

**Rotulamento A: 3201**

**Passo 1:**  $U + A = G$  (soma das bases da terceira posição). O resultado antecede à base A adicionada. Desta forma, é necessário adicionar U na próxima soma.

**Passo 2:**  $C + G = C + U = A$  (soma das bases da primeira posição).

**Passo 3:**  $G + C = C$  (soma das bases da segunda posição).

Logo, temos  $CGU + GCA = ACG$ .

**Rotulamento B: 0123**

**Passo 1:**  $U + A = U$  (soma das bases da terceira posição).

**Passo 2:**  $C + G = U$  (soma das bases da primeira posição).

**Passo 3:**  $G + C = U$  (soma das bases da segunda posição).

Logo, temos  $CGU + GCA = UUU$ .

**Rotulamento C: 1320**

**Passo 1:**  $U + A = A$  (soma das bases da terceira posição).

**Passo 2:**  $C + G = A$  (soma das bases da primeira posição). O resultado antecede à base  $G$  adicionada. Desta forma, é necessário adicionar  $A$  na próxima soma.

**Passo 3:**  $G + C = A + A = G$  (soma das bases da segunda posição).

Logo, temos  $CGU + GCA = AGA$ .

### 2.3.3 Algoritmo soma com transporte

Este algoritmo, proposto inicialmente por [3], foi implementado e testado no trabalho desenvolvido por [15].

De acordo com [3], o algoritmo soma com transporte segue os seguintes passos:

1. Especificar o rótulo utilizado.
2. Somar as terceiras bases em  $\mathbb{Z}_4$ . Se o valor encontrado for superior a 3, transporte 1 para a próxima soma.
3. Somar as primeiras bases em  $\mathbb{Z}_4$ , adicionando 1, caso a soma do passo 1 seja superior a 3. Se o valor encontrado for superior a 3, transporte 1 para a próxima soma.
4. Somar as segundas bases em  $\mathbb{Z}_4$ , adicionando 1, caso a soma anterior seja superior a 3.

**Exemplo 5.** Da mesma forma que o Exemplo 4 considere os códons  $CGU$  e  $GCA$  e efetue a soma desses códons, utilizando as permutações 3201, 0123 e 1320, referentes aos rotulamentos  $A$ ,  $B$  e  $C$ , respectivamente.

**Rotulamento A: 3201**

**Passo 1:** Considerando o rótulo, temos  $CGU + GCA = 201 + 023$ .

**Passo 2:**  $1 + 3 = 4 = 0 \pmod{4}$  (soma das terceiras bases).

**Passo 3:**  $2 + 0 + 1 = 3 \pmod{4}$  (soma das primeiras bases com o transporte de 1).

**Passo 4:**  $0 + 2 = 2 \pmod{4}$  (soma das segundas bases).

Portanto,  $201 + 023 = 320$ , que corresponde ao códon ACG.

### **Rotulamento B: 0123**

**Passo 1:** Considerando o rótulo, temos  $CGU + GCA = 123+210$ .

**Passo 2:**  $3+0=3 \pmod{4}$  (soma das terceiras bases).

**Passo 3:**  $1+2=3 \pmod{4}$  (soma das primeiras bases).

**Passo 4:**  $2+1=3 \pmod{4}$  (soma das segundas bases).

Portanto,  $123+210=333$ , que corresponde ao códon UUU.

### **Rotulamento C: 1320**

**Passo 1:** Considerando o rótulo, temos  $CGU + GCA = 320+231$ .

**Passo 2:**  $0+1=1 \pmod{4}$  (soma das terceiras bases).

**Passo 3:**  $3+2=5=1 \pmod{4}$  (soma das primeiras bases).

**Passo 4:**  $2+3+1=2 \pmod{4}$  (soma das segundas bases).

Portanto,  $320+231=121$ , que corresponde ao códon AGA.

## 2.4 Algoritmo Soma $\mathbb{Z}_{64}$

O algoritmo soma  $\mathbb{Z}_{64}$ , proposto inicialmente por [6], utilizado por [3] em uma série de análises e que será implementado neste trabalho, segue os seguintes passos:

1. Especificar o rótulo utilizado.
2. Identificar o elemento de  $\mathbb{Z}_{64}$  correspondente aos códons, multiplicando a primeira posição por  $4^1$ , a segunda posição por  $4^2$  e a terceira posição por  $4^0$  e, em seguida, somando esses produtos, ou seja, obedecendo a ordem de importância biológica das bases nitrogenadas no códon.
3. Somar os elementos correspondentes a cada códon reduzidos módulo 64.

Os cálculos a serem efetuados tomam como referência Tabelas 5, 6 e 7, as quais também serão utilizadas no algoritmo implementado deste trabalho, por meio do *Python*. Essas tabelas foram construídas a partir das permutações 3201, 0123 e 1320 associadas

aos rotulamentos A, B e C, respectivamente. Como as tabelas seguem a ordem  $\{0, 1, 2, 3\}$ , para o rotulamento A considera-se  $\{G, U, C, A\}$ , para o rotulamento B  $\{A, C, G, U\}$  e para o rotulamento C  $\{U, A, G, C\}$ .

Ressalta-se que os cálculos realizados em [3] foram todos manuais e neste trabalho os resultados foram obtidos por meio da implementação do algoritmo soma  $\mathbb{Z}_{64}$ .

A Tabela 4 apresenta uma representação dos aminoácidos e as abreviações que serão utilizadas no decorrer deste trabalho, como forma de simplificação.

Aminoácido	Abreviatura	Aminoácido	Abreviatura
Alanina	A	Leucina	L
Arginina	R	Lisina	K
Asparagina	N	Metionina	M
Ácido Aspártico	D	Fenilalanina	F
Cisteína	C	Prolina	P
Glutamina	Q	Serina	S
Ácido Glutâmico	E	Treonina	T
Glicina	G	Triptofano	W
Histidina	H	Tirosina	Y
Isoleucina	I	Valina	V
STOP	@		

Tabela 4 – Abreviações para os aminoácidos.  
Fonte: Adaptado de [3].

	G			U			C			A			
	$\mathbb{Z}_{64}$	Códon	Am.	$\mathbb{Z}_{64}$	Códon	Am.	$\mathbb{Z}_{64}$	Códon	Am.	$\mathbb{Z}_{64}$	Códon	Am.	
G	0	GGG	G	16	GUG	V	32	GCG	A	48	GAG	E	G
	1	GGU	G	17	GUU	V	33	GCU	A	49	GAU	D	U
	2	GGC	G	18	GUC	V	34	GCC	A	50	GAC	D	C
	3	GGA	G	19	GUA	V	35	GCA	A	51	GAA	E	A
U	4	UGG	W	20	UUG	L	36	UCG	S	52	UAG	STOP	G
	5	UGU	C	21	UUU	F	37	UCU	S	53	UAU	Y	U
	6	UGC	C	22	UUC	F	38	UCC	S	54	UAC	Y	C
	7	UGA	STOP	23	UUA	L	39	UCA	S	55	UAA	STOP	A
C	8	CGG	R	24	CUG	L	40	CCG	P	56	CAG	Q	G
	9	CGU	R	25	CUU	L	41	CCU	P	57	CAU	H	U
	10	CGC	R	26	CUC	L	42	CCC	P	58	CAC	H	C
	11	CGA	R	27	CUA	L	43	CCA	P	59	CAA	Q	A
A	12	AGG	R	28	AUG	M	44	ACG	T	60	AAG	K	G
	13	AGU	S	29	AUU	I	45	ACU	T	61	AAU	N	U
	14	AGC	S	30	AUC	I	46	ACC	T	62	AAC	N	C
	15	AGA	R	31	AUA	I	47	ACA	T	63	AAA	K	A

Tabela 5 – Código genético segundo o rotulamento A.  
Fonte: Adaptado de [3].

A			C			G			U				
	$\mathbb{Z}_{64}$	Códon	Am.	$\mathbb{Z}_{64}$	Códon	Am.	$\mathbb{Z}_{64}$	Códon	Am.	$\mathbb{Z}_{64}$	Códon	Am.	
A	0	AAA	K	16	ACA	T	32	AGA	R	48	AUA	I	A
	1	AAC	N	17	ACC	T	33	AGC	S	49	AUC	I	C
	2	AAG	K	18	ACG	T	34	AGG	R	50	AUG	M	G
	3	AAU	N	19	ACU	T	35	AGU	S	51	AUU	I	U
C	4	CAA	Q	20	CCA	P	36	CGA	R	52	CUA	L	A
	5	CAC	H	21	CCC	P	37	CGC	R	53	CUC	L	C
	6	CAG	Q	22	CCG	P	38	CGG	R	54	CUG	L	G
	7	CAU	H	23	CCU	P	39	CGU	R	55	CUU	L	U
G	8	GAA	E	24	GCA	A	40	GGA	G	56	GUA	V	A
	9	GAC	D	25	GCC	A	41	GGC	G	57	GUC	V	C
	10	GAG	E	26	GCG	A	42	GGG	G	58	GUG	V	G
	11	GAU	D	27	GCU	A	43	GGU	G	59	GUU	V	U
U	12	UAA	STOP	28	UCA	S	44	UGA	STOP	60	UUA	L	A
	13	UAC	Y	29	UCC	S	45	UGC	C	61	UUC	F	C
	14	UAG	STOP	30	UCG	S	46	UGG	W	62	UUG	L	G
	15	UAU	Y	31	UCU	S	47	UGU	C	63	UUU	F	U

Tabela 6 – Código genético segundo o rotulamento B.

Fonte: Adaptado de [3].

U			A			G			C				
	$\mathbb{Z}_{64}$	Códon	Am.	$\mathbb{Z}_{64}$	Códon	Am.	$\mathbb{Z}_{64}$	Códon	Am.	$\mathbb{Z}_{64}$	Códon	Am.	
U	0	UUU	F	16	UAU	Y	32	UGU	C	48	UCU	S	U
	1	UUA	L	17	UAA	STOP	33	UGA	STOP	49	UCA	S	A
	2	UUG	L	18	UAG	STOP	34	UGG	W	50	UCG	S	G
	3	UUC	F	19	UAC	Y	35	UGC	C	51	UCC	S	C
A	4	AUU	I	20	AAU	N	36	AGU	S	52	ACU	T	U
	5	AUA	I	21	AAA	K	37	AGA	R	53	ACA	T	A
	6	AUG	M	22	AAG	K	38	AGG	R	54	ACG	T	G
	7	AUC	I	23	AAC	N	39	AGC	S	55	ACC	T	C
G	8	GUU	V	24	GAU	D	40	GGU	G	56	GCU	A	U
	9	GUA	V	25	GAA	E	41	GGA	G	57	GCA	A	A
	10	GUG	V	26	GAG	E	42	GGG	G	58	GCG	A	G
	11	GUC	V	27	GAC	D	43	GGC	G	59	GCC	A	C
C	12	CUU	L	28	CAU	H	44	CGU	R	60	CCU	P	U
	13	CUA	L	29	CAA	Q	45	CGA	R	61	CCA	P	A
	14	CUG	L	30	CAG	Q	46	CGG	R	62	CCG	P	G
	15	CUC	L	31	CAC	H	47	CGC	R	63	CCC	P	C

Tabela 7 – Código genético segundo o rotulamento C.

Fonte: Adaptado de [3].

**Exemplo 6.** Considere os códons  $CGU$  e  $GCA$ , apresentados nos Exemplos 4 e 5. Efetue a soma dos códons  $CGU + GCA$ , utilizando o algoritmo soma em  $\mathbb{Z}_{64}$ .

**Rotulamento A: 3201**

**Passo 1:** Considerando a permutação 3201, temos  $CGU + GCA = 201 + 023$ .

**Passo 2:**  $CGU = 2 \times 4^1 + 0 \times 4^2 + 1 \times 4^0 = 9$  e  $GCA = 0 \times 4^1 + 2 \times 4^2 + 3 \times 4^0 = 35$ .

**Passo 3:**  $9 + 35 = 44 \pmod{64}$ , que corresponde ao códon  $ACG$ .

Portanto,  $CGU + GCA = ACG$ .

**Rotulamento B: 0123**

**Passo 1:** Considerando a permutação 0123, temos  $123+210$ .

**Passo 2:**  $CGU = 1 \times 4^1 + 2 \times 4^2 + 3 \times 4^0 = 38$  e  $GCA = 2 \times 4^1 + 1 \times 4^2 + 0 \times 4^0 = 24$ .

**Passo 3:**  $39+24 = 63 \pmod{64}$ , que corresponde ao códon  $UUU$ .

Portanto,  $CGU+GCA=UUU$ .

**Rotulamento C: 1320**

**Passo 1:** Considerando a permutação 1320, temos  $320+231$ .

**Passo 2:**  $CGU = 3 \times 4^1 + 2 \times 4^2 + 0 \times 4^0 = 44$  e  $GCA = 2 \times 4^1 + 3 \times 4^2 + 1 \times 4^0 = 57$ .

**Passo 3:**  $44+57 = 37 \pmod{64}$ , que corresponde ao códon  $AGA$ .

Portanto,  $CGU+GCA=AGA$ .

A partir do estudo, análise e comparações dos aspectos teóricos associados aos algoritmos SMG, soma com transporte e soma  $\mathbb{Z}_{64}$ , no próximo capítulo será apresentada a implementação e algumas análises associadas ao algoritmo soma  $\mathbb{Z}_{64}$ .

## 3 **IMPLEMENTAÇÃO E APLICAÇÕES DO ALGORITMO SOMA $\mathbb{Z}_{64}$**

Neste capítulo serão apresentados os resultados obtidos durante a realização deste Trabalho de Conclusão de Curso. Ressalta-se que foram utilizados elementos de Biologia, Álgebra e Ciência da Computação, mostrando uma importante caracterização interdisciplinar, envolvendo a modelagem do processo de análise de sequências de proteínas, com elementos do código genético.

Este capítulo está estruturado da seguinte forma: na seção 3.1 será apresentado o detalhamento do código utilizado para as implementações. O procedimento de implementação baseou-se em uma abordagem do *Python*. Em seguida, na seção 3.2 serão apresentados alguns exemplos envolvendo a aplicação do algoritmo e algumas análises dos resultados obtidos.

As referências utilizadas para o desenvolvimento destes resultados foram: [2], [3], [6], [14] e [15].

### 3.1 Código Utilizado

Nesta seção será apresentada a implementação do algoritmo soma  $\mathbb{Z}_{64}$ , descrito por [3] e [6]. Foram utilizadas as permutações 3201, 0123 e 1320, para os rotulamentos A, B e C, respectivamente.

O algoritmo segue uma série de etapas durante a realização das operações. Essas etapas são detalhadas a seguir:

1. Defina o rotulamento a ser utilizado;
2. Os valores correspondentes a cada códon são calculados de acordo com as posições de cada base nitrogenada, sendo a primeira posição multiplicada por  $4^1$ , a segunda posição por  $4^2$  e a terceira posição por  $4^0$ . Em seguida realiza-se a soma desses valores, gerando o resultado final, de forma que se respeite a ordem de importância biológica das bases nitrogenadas.
3. Some os valores obtidos para cada códon e, em seguida, aplique a congruência mod 64, para obter os resultados finais em  $\mathbb{Z}_{64}$ .

O código a ser implementado tem o objetivo de automatizar as operações de soma entre códons, acelerando e facilitando o processo, permitindo assim analisar sequências de

comprimentos maiores, de forma eficiente. Essa automatização é especialmente vantajosa para análises de grandes volumes de dados genéticos, algo comum em estudos de genômica moderna. Ressalta-se que neste trabalho o foco principal está na implementação do algoritmo e a análise em sequências menores, deixando como sugestões para trabalhos futuros as análises de sequências maiores e possivelmente reais.

A implementação do algoritmo foi realizada utilizando a linguagem de programação *Python*, devido à sua simplicidade, além de ser conhecida por suas capacidades de manipulação de dados. A seguir, será apresentado o pseudocódigo do algoritmo soma  $\mathbb{Z}_{64}$ .

### Pseudocódigo: Algoritmo Soma $\mathbb{Z}_{64}$

1. **Início do Algoritmo;**
2. Selecione um rotulamento entre A, B ou C;
3. Escolha um método de soma:
  - 4.1 **Se** o método de soma for códon a códon:
    - 4.1.1 Determine o número de pares de códons a serem somados.
    - 4.1.2 **Repetir** para cada par de códons:
      - Especificar o primeiro e o segundo códon.
      - Converter os códons de acordo com a posição de cada base;
      - Calcular a soma;
      - Exibir o resultado.
      - **Se** um códon de parada for encontrado, encerrar o processo.
  - 4.2 **Se** o método de soma for sequência completa:
    - 4.2.1 Solicitar ao usuário a primeira sequência.
    - 4.2.2 Solicitar ao usuário a segunda sequência.
    - 4.2.3 Converter ambas as sequências de acordo com a posição de cada base.
    - 4.2.4 **Repetir** para cada par de 3 bases:
      - Calcular a soma dos elementos.
      - **Se** um códon de parada for encontrado, encerrar o processo.
    - 4.2.5 Exibir a sequência final composta pelos códons e aminoácidos resultantes.
4. **Fim do Algoritmo**

O fluxograma da Figura 9 apresenta a estrutura do código implementado.

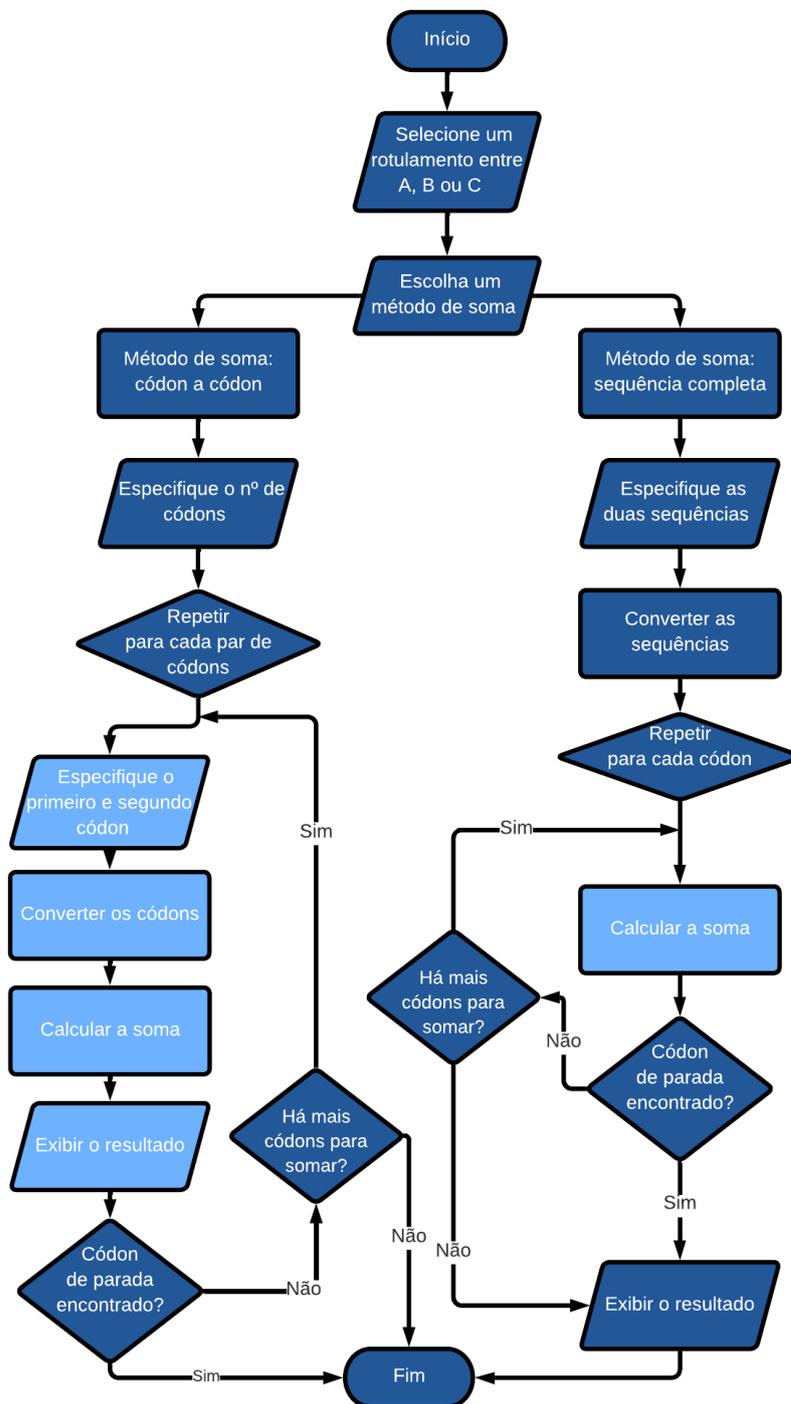


Figura 9 – Fluxograma do código.  
Fonte: O autor.

A implementação oferece ao usuário a possibilidade de optar entre os 3 rotulamentos a partir das permutações 3201, 0123 e 1320, para os rotulamentos A, B e C, respectivamente, permitindo testar diversas hipóteses. Além disso, o código incorpora validações para garantir a integridade das entradas do usuário e interrompe as operações caso seja

identificado um códon de parada, fato que ocorre no meio biológico no processo de síntese proteica, que é interrompida a partir do momento que ocorre o códon de parada STOP.

O código oferece dois métodos de soma. O método códon a códon oferece precisão na análise, onde cada operação é detalhada e, no final é exibido o códon resultante e o aminoácido correspondente.

Por outro lado, o método de soma com a sequência completa (comprimentos a serem determinados pelo usuário), possibilita o estudo e análise de sequências maiores, oferecendo uma visão mais ampla para futura análise de dados reais, possibilitando a redução de trabalhos manuais. Ao final da operação, o código apresenta a sequência resultante completa.

Para assegurar a consistência dos resultados, o método de soma com a sequência completa verifica se a primeira sequência inserida é um múltiplo de 3 (devido ao códon conter 3 bases) e se a segunda sequência possui a mesma dimensão que a primeira, de modo a prevenir a ocorrência de erros.

## 3.2 Aplicações do Algoritmo Soma $\mathbb{Z}_{64}$ envolvendo os Rotulamentos do Código Genético

Nesta seção serão apresentados alguns testes realizados com o código descrito na seção 3.1. Foram utilizados os dois tipos de somas disponíveis no código: a soma códon a códon e a soma com sequências completas.

A seguir, serão descritos os testes realizados, utilizando os três rotulamentos A, B e C, e os resultados obtidos para cada tipo de soma.

### 3.2.1 Testes com o método de soma códon a códon

Serão apresentados inicialmente alguns exemplos envolvendo a soma de códons, para o caso mais simples, envolvendo a soma códon a códon, com situações escolhidas aleatoriamente dentre os 3 rotulamentos do código genético e as permutações associadas a esses rotulamentos: A - 3201, B - 0123 e C - 1320.

Ressalta-se que nesse estágio inicial o objetivo principal é testar a validade do código e possibilitar a aplicação para sequências maiores, a fim de analisar situações, como os casos envolvendo mutações genéticas.

**Exemplo 7.** Neste teste será realizada a soma entre os códons CGU, que codifica o aminoácido arginina e GCA, que codifica o aminoácido alanina.

- Número de códons: 1

- *Códons utilizados: CGU, GCA*

A Tabela 8 apresenta os resultados da soma entre os códons apresentados anteriormente. Ressalta-se que neste exemplo e nos demais serão utilizadas as abreviações *Rot.* para Rotulamento, *Perm.* para Permutação e *Am.* para Aminoácido.

Rot.	Perm.	Códons	$\mathbb{Z}_{64}$	Soma	Resultado	Am.
A	[3, 2, 0, 1]	CGU + GCA	9 + 35	44	ACG	Treonina
B	[0, 1, 2, 3]	CGU + GCA	39 + 24	63	UUU	Fenilalanina
C	[1, 3, 2, 0]	CGU + GCA	44 + 57	37	AGA	Arginina

Tabela 8 – Resultados da soma com um par de códons.

Fonte: O autor.

Para os rotulamentos A, B e C, pode-se perceber resultados distintos, devido às diferentes permutações aplicadas e às tabelas construídas para cada um dos rotulamentos e sua utilização na implementação do algoritmo, demonstrando a variabilidade que elas podem gerar. Os resultados obtidos a partir da implementação do algoritmo validam os resultados apresentados nas Tabelas 5, 6 e 7 do código genético, apresentadas no capítulo anterior.

**Exemplo 8.** *Considere agora a soma entre as seqüências GUGGCG e AAUCUA, formadas a partir de 2 pares de códons, cujos aminoácidos associados estão descritos nos códons utilizados, apresentados a seguir. Os resultados serão detalhados e resumidos na Tabela 9.*

- *Número de códons: 2*
- *Códons utilizados:*
  - *Par 1: GUG (Valina) + AAU (Asparagina)*
  - *Par 2: GCG (Alanina) + CUA (Leucina)*

Rot.	Perm.	Códons	$\mathbb{Z}_{64}$	Soma	Resultado	Am.
A	[3, 2, 0, 1]	GUG + AAU	16 + 61	13	AGU	Serina
		GCG + CUA	32 + 27	59	CAA	Glutamina
B	[0, 1, 2, 3]	GUG + AAU	58 + 3	61	UUC	Fenilalanina
		GCG + CUA	26 + 52	14	UAG	STOP
C	[1, 3, 2, 0]	GUG + AAU	10 + 20	30	CAG	Glutamina
		GCG + CUA	58 + 13	7	AUC	Isoleucina

Tabela 9 – Resultados da soma com dois pares de códons.

Fonte: O autor.

A presença de um códon de parada (STOP) na tradução relacionada ao rotulamento B demonstra a capacidade do algoritmo em detectar as instruções de parada, o que muitas vezes podem ocorrer em sequências reais, nas quais a presença de um códon STOP pode interromper a síntese proteica e acarretar malefícios, como em muitos casos ocorridos em mutações genéticas.

### 3.2.2 Testes com o método de soma com a sequência completa

Nesta subseção serão apresentados alguns exemplos envolvendo a aplicação do algoritmo soma  $\mathbb{Z}_{64}$ , para casos envolvendo sequências fictícias de comprimento 18, 36 e 63. Ressalta-se que nesses casos, a importância da utilização da soma entre códons se dá pela possibilidade de um determinado organismo ser afetado por exemplo, por uma bactéria e nesse caso ocorrer a associação entre o material genético do organismo e o da bactéria, podendo acarretar em doenças e possivelmente algum malefício para o organismo. Não serão analisados neste trabalho esses malefícios e mutações, no entanto, fica como sugestão para futuras pesquisas a análise desses impactos no organismo, utilizando tanto o algoritmo soma  $\mathbb{Z}_{64}$ , que é o foco deste trabalho, quanto os outros algoritmos mencionados e realizar comparações, além da análise das implementações e aplicações.

**Exemplo 9.** *Neste teste será efetuada a soma entre as sequências fictícias CGAAAU-GUCCGGGGGCAU e ACUUUGAGCCGGCGAGUC, escolhidas de forma aleatória, para continuar a realização dos testes do algoritmo.*

- *Número de códons: 6*
- *Sequências utilizadas:*
  - *Sequência 1: CGAAAU-GUCCGGGGGCAU*
  - *Sequência 2: ACUUUGAGCCGGCGAGUC*

#### **Resultados:**

<b>Rot.</b>	<b>Perm.</b>	<b>Sequência Resultante</b>
A	[3, 2, 0, 1]	CAGGUUGCGGUGCGACGA
B	[0, 1, 2, 3]	CUUAACGCGUAA
C	[1, 3, 2, 0]	UGA

Tabela 10 – Resultados da soma com 6 códons.

Fonte: O autor.

*Observe que originalmente as sequências 1 e 2 eram compostas pelos aminoácidos RNVRGH e TLSRRV, respectivamente, conforme Tabela 4 e nas sequências resultantes para cada um dos rotulamentos, temos os seguintes aminoácidos associados, que podem ser verificados na Tabela 11.*

Rot.	Aminoácidos
A	QVAVRR
B	LNA@
C	@

Tabela 11 – Aminoácidos associados à soma com 6 códons.  
Fonte: O autor.

Pode-se perceber que se compararmos os resultados dos rotulamentos A, B e C, os aminoácidos obtidos indicam que apenas no rotulamento A a soma foi realizada até o fim da sequência. Diferente dos rotulamentos B e C, que tiveram suas operações interrompidas devido ao códon de parada encontrado, representado por "@" na sequência de aminoácidos.

**Exemplo 10.** Agora será apresentada a operação entre as sequências fictícias AGGGCUAGUUCGCGAAAAACGGGUUUGUCAACGGUC e AAGACUCGUUUGGCAUGCCGU-GUAAUCCUGUGUUGG, com comprimento 36 nucleotídeos. As análises e interpretações dos resultados são apresentados a seguir:

- Número de códons: 12
- Sequências utilizadas:
  - Sequência 1: AGGGCUAGUUCGCGAAAAACGGGUUUGUCAACGGUC
  - Sequência 2: AAGACUCGUUUGGCAUGCCGUGUAAUCCUGUGUUGG

### Resultados:

Rot.	Perm.	Sequência Resultante
A	[3, 2, 0, 1]	CGGAGCUUCCAGACCUGUUAUUUGGACAAAGAUUUC
B	[0, 1, 2, 3]	CGAUGGGAGUCAUUAUGCGUCAGUUGUACGAACCGU
C	[1, 3, 2, 0]	CCUCGUUAUACUAGGGCUUGGUCAGUACCCAAGCGA

Tabela 12 – Resultados da soma com 12 códons.  
Fonte: O autor.

Da mesma forma que foi realizado no Exemplo 9, neste caso, as sequências originais eram compostas pelos aminoácidos RASSRKTGLSTV e KTRLACRVILCW, respectivamente, e as sequências resultantes apresentam os aminoácidos associados conforme Tabela 13.

Rot.	Aminoácidos
A	RSFQTCYLDKDF
B	RWESLCVSCTNR
C	PRYTRAWSVPKR

Tabela 13 – Aminoácidos associados à soma com 12 códons.  
Fonte: O autor.

Ao contrário dos resultados observados no exemplo anterior, neste caso, as operações foram concluídas integralmente para os rotulamentos  $A$ ,  $B$  e  $C$ . Assim, cada um dos três rotulamentos produziu uma sequência completa de 12 aminoácidos, sem a interrupção causada por códons de parada.

**Exemplo 11.** Considere agora as sequências de comprimento 63 nucleotídeos. Os resultados e a análise são apresentados a seguir:

- Número de códons: 21
- Sequências utilizadas:
  - Sequência 1: AAGUCCAGGGCCAGGUCACGGGGGUCGUAAGCGCUAA-GAGGAGCCGUUGUUUCACUUGAACGA
  - Sequência 2: CGUUGAAGUUCGGGGACGGCAAUGCGUUGCCUAGCCU-CUAUUACGGCACACCGGGCACAGUCG

### Resultados:

Rot.	Perm.	Sequência Resultante
A	[3,2,0,1]	UGUACUCUUUGCAGGGUACCAAUGACUAAUCAAAAUUAG
B	[0,1,2,3]	GGCGACCACCUUUA
C	[1,3,2,0]	UUGAAUGUGCGAUUAUAGCAGCUCUCAGACUAUCGUUAAACGAUCAUUAAGCCACAUAACCCAC

Tabela 14 – Resultados da soma com 21 códons.

Fonte: O autor.

As sequências iniciais consistiam dos aminoácidos  $KSRARSRGS@ALRGAVVSLER$  e  $R@SSGTAMRCLASITAHRAQS$  para as sequências 1 e 2, respectivamente. As sequências finais resultantes possuem os aminoácidos associados, conforme mostrado na Tabela 15.

Rot.	Aminoácidos
A	CTLCRVPMTNQN@
B	GDHL@
C	LNVRYSSSQTIVKRSLSHITH

Tabela 15 – Aminoácidos associados à soma com 21 códons.

Fonte: O autor.

É evidente que apenas no rotulamento  $C$  foi possível realizar a soma dos códons integralmente, resultando em uma sequência de 21 aminoácidos. Em contrapartida, os rotulamentos  $A$  e  $B$  tiveram suas operações interrompidas devido ao encontro de códons de parada, representados por "@".

Por meio dos exemplos apresentados pode-se perceber a importância do processo de implementação do algoritmo soma  $\mathbb{Z}_{64}$  na análise de possíveis interferências que podem ocorrer com a associação de duas sequências de proteínas. Ressalta-se que neste trabalho

foram utilizadas sequências fictícias, deixando como proposta futura a busca de dados em sequências reais, a fim de analisar os impactos dos cálculos realizados em situações associadas a estudos de fenômenos mutacionais e se isso acarretaria malefícios ao indivíduo.

## 4 **CONSIDERAÇÕES FINAIS**

Este trabalho baseou-se na implementação e análise do algoritmo soma  $\mathbb{Z}_{64}$ , explorando sua aplicabilidade na modelagem matemática de sequências de proteínas. O objetivo principal foi investigar como esse algoritmo pode ser utilizado para analisar e prever comportamentos genéticos, com particular atenção às implicações de mutações e à síntese de proteínas.

A metodologia adotada envolveu a modelagem do código genético através de ferramentas matemáticas e computacionais, especificamente com o desenvolvimento do algoritmo em *Python*. Foram realizados diversos testes com sequências genéticas fictícias para avaliar a eficácia do algoritmo. Esses testes permitiram não apenas validar o algoritmo, mas também explorar como diferentes configurações do modelo podem afetar os resultados da análise genética.

Os resultados obtidos demonstraram que a implementação do algoritmo soma  $\mathbb{Z}_{64}$  é eficaz na análise de sequências genéticas, proporcionando uma compreensão significativa sobre o mapeamento e as possíveis mutações associadas. Este estudo contribuiu ao demonstrar a aplicabilidade de conceitos matemáticos na solução de problemas biológicos e ao desenvolver uma ferramenta que pode ser utilizada por outros pesquisadores para realizar análises genéticas mais aprofundadas, incluindo análises em condições reais.

Estudos futuros poderiam explorar a aplicação do algoritmo soma  $\mathbb{Z}_{64}$  em sequências genéticas reais e em maior escala, superando a limitação da utilização de sequências genéticas fictícias, presentes neste estudo. Além disso, a implementação do algoritmo para realizar operações com todas as 24 permutações associadas aos rotulamentos A, B e C, e também para permitir operações entre casos dual e primal, conforme explorado em Oliveira (2012), possibilitando uma aplicação mais abrangente e flexível do algoritmo em diferentes contextos genéticos.

A integração das áreas de Biologia, Matemática e Computação neste trabalho foi essencial para minha formação, destacando a importância da interdisciplinaridade no ensino. Esta experiência mostrou como o conhecimento integrado pode enriquecer o aprendizado, permitindo aplicar teoria à prática em contextos reais e complexos.

Os resultados obtidos durante a realização deste trabalho serão submetidos para possível apresentação no evento WMMA - Workshop de Matemática e Matemática Aplicada, a ser realizado em Ouro Preto-MG, no segundo semestre de 2024.

# Referências

- [1] FERNANDES, R. S.; OLIVEIRA, A. J. Análise das propriedades físico-químicas dos aminoácidos por meio das distâncias de Hamming associadas ao rotulamento B do código genético. **Revista Mundi, Engenharia e Gestão**, Paranaguá-PR, v. 6, n. 3, p. 364-01, 364-10, 2021.
- [2] FARIA, L. C. B. **Existências de Códigos Corretores de Erros e Protocolos de Comunicação em Sequências de DNA**. Tese (Doutorado em Engenharia Elétrica) - Universidade Estadual de Campinas, 2011.
- [3] OLIVEIRA, A. J. **Análise Algébrica dos Rotulamentos Associados ao Mapeamento do Código Genético**. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Estadual de Campinas, 2012.
- [4] FERNANDES, R. S. **Aplicações Do Diagrama De Hasse Na Análise Das Propriedades Dos Aminoácidos Do Código Genético**. Dissertação (Mestrado em Estatística Aplicada e Biometria) – Universidade Federal de Alfenas, 2021.
- [5] FERNANDES, R. S.; OLIVEIRA, A. J. Caracterização das propriedades dos aminoácidos por meio do Diagrama de Hasse associado ao rotulamento A do código genético. **Brazilian Electronic Journal of Mathematics**, v.2 - n.4, jul/dez, 2021.
- [6] SANCHEZ R.; MORGADO, E.; GRAU, R. Gene algebra from a genetic code algebraic structure, **J.Math.Biol.**, v. 51 pp. 431–457, 2005.
- [7] ALBERTS, B. et al. **Biologia Molecular da Célula**. 6. ed. Porto Alegre: Artmed, 2017.
- [8] MARZZOCO, A.; TORRES, B.B. **Bioquímica Básica**. 3. ed. [S.l]: Guanabara-Koogan, 2013.
- [9] SNUSTAD, D. P.; SIMMONS, M. J. **Fundamentos de Genética**. 7. ed. Rio de Janeiro: Guanabara Koogan, 2017.
- [10] JUNQUEIRA, L. C.; CARNEIRO, J. **Biologia Celular e Molecular**. 9. ed. São Paulo: Guanabara Koogan, 2005.
- [11] DOMINGUES, H. H.; IEZZI, G. **Álgebra Moderna**. 4. ed. São Paulo: Atual, 2003.
- [12] GARCIA, A.; LEQUAIN, Y. **Elementos de Álgebra**. 5. ed. Rio de Janeiro: IMPA, 2010.

- 
- [13] LIN, S.; COSTELO JR, D. J. **Error Control Coding: Fundamentals and Applications**. New Jersey: Prentice Hall, Englewood Clisffs, 1983.
- [14] ROCHA, A. S. L. **Modelo de Sistema de Comunicações Digital para o Mecanismo de Importação de Proteínas Mitocondriais através de Códigos Corretores de Erros**. Tese (Doutorado em Engenharia Elétrica) - Universidade Estadual de Campinas, 2010.
- [15] FERNANDES, J. S. **Análise Mutacional via Implementação do Algoritmo Soma com Transporte**. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) - Universidade Federal de Alfenas, 2020.