



**UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE MATEMÁTICA**

RENAN DE LANA GOMES

AUTÔMATOS CELULARES NA BIOLOGIA

VIÇOSA
2025

RENAN DE LANA GOMES


AUTÔMATOS CELULARES NA BIOLOGIA

Trabalho de Conclusão de Curso apresentado ao curso de Licenciatura em Matemática da Universidade Federal de Viçosa como requisito parcial para obtenção do título de Licenciado em Matemática.

Orientadora: Pouya Mehdipour


VIÇOSA
2025

RENAN DE LANA GOMES


Documento assinado digitalmente
 **RENAN DE LANA GOMES**
Data: 15/12/2025 14:00:47-0300
Verifique em <https://validar.iti.gov.br>

AUTÔMATOS CELULARES NA BIOLOGIA


Trabalho de Conclusão de Curso aprovado em 26 de novembro de 2025, pela banca examinadora composta por:

Documento assinado digitalmente
 **POUYA MEHDIPOUR**
Data: 11/12/2025 18:11:56-0300
Verifique em <https://validar.iti.gov.br>

Profa. Dra. Pouya Mehdipour
(Orientadora – Universidade Federal de Viçosa)

Documento assinado digitalmente
 **WALTER TEOFILU HUARACA VARGAS**
Data: 13/12/2025 08:39:01-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Walter Teófilo Huaraca Vargas
(Professor – Universidade Federal de Viçosa)

Documento assinado digitalmente
 **LUCY TIEMI TAKAHASHI**
Data: 12/12/2025 01:52:35-0300
Verifique em <https://validar.iti.gov.br>

Profa. Dra. Lucy Tiemi Takahashi
(Professora – Universidade Federal de Juiz de Fora)

VIÇOSA
2025

DEDICATÓRIA

Dedico este trabalho aos meus pais, aos meus amigos e colegas de curso. Em especial a minha irmã Luciana e meu amigo Jablonski.

Com gratidão e carinho, por todo apoio e incentivo nesta caminhada.

EPÍGRAFE

"Eadem Mutata Resurgo"

Jacob Bernoulli

Resumo

Autômatos Celulares (ACs) são sistemas dinâmicos discretos compostos por uma grade de células, cada uma em um estado finito, que evoluem em passos de tempo discretos segundo regras locais de transição. Diante disso, este trabalho investiga as aplicações de autômatos celulares na biologia, com foco na modelagem da relação entre genótipo (as regras de transição do autômato) e fenótipo (o comportamento global emergente) através de estudo de mutações genéticas. A metodologia que usamos baseia-se na revisão bibliográfica e na implementação computacional em Python para simular ACs, analisar sua estabilidade frente a perturbações (variações no genótipo ou mutações) e explorar conceitos como o parâmetro λ e a classificação de Wolfram. Os resultados obtidos mostraram evidências que incluem uma melhor compreensão de como operações genéticas simples podem gerar complexidade fenotípica, oferecendo insights para áreas como biomatemática e bioinformática.

Palavras-chave: Sistemas Dinâmicos. Genótipo. Fenótipo. Biomatemática. Python. Mutação.

Abstract

Cellular Automata (CAs) are discrete dynamical systems composed of a grid of cells, each in a finite state, which evolve in discrete time steps according to local transition rules. This work investigates the applications of cellular automata in biology, focusing on modeling the relationship between genotype (the automaton's transition rules) and phenotype (the emergent global behavior) through the genetic mutations. The methodology is based on a literature review and computational implementations in Python to simulate CAs, analyze their stability under perturbations (variations in the genotype or mutations), and explore concepts such as the λ parameter and Wolfram's classification. The results obtained provide evidence for a deeper understanding of how simple genetic operations can generate phenotypic complexity, offering insights for fields such as biomathematics and bioinformatics.

Keywords: Dynamical Systems. Genotype. Phenotype. Biomathematics. Python. Mutation.

Sumário

1	9
2	11
2.1	Objetivo Geral	11
2.2	Objetivos Específicos	11
3	12
3.1	Matemática	12
3.1.1	Sistemas Dinâmicos - Caos de Devaney	12
3.1.2	Espaço de Sequências $\Sigma = \{0, 1\}^{\mathbb{Z}}$	15
3.1.3	Autômatos Celulares na Computação (definição)	16
3.1.3.1	Condições de Fronteira	17
3.1.3.2	A Métrica de Hamming	17
3.1.4	ACE - Autômatos Celulares Elementares	18
3.1.5	Classificação de Wolfram para ACE	18
3.1.5.1	Parâmetro λ	20
3.1.6	Perturbações nas Regras de Wolfram	21
3.1.7	Autômatos Celulares Áureos	22
3.2	Biologia	24
3.2.1	Genótipo e Fenótipo em AC	24
3.2.2	Modelagem de Processos	25
3.2.3	Dinâmica de Populações e Evolução	25
3.2.4	Estabilidade Fenotípica e Mutação	26
3.2.5	DNA e RNA	26
4	29
4.1	Primeira Fase	29
4.2	Segunda Fase	30
4.2.0.1	Passo I	31
4.2.1	Passo II	39
4.3	Fase Final	46
4.3.1	Passo I	47
4.3.2	Passo II	48
5	51

	REFERÊNCIAS	52
6	54
	Apêndice A - Geração de Regras e Perturbações	54
	Apêndice B - Geração de Imagens Coloridas	56

1 Introdução

Este trabalho é continuação de dois projetos de iniciação científica desenvolvidos no período de 2023 a 2025, os projetos intitulado "**Caos e Fractais I e II**". Nesses projetos desenvolvemos estudos sobre a base teórica necessária para sistemas dinâmicos, dinâmica simbólica, autômatos celulares. Além disso, utilizamos a programação em Python para estudar autômatos binários com blocos proibidos, denominados **Autômatos Celulares Áureos**, explorando desde definições até exemplos unidimensionais e bidimensionais, como o Jogo da Vida.

Os **autômatos celulares (AC)**, estudados tanto na computação quanto na matemática pura, são sistemas dinâmicos discretos com aplicações em áreas como dinâmica de populações, dinâmica de fluidos e criptografia (GOMES; MEYER; BARROS, 2021). Há uma crescente necessidade de ferramentas capazes de modelar sistemas biológicos complexos, como mutações, replicações e a relação entre genótipo e fenótipo (CARDOSO, 2023).

Daí, surge a necessidade de desenvolver outros projetos voltados para a aplicação dos autômatos celulares para a Biologia, em especial nesse contexto, entre outros objetivos, simular a evolução da vida (MELOTTI, 2009).

Os autômatos celulares apresentam potencial para solucionar problemas em várias áreas, mais do que isso, contribui para avanços na biomatemática, na bioinformática e na pesquisa científica. Além disso, autômatos celulares (ACs) são modelos matemático-computacionais, considerados sistemas dinâmicos discretos, os quais consistem em uma malha de células, cada uma podendo estar em um número finito de estados, onde o estado de cada célula no tempo $t+1$ é determinado por uma **regra local** que depende dos estados de sua **vizinhança** no tempo t .

Na aplicação dessa teoria à biologia podemos interpretar que o genótipo do autômato celular - conjunto de genes que um organismo possui em seu DNA, como elemento que determina suas regras de transição. Em AC, a regra de transição (genótipo) determina o comportamento global emergente (fenótipo), ou seja, o genótipo influencia o fenótipo, relação a qual foi observada na pesquisa realizada por (CARDOSO, 2023).

Por consequência, nosso objetivo está direcionado na modelagem da relação genótipo-fenótipo e na análise da estabilidade do fenótipo frente a mutações no genótipo. Para alcançar as metas, utilizamos o ferramental matemático dos sistemas dinâmicos discretos e implementações computacionais em Python para simular e analisar esses processos.

Neste trabalho usando a ideia de estabilidade estrutural em sistemas dinâmicos,

definimos conceito de estabilidade e instabilidade para Autômatos Celulares Elementares (ACE) e a partir disso abordamos 3 fases de simulações. Na **primeira fase** construímos e trabalhamos sobre as perturbações nas regras de Wolfram e regras de Autômatos Celulares Áureos. Os resultados principais da 1ª fase de estudos seguem abaixo.

Estabilidade de Autômatos Celulares Elementares:

- *Classe I:* 0% estáveis (nenhuma regra)
- *Classe II:* 91,6% estáveis (176 de 192 regras)
- *Classe III:* 0% estáveis
- *Classe IV:* 0% estáveis

Estabilidade de Autômatos Celulares Áureos:

- 56 regras áureas no total
- 38 regras estáveis (68%)
- Comportamento predominantemente das Classes I e II

Já na **segunda fase** simulamos, usando autômatos para representar o DNA, assim como no trabalho de (SIRAKOULIS et al., 2003), onde temos a modelagem de DNA, utilizando autômatos celulares unidimensionais, a qual associa a cada base nitrogenada um número correspondente ($A \rightarrow 0, C \rightarrow 1, T \rightarrow 2, G \rightarrow 3$). Em razão disso, consideramos o sistema de base 4, para que a regra de evolução considere as interações entre vizinhos imediatos, dessa maneira, definimos: $C_i \in \{0, 1, 2, 3\}$, e na forma matricial $A' = MA$, onde M a matriz de evolução composta por 0 e 1, A é a configuração da grade no tempo t , daí a resultante A' é uma nova grade de células considerando $t + 1$, para isso utilizou-se módulo 4.

Na **terceira fase** fizemos um estudo com ligação direta entre DNA e autômatos celulares áureos. Tratamos a ausência de um código correspondente a um aminoácido específico como um bloco proibido dentro da sequência de DNA, estabelecendo uma relação com autômatos celulares do tipo áureo. Em seguida, aplicamos uma perturbação na matriz (ou regra local), interpretada como um tratamento destinado a corrigir essa ausência. Com essa intervenção, o código anteriormente faltante pôde ser recuperado e corretamente reproduzido nas simulações que foram apresentadas na Seção 4.3.1.

2 Objetivos

2.1 Objetivo Geral

Trabalhar sobre perturbações em regras de Wolfram, partindo de um estudo de estabilidade e investigar por meio da modelagem baseada em autômatos celulares podem representar fenômenos biológicos desde relações entre fenótipo e genótipo, até mutações.

2.2 Objetivos Específicos

- Analisar e estudar os sistemas dinâmicos discretos do tipo autômatos celulares, destacando seu potencial de aplicação em problemas de biomatemática.
- Compreender conceitos fundamentais de sistemas dinâmicos, como estabilidade estrutural e classificações focado em autômatos celulares elementares.
- Explorar aspectos computacionais envolvidos na simulação de autômatos celulares, interpretando, adaptando e validando os códigos desenvolvidos em linguagem Python.
- Desenvolver conceitos de estabilidade para autômatos celulares elementares e áureos, e suas possíveis aplicações em estrutura de DNA e RNA.

3 Fundamentação Teórica

3.1 Matemática

3.1.1 Sistemas Dinâmicos - Caos de Devaney

Um sistema dinâmico é determinado por uma função que descreve a evolução, no tempo, dos pontos de um dado espaço, ou seja, descreve o movimento dos pontos desse espaço, chamado espaço de fase, ao longo do tempo.

Definição 1. (MUNKRES, 2000) *Seja X um conjunto. Uma topologia em X é uma coleção τ de subconjuntos de X satisfazendo as seguintes condições:*

1. $\emptyset \in \tau$ e $X \in \tau$;

2. A união arbitrária de elementos de τ pertence a τ ; isto é, se $\{U_\alpha\}_{\alpha \in I} \subseteq \tau$, então

$$\bigcup_{\alpha \in I} U_\alpha \in \tau;$$

3. A interseção finita de elementos de τ pertence a τ ; isto é, se $U_1, \dots, U_n \in \tau$, então

$$\bigcap_{i=1}^n U_i \in \tau.$$

Os elementos de τ são chamados de conjuntos abertos de X , já os elementos que o complementar pertencem a topologia são chamados de fechados. O par (X, τ) é chamado de espaço topológico.

Definição 2. (MUNKRES, 2000) *Sejam (X, τ_X) e (Y, τ_Y) espaços topológicos. Uma função $f : X \rightarrow Y$ é dita contínua se, e somente se, para todo aberto $V \in \tau_Y$, a imagem inversa $f^{-1}(V)$ pertence a τ_X ; isto é,*

$$f^{-1}(V) \in \tau_X \quad \text{para todo } V \in \tau_Y.$$

Equivalente e intuitivamente, f é contínua quando o preimagem de qualquer conjunto aberto de Y é um conjunto aberto em X .

Definição 3. *Sejam (X, τ_X) e (Y, τ_Y) espaços topológicos. Uma função $f : X \rightarrow Y$ é chamada um homeomorfismo se satisfaz:*

1. f é bijetiva;

2. f é contínua;
3. A função inversa $f^{-1} : Y \rightarrow X$ é contínua.

Quando existe um homeomorfismo entre X e Y , dizemos que os espaços são homeomorfos e escrevemos $X \cong Y$.

Seja (X, d) um espaço métrico. Denote por τ_d a coleção de todos os conjuntos $U \subseteq X$ tais que, para todo $x \in U$, existe $r > 0$ com

$$B_d(x, r) \subseteq U,$$

onde $B_d(x, r) = \{y \in X : d(x, y) < r\}$ é a bola aberta de centro x e raio r .

Então τ_d é uma topologia em X . Em particular, cada espaço métrico (X, d) determina um espaço topológico (X, τ_d) .

Teorema 1. (LIMA, 2009) *Todo espaço métrico é um espaço topológico.*

Definição 4. *Seja X um espaço métrico, chamado espaço de fase, um conjunto discreto de instantes de tempo $\{t_0 < t_1 < t_2 < \dots\}$ e*

$$f : X \longrightarrow X,$$

uma função contínua ou homeomorfismo que permite calcular o estado y_{n+1} no instante de tempo t_{n+1} a partir do estado y_n no instante anterior t_n por meio da equação de evolução

$$y_{n+1} = f(y_n).$$

Definição 5 (Órbita de x). *Seja $f : M \rightarrow M$ um sistema dinâmico definido no espaço métrico M . A órbita de $x \in M$, descreve a história do movimento de x , ou seja, o caminho percorrido. Dependendo de f ser bijetora, ou não, este caminho percorrido pode ser nas direções positiva ou negativa.*

$$O^+(x) = \{x, f(x), f^2(x), \dots\}; \quad (\text{Órbita positiva}).$$

$$O^-(x) = \{x, f^{-1}(x), f^{-2}(x), \dots\}; \quad (\text{Órbita negativa}).$$

Caso f seja sobrejetora e não injetora, podemos definir órbita total de f como seguinte:

$$O_T(x) = \{f^n(x) : n \in \mathbb{Z}\} \quad (\text{Órbita total}).$$

Definição 6 (Ponto Periódico). (DEVANEY, 2003) *Seja $f : X \rightarrow X$ uma função contínua, onde X é o espaço métrico. Considere um ponto $x \in X$ é periódico de período n , se:*

$$f^n(x) = x \tag{3.1}$$

e

$$f^k(x) \neq x \quad \text{para todo } 0 < k < n. \quad (3.2)$$

Caso $n = 1$ o ponto x é chamado de ponto fixo.

Definição 7. (LIMA, 2009) Seja (X, τ) um espaço topológico. Dizemos que um subconjunto $A \subseteq X$ é **denso em X** quando o fecho de A coincide com todo o espaço, isto é,

$$\overline{A} = X.$$

Equivalente a isso, todo aberto não vazio de X intersecta A , ou seja, para todo aberto $U \in \tau$ com $U \neq \emptyset$, vale que $U \cap A \neq \emptyset$.

Exemplo 1. Considere o espaço topológico (\mathbb{R}, τ_u) , onde τ_u denota a topologia usual da reta real. O conjunto dos números racionais \mathbb{Q} é denso em \mathbb{R} , pois o fecho de \mathbb{Q} é igual a \mathbb{R} :

$$\overline{\mathbb{Q}} = \mathbb{R}.$$

Definição 8 (Órbita densa). A órbita de um ponto $x \in M$ é dita densa se $\forall U \subset X$ aberto, tem-se $O(x) \cap U \neq \emptyset$.

Definição 9 (Sensibilidade à condição inicial). Um sistema dinâmico $f : M \rightarrow M$ tem sensibilidade à condição inicial com constante de sensibilidade $\delta > 0$, se $\forall x \in X$ e $\forall U(x) \subset X$ aberto que contém x , existe $y \in U(x)$ e $n \in \mathbb{N}$ tal que tem-se

$$d(f^n(x), f^n(y)) > \delta.$$

Definição 10 (Caos Devaney). Seja M um espaço métrico e $f : M \rightarrow M$ uma função contínua, então f representa um **sistema dinâmico caótico** se satisfaz as seguintes propriedades:

1. f tem sensibilidade à condição inicial.
2. f é transitiva (existe Órbita densa).
3. Densidade de pontos periódicos.

Definição 11. (DEVANEY, 2003) Sejam (X, τ_X) e (Y, τ_Y) espaços topológicos e sejam $f : X \rightarrow X$ e $g : Y \rightarrow Y$ duas aplicações contínuas. Dizemos que f e g são **topologicamente conjugadas** quando existe um homeomorfismo $h : X \rightarrow Y$ tal que comuta da seguinte forma:

$$h \circ f = g \circ h,$$

isto é, para todo $x \in X$ tem-se

$$h(f(x)) = g(h(x)).$$

Nesse caso, o homeomorfismo h é chamado de **conjugação topológica** entre f e g .

Definição 12 (Estabilidade Estrutural). (PALIS; MELO, 1982) Seja $f : M \rightarrow M$ uma aplicação contínua definida em um espaço métrico (compacto) M . Dizemos que f é **estruturalmente estável** se existir uma vizinhança \mathcal{U} de f no espaço $C^0(M, M)$ (munido da topologia uniforme (MUNKRES, 2000)) tal que, para toda $g \in \mathcal{U}$, exista um homeomorfismo $h : M \rightarrow M$ satisfazendo

$$h \circ f = g \circ h.$$

Em outras palavras, toda aplicação g suficientemente próxima de f é topologicamente conjugada a f .

Intuitivamente, a estabilidade estrutural significa que o comportamento qualitativo do sistema dinâmico se mantém inalterado sob pequenas perturbações.

3.1.2 Espaço de Sequências $\Sigma = \{0, 1\}^{\mathbb{Z}}$

Definição 13. O espaço de sequências é definido da seguinte maneira:

$$\{0, 1\}^{\mathbb{Z}} := \{x = (x_i) | x_i \in \{0, 1\}, i \in \mathbb{Z}\}.$$

Sejam $x, y \in \{0, 1\}^{\mathbb{Z}}$, ou seja, $x = (x_i)$ e $y = (y_i)$ tais que $i \in \mathbb{Z}$. Definimos a métrica da seguinte maneira:

$$N(x, y) = \begin{cases} \min\{|i| : x_i \neq y_i\}, & \text{se } x \neq y \\ +\infty, & \text{se } x = y \end{cases}, \quad d(x, y) = \left(\frac{1}{2}\right)^{N(x, y)}. \quad (3.3)$$

Observação 1. (GOMES, 2024) Note que $d : \{0, 1\}^{\mathbb{Z}} \times \{0, 1\}^{\mathbb{Z}} \rightarrow [0, 1]$ é uma métrica.

Definição 14. Seja $X = (\{0, 1\}^{\mathbb{Z}}, d)$ um espaço métrico, onde a métrica d é a definida anteriormente (3.3). A dinâmica $\sigma : \{0, 1\}^{\mathbb{Z}} \rightarrow \{0, 1\}^{\mathbb{Z}}$, onde

$$\sigma(\dots, x_{-1}; x_0, x_1, \dots) = (\dots, x_{-1}, x_0; x_1, x_2, \dots),$$

é chamado **shift bilateral** e

$$\sigma^{-1} : \{0, 1\}^{\mathbb{Z}} \rightarrow \{0, 1\}^{\mathbb{Z}},$$

onde

$$\sigma^{-1}(\dots, x_{-1}; x_0, x_1, \dots) = (\dots, x_{-2}; x_{-1}, x_0, x_1, \dots)$$

é a inversa de σ .

Proposição 1. (GOMES, 2024) Dinâmica shift bilateral é um homeomorfismo.

Definição 15. (LIND; MARCUS, 1995a) Seja \mathcal{S} um alfabeto finito e considere o espaço pleno $\mathcal{S}^{\mathbb{Z}}$, munido da topologia métrica. Um conjunto $X \subseteq \mathcal{S}^{\mathbb{Z}}$ é chamado de espaço de shift ou subshift se satisfaz:

1. X é subconjunto fechado na topologia métrica;
2. X é shift-invariante, isto é, $\sigma(X) = X$, onde σ é o operador shift definido por $(\sigma(x))_i = x_{i+1}$.

3.1.3 Autômatos Celulares na Computação (definição)

Um autômato celular (AC) unidimensional é definido por uma quintúpla (L, S, n, R, c_0) , onde:

- L é uma rede unidimensional de células (finitas ou infinitas), tipicamente $L = \mathbb{Z}$.
- S é um conjunto finito de estados.
- n é tamanho de bloco de vizinhança de cada célula.
- $f : S^n \rightarrow S$ é a regra local de transição que define o novo estado de uma célula baseado na configuração de sua vizinhança de tamanho n .
- c_0 configuração inicial:...

Seja \mathcal{C} o conjunto de todas configurações iniciais possíveis cuja cada célula pertence a S . O mapa $\tilde{R} : \mathcal{C} \rightarrow \mathcal{C}$ induzida pela regra local R é o nosso sistema dinâmico discreto chamado **Regra global** de AC.

A **vizinhança** de uma célula geralmente inclui a própria célula e r células à esquerda e à direita, totalizando $n = 2r + 1$ células. Para $r = 1$, temos uma vizinhança de 3 células. Esse modelo será a base de todo nosso trabalho.

Definição 16 (blocos). *Seja $w = w_1w_2\dots w_n$, onde w_i , com $i \in \{1, 2, \dots, n\}$, é um dos estados de S , w é o concatenado finito de estados em uma dada ordem, ou seja, justaposição de estados ordenados.*

Exemplo 2. *Considerando os blocos de comprimento dois e apenas dois estados 0 e 1, temos os seguintes blocos: 10; 01; 00; 11.*

Exemplo 3. *Considere este bloco $w = 1001$ de tamanho quatro com apenas dois estados 0 e 1*

Definição 17. *(LIND; MARCUS, 1995b) Seja F um conjunto de blocos em S^k com $k \in \mathbb{N}$. O espaço $X_F \subset \Sigma$, é definido como : $X_F = \{x \in \Sigma | x \text{ não contém nenhum bloco de } F\}$. Os blocos $C \in F$ são chamados de blocos proibidos.*

Proposição 2. *(LIND; MARCUS, 1995b) Qualquer espaço X_F , onde F é o conjunto de blocos proibidos, é um espaço shift.*

Definição 18. (LIND; MARCUS, 1995b) Um exemplo clássico de espaço de shift é definido da seguinte maneira: Seja $S = \{0, 1\}$ e $\Sigma = S^{\mathbb{Z}}$. Se $F = \{11\}$, seja o conjunto de blocos proibidos, o X_F associado é um espaço shift chamado de **Golden Shift**.

Definição 19. (LIND; MARCUS, 1995b) Seja X um espaço de shift com símbolos em S e Y um espaço de shift com símbolos em B . Um **código de bloco deslizante** é um mapa contínuo $\tilde{R} : X \rightarrow Y$ com $R : B_N(X) \rightarrow B$ um mapa de código, onde $N = 2n + 1$ e $B_N(X)$ é conjunto de blocos de tamanho N em X tal que regra global $[\tilde{R}(x)]_i = R(x_{[i-n, i+n]})$. Quando espaço $X = Y$ e $B = S$ o mapa de código R é a regra local associado e o \tilde{R} representa o sistema dinâmico discreto que chamamos de autômato celular.

Teorema 2 (Curtis–Hedlund–Lyndon). (BRIN; STUCK, 2002) Sejam \mathcal{A} e \mathcal{B} alfabetos finitos e sejam $X \subseteq \mathcal{A}^{\mathbb{Z}}$ e $Y \subseteq \mathcal{B}^{\mathbb{Z}}$ subshifts com a topologia métrica. Uma aplicação $F : X \rightarrow Y$ é um autômato celular se, e somente se, F é contínua e comuta com o shift, isto é:

$$F \circ \sigma = \sigma \circ F.$$

Além disso, se F satisfaz essas condições então existe um inteiro $r \geq 0$ (chamado raio) e uma função local $\mu : \mathcal{A}^{2r+1} \rightarrow \mathcal{B}$ tal que, para todo $x \in X$ e todo $n \in \mathbb{Z}$,

$$(F(x))_n = \mu(x_{n-r}, \dots, x_{n+r}).$$

Definição 20 (Golden AC). Seja X_F o espaço shift do tipo Golden então, um código de bloco deslizante $\tilde{R} : X_F \rightarrow X_F$ com $R : B_N(X_F) \rightarrow S$ chamamos de **Golden AC**.

3.1.3.1 Condições de Fronteira

Em redes finitas, é necessário definir condições de fronteira:

- **Fixas:** As células das extremidades mantêm um valor fixo, ou seja inalterado em função do tempo.
- **Reflexivas:** As extremidades refletem os valores das células adjacentes.
- **Periódicas:** A rede é considerada circular, conectando a última célula à primeira, dessa forma a fita, que é uma grade de células unidimensionais, é colada.

3.1.3.2 A Métrica de Hamming

A distância de Hamming (DH) pode ser usada para quantificar a diferença entre duas regras ou fitas (configurações):

$$DH = \frac{1}{L} \sum_{i=1}^L |x_i(t) - y_i(t)|$$

onde L é o número de células (ou tamanho de regras), $x_i(t)$ é o estado da célula i no tempo t na configuração 1 (ou na regra 1) e $y_i(t)$ é o estado correspondente na configuração 2 (ou na regra 2).

Exemplo 4. Considere duas fitas, x e y (ou duas regras R_1, R_2), com apenas dois estados 0 ou 1, e $L = 8$, ou seja, quantidade de células dispostas na fita, da seguinte maneira

$$x = (1, 1, 1, 0, 0, 1, 0, 1)$$

$$y = (0, 1, 1, 1, 0, 0, 0, 0)$$

Dessa forma, $DH = (\frac{1}{8}) \cdot (1 + 1 + 1 + 1) = \frac{1}{2}$

3.1.4 ACE - Autômatos Celulares Elementares

Um *autômato celular elementar* (ACE) é um autômato celular unidimensional no qual o conjunto de estados é dado por

$$S = \{0, 1\},$$

a vizinhança é

$$V = (\text{esquerda}, \text{central}, \text{direita}), \quad n = 3$$

e a dinâmica ocorre sobre o espaço de todas as configurações binárias $S^{\mathbb{Z}}$. Em cada instante t , o estado da célula $n \in \mathbb{Z}$ é atualizado segundo a regra local

$$f : S^3 \longrightarrow S,$$

isto é,

$$x_i^{t+1} = f(x_{i-1}^t, x_i^t, x_{i+1}^t), \quad i \in \mathbb{Z}.$$

Como o alfabeto é binário e a vizinhança possui três células, existem exatamente $2^{2^3} = 256$ regras locais possíveis para ACEs.

3.1.5 Classificação de Wolfram para ACE

Uma classificação intuitiva, a classificação de Wolfram para os autômatos celulares baseada em seu comportamento global, considerando fitas de tamanho $L = 31$, vizinhança de tamanho 1, olhando apenas para apenas para dois estados 0 e 1

1. **Classe I:** Evoluem para um estado homogêneo e uniforme.
2. **Classe II:** Evoluem para estruturas simples estáveis ou periódicas.
3. **Classe III:** Apresentam comportamento caótico.

4. **Classe IV:** Exibem comportamento complexo, com estruturas persistentes e potencialmente universais.

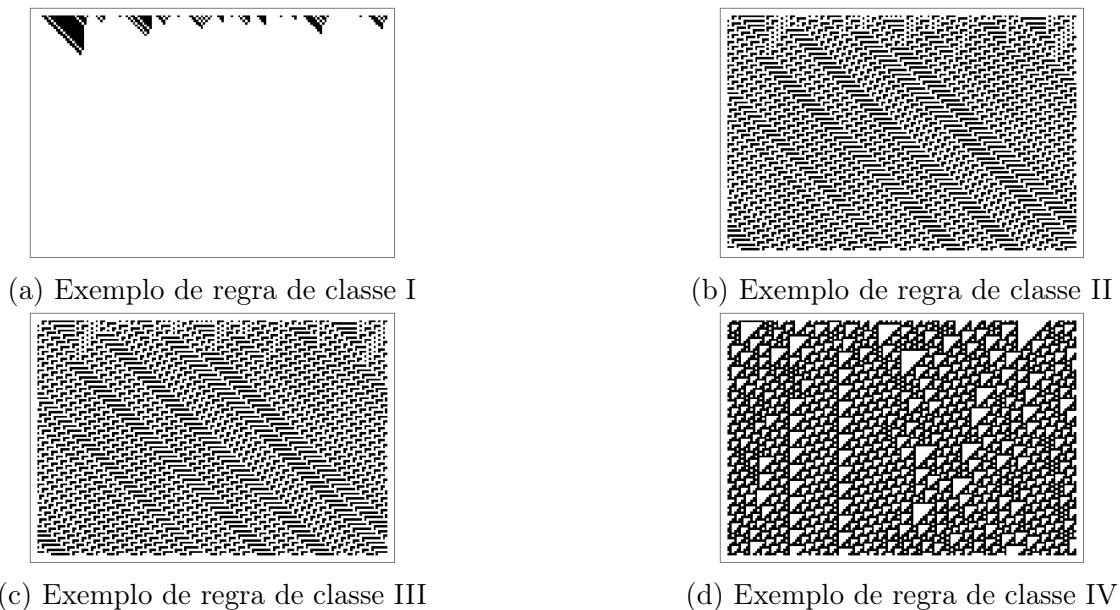


Figura 1 – Que traz as seguintes regras de Wolfram: (a) R_{224} . (b) R_9 . (c) R_{30} . (d) R_{110}
 Fonte: (MEHDIPOUR; GIBRIM; ALBUQUERQUE, 2025).

Tabela 1 – Tabela de classificação das Regras de Wolfram. Fonte: (WOLFRAM, 2002)

Classes	Regras
I	0, 8, 32, 40, 64, 96, 128, 136, 160, 168, 192, 224, 234, 235, 238, 239, 248, 249, 250, 251, 252, 253, 254, 255
II	1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 31, 33, 34, 35, 36, 37, 38, 39, 42, 43, 44, 46, 47, 48, 49, 50, 51, 52, 53, 55, 56, 57, 58, 59, 61, 62, 63, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 87, 88, 91, 92, 93, 94, 95, 98, 99, 100, 103, 104, 108, 109, 111, 112, 113, 114, 115, 116, 117, 118, 119, 123, 125, 127, 130, 131, 132, 133, 134, 138, 139, 140, 141, 142, 143, 144, 145, 148, 152, 154, 155, 156, 157, 158, 159, 162, 163, 164, 166, 167, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 184, 185, 186, 187, 188, 189, 190, 191, 194, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 226, 227, 228, 229, 230, 231, 232, 233, 236, 237, 240, 241, 242, 243, 244, 245, 246, 247
III	18, 22, 30, 45, 60, 75, 86, 89, 90, 101, 102, 105, 122, 126, 129, 135, 146, 149, 150, 151, 153, 161, 165, 182, 183, 195
IV	41, 54, 97, 106, 107, 110, 120, 121, 124, 137, 147, 169, 193, 225

Alguns autômatos mais complexos com grandes vizinhanças não apresentam propriedades que pertencem a apenas uma classe, podendo exibir características mistas (ex: Classe 3 e 4).

3.1.5.1 Parâmetro λ

A classificação de Wolfram em quatro classes abrange apenas os 256 autômatos celulares elementares, o que se torna insuficiente quando se pretende analisar autômatos mais sofisticados, com maior número de estados por célula ($n > 2$) ou vizinhanças mais amplas ($m > 3$).

Em 1986, Christopher Langton (LANGTON, 1990) propôs um método alternativo, baseado em um parâmetro contínuo, λ , que permite classificar os autômatos celulares de forma quantitativa. Por exemplo, se o número de estados possíveis for $n = 4$, é definida a célula com o estado 0 como célula "morta" e as células com o estado 1, 2 e 3 como células vivas.

O parâmetro λ representa a probabilidade de que uma determinada configuração de vizinhança (n^m possíveis) gere uma célula ativa. Seus valores variam de 0 a 1, onde $\lambda = 0$ corresponde a sistemas que rapidamente se fixam em um estado estável, e $\lambda = 1$ indica comportamentos caóticos e imprevisíveis ($\lambda \in [0, 1]$):

$$\lambda = \frac{N - n_q}{N},$$

onde N é o número total de configurações possíveis da vizinhança e n_q é o número de configurações que resultam no estado morto. Valores de λ próximos de 0 tendem a gerar comportamentos ordenados (Classes 1 e 2), enquanto valores próximos de 1 geram caos (Classe 3). Valores intermediários podem gerar complexidade (Classe 4).

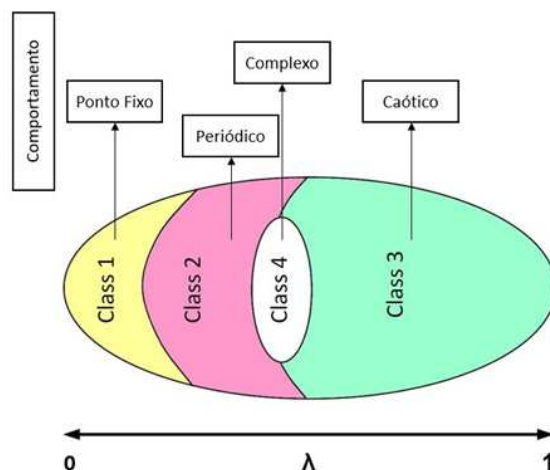


Figura 2 – Correlação dos valores de λ com as classes de Wolfram. Fonte: Adaptado de (CARDOSO, 2023).

3.1.6 Perturbações nas Regras de Wolfram

As regras de Wolfram totalizam 256 regras, cada uma delas pode ser escrita como um número de 0 a 255 e também representada por este mesmo número na base 2, por exemplo a regra 30 que está definida na tabela 2:

Tabela 2 – Tabela de Transição para Autômato Celular (Exemplo de Regra 30. Fonte: Autores)

Vizinho Esquerdo	Célula Central	Vizinho Direito	Novo Estado
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	0

Esta regra pode ser representada pela sequência binária:

00011110

que, quando convertida para decimal, resulta em:

$$00011110_2 = 0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 16 + 8 + 4 + 2 = 30$$

Embora a estabilidade estrutural clássica se aplique a mapas contínuos, adotamos uma noção análoga para autômatos celulares unidimensionais de Wolfram:

Definição 21. Dizemos que uma regra é *estruturalmente robusta* ou *estável* se pequenas perturbações em sua tabela de regras resultarem em autômatos celulares pertencentes à mesma classe de Wolfram, preservando o comportamento qualitativo. Caso contrário, chamamos *estruturalmente não-robusta* ou *inestável*.

Exemplo 5 (Perturbações da $R_{30} = 00011110$ de Wolfram). Considere a R_{30} de Wolfram, fizemos as perturbações de tamanho $1/8$ (usando menor vizinhança pelo DH). Ou seja, perturbamos uma das mini-regras (locais) por assim dizer, o que resulta em outras regras, aos quais pertencem à vizinhança de tamanho $1/8$ de R_{30} :

- **Regra 158:** $10011110_2 = 1 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 158$

- **Regra 94:** $01011110_2 = 0 \times 128 + 1 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 94$
- **Regra 62:** $00111110_2 = 0 \times 128 + 0 \times 64 + 1 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 62$
- **Regra 14:** $00001110_2 = 0 \times 128 + 0 \times 64 + 0 \times 32 + 0 \times 16 + 1 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 14$
- **Regra 22:** $00010110_2 = 0 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 0 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 22$
- **Regra 26:** $00011010_2 = 0 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 = 26$
- **Regra 28:** $00011100_2 = 0 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 0 \times 1 = 28$
- **Regra 31:** $00011111_2 = 0 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1 = 31$

No primeiro estudo deste projeto de TCC, foram realizadas perturbações aplicadas nas regras de autômatos celulares elementares (ECA), apresentadas na Tabela 1, a partir de códigos em Python, cujos resultados serão apresentados na seção de Resultados.

3.1.7 Autômatos Celulares Áureos

Nas pesquisas anteriores, vínhamos trabalhando sobre autômatos celulares, tratamos de autômatos celulares áureos, uma vez que estávamos considerando blocos proibidos. De (GOMES, 2025a) temos que, se X_F é o espaço shift do tipo golden, então chamamos de **Golden AC** um código de bloco deslizante $\phi : X_F \rightarrow X_F$ bem definido. Para isso, fixamos uma fita unidimensional de comprimento $n = 20$. As regras locais são aplicadas considerando a vizinhança composta por três células: a anterior, a atual e a posterior.

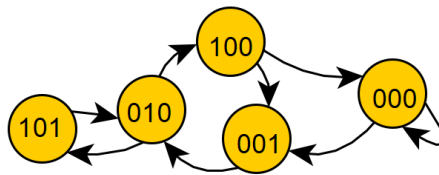


Figura 3 – Grafo construído para delimitação das regras. Fonte: (GOMES, 2025b).

Adotamos a condição de contorno periódica, na qual os extremos da fita são considerados vizinhos. Isso significa que o autômato é tratado de forma circular: a célula de índice $i = 19$ (20^{a} posição) tem como vizinha à direita a célula de índice $i = 0$, e esta, por sua vez, tem como vizinha à esquerda a célula de índice $i = 19$. Dito isso, teríamos 2^{20} possibilidades de fitas, já que temos apenas dois estados, 0 ou 1. Para o desenvolvimento resolvemos limitar para apenas as sequências de comprimento 20 que não continham 11, daí chegamos a 15126 fitas, que é aproximadamente 1,44% do total de fitas.

Nessas 15126 fitas, aplicamos 7 regras em cada uma delas, as quais foram definidas a partir do **Grafo** a seguir de maneira que não haja o bloco 11. Dessa forma, por meio de

um código em **Python**, foi possível identificar que todas as 6 regras aplicadas às 15126 fitas tinham um comportamento homogêneo ou periódico.

Tais regras são exibidas na Tabela 3.

Padrão (Esq, Centro, Dir)	R'_1	R'_2	R'_3	R'_4	R'_5	R'_6	R'_7
000	0	0	0	0	0	0	0
010	0	1	0	0	0	0	0
001	0	0	1	0	0	0	1
100	0	0	0	1	0	1	0
101	0	0	0	0	1	1	1

Tabela 3 – Transições das Regras R'_1 a R'_7 para alguns padrões de 3 células. (GOMES, 2025b)

Estendendo tais regras para comparar com Wolfram, chegamos a 48 regras, que, quando aplicadas àquelas fitas, fazem com que se obtenham classificações do tipo I ou II. Cada uma das tabelas abaixo corresponde a uma **regra associada**, que foram seis ao todo, as quais predefinimos na construção do código. Daí, estendemos essas regras para comparar com a classificação de Wolfram, totalizando 48 regras, pois, a cada regra do código, foi possível associar 8 de Wolfram.

No segundo estudo deste projeto de TCC, trabalhamos com regras locais de autômatos celulares representadas na forma de matrizes, as quais foram aplicadas a sequências de DNA de tamanho fixo associadas a genes. A ausência de um código correspondente a um aminoácido específico foi tratada como um bloco proibido dentro dessa sequência.

3.2 Biologia

3.2.1 Genótipo e Fenótipo em AC

A analogia com a biologia é vasta, a princípio focamos no trabalho de (CARDOSO, 2023), tratando especificamente de genótipo e fenótipo.

- **Genótipo:** O genótipo de um AC corresponde à **regra** que define a evolução do autômato, ou seja, como o estado de cada célula muda de acordo com os estados das células vizinhas.
- **Fenótipo:** O **padrão global** emergente do autômato ao longo do tempo a partir de uma condição inicial.

Mutações podem ser simuladas alterando a regra f , e a estabilidade fenotípica pode ser medida usando métricas como a distância de Hamming entre configurações resultantes de genótipos ligeiramente diferentes.

As **mutações genéticas** correspondem a alterações permanentes na sequência de DNA ou RNA de um organismo. De forma análoga, uma pequena modificação em uma *regra local* de um autômato celular pode alterar significativamente o comportamento emergente do sistema (fenótipo). Assim como mutações genéticas podem afetar a estrutura e a função das proteínas, causando diversas doenças, mudanças pontuais nas regras podem levar a padrões estáveis, periódicos ou caóticos nos autômatos.

Entre exemplos de mutações genéticas conhecidas estão a *anemia falciforme*, a *doença de Huntington* e o *albinismo*, que ilustram como pequenas alterações genéticas podem ter efeitos marcantes no organismo, tais mutações serão pontuadas a seguir:

1. A **anemia falciforme** (SILVA; RAMALHO, 2003) resulta de uma mutação pontual no gene responsável pela síntese da hemoglobina, levando à formação de uma variante anormal (HbS). Essa alteração modifica o formato das hemácias, que adquirem aspecto de foice, comprometendo o transporte eficiente de oxigênio.
2. A **doença de Huntington** (MOHAPEL; REGO, 2011), por sua vez, é causada por uma expansão anormal de repetições do trinucleotídeo CAG no gene HTT, o que leva à produção de uma proteína defeituosa. Essa proteína acumula-se nos neurônios, provocando degeneração progressiva do sistema nervoso central.
3. Já o **albinismo** é consequência de mutações em genes envolvidos na síntese de melanina, como o TYR, responsável pela produção da enzima tirosinase. Essas alterações reduzem ou impedem a formação do pigmento que determina a coloração da pele, dos cabelos e dos olhos.

Desse modo, observa-se que mesmo pequenas alterações na sequência gênica podem gerar consequências significativas, afetando diretamente a expressão e a função das proteínas e, portanto, o fenótipo do indivíduo.

3.2.2 Modelagem de Processos

1. Ontogénéticos (XIAO; WANG; CHOU, 2011) Os processos ontogenéticos dizem respeito a evolução e mudanças contínuas e observáveis sobre um único organismo, no caso temos o exemplo do processo embriónico.
2. Filogenético Processo Filogenéticos tratam sobre a evolução de uma dada população que interage com um ambiente, e os organismos da população interagem entre si, e a evolução é tanto vista nos organismos individuais como nas características globais da população.

3.2.3 Dinâmica de Populações e Evolução

Na escala filogenética, ACs podem modelar a evolução de populações de organismos que interagem entre si e com o ambiente. O **Jogo da Vida de Conway** é um exemplo clássico que demonstra como regras simples podem gerar comportamentos complexos remanescentes de nascimento, morte e sobrevivência em populações biológicas. (GOMES, 2025a)

O **Jogo da Vida** é um autômato celular famoso desenvolvido na década de 70 por *John Conway* é um modelo em que cada célula tem apenas dois estados possíveis; viva ou morta, no caso tal jogo é composto por infinitas células. Para determinar o estado de uma célula na próxima geração devemos olhar para ela e sua redondeza, ou seja, sua vizinhança, que no caso a vizinhança de uma célula são outras oito formando um quadrado em volta da célula estudada. Mesmo tendo uma regra simples que será mostrada logo a seguir, a partir disso, temos comportamentos complexos e até caóticos. A regra local R é definida da seguinte forma:

- **Sobrevivência:** Se uma célula viva tem dois ou três vizinhos vivos, então tal célula permanece viva para a próxima geração.
- **Morte por Superlotação:** Se uma célula viva tem mais de três vizinhos vivos, daí ela morre.
- **Morte por Solidão:** Se uma célula viva tem menos de dois vizinhos vivos, daí ela morre.
- **Nascimento:** Se uma célula morta tem exatamente 3 vizinhos vivos, daí ela nasce.

Considere C_i células tal que $i \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, e seja as células em vermelho vivas e as demais mortas, tome a geração 0 da seguinte maneira:

C1	C2	C3
C4	C5	C6
C7	C8	C9

Dessa forma, a geração 1 fica da seguinte maneira usando a regra 3.2.3.

C1	C2	C3
C4	C5	C6
C7	C8	C9

Perceba que C_8 nasceu já que possuía exatamente três vizinhos vivos, já C_5, C_6, C_9 permaneceram vivas uma vez que cada uma delas tinha dois vizinhos vivos na geração 0 e já as demais permaneceram mortas.

Ao se observar autômatos celulares com mesmo genótipo, podem gerar autômatos com fenótipos distintos, faremos uma análise na seção 4.1 sobre isso, utilizando códigos em Python, que tratará mais especificamente de estabilidade.

3.2.4 Estabilidade Fenotípica e Mutação

A estabilidade do fenótipo frente a mutações no genótipo é um conceito central na biologia evolutiva. Em ACs, podemos estudar isso submetendo um autômato a operações genéticas como mutação e recombinação, e observando como o fenótipo (padrão global) se altera.

A distância de Hamming (DH) pode ser usada para quantificar a diferença entre dois fenótipos:

$$DH = \frac{1}{L} \sum_{i=1}^L |x_i(t) - y_i(t)|$$

onde L é o número de células, $x_i(t)$ é o estado da célula i no tempo t no autômato original, e $y_i(t)$ é o estado correspondente no autômato mutante.

3.2.5 DNA e RNA

Sabemos que os **nucleotídeos** são unidades básicas do DNA e RNA, eles são compostos por: um açúcar (desoxirribose ou ribose); um grupo fosfato; uma **base nitrogenada**.

Em 1953, os cientistas James Watson e Francis Crick descreveram, pela primeira vez, a estrutura tridimensional do DNA, reunindo e integrando informações obtidas por diversos pesquisadores e inspirando-se na metodologia de Linus Pauling, que havia utilizado modelos físicos para representar a estrutura de proteínas. (RIBEIRO, 2014)

O modelo proposto por Watson e Crick destacou-se pela simplicidade e elegância, ao mesmo tempo em que revelou aspectos fundamentais do funcionamento e da replicação da molécula. De acordo com essa proposta, o DNA é constituído por unidades estruturais chamadas nucleotídeos, que se organizam em longas cadeias. Cada nucleotídeo é formado por uma base nitrogenada (adenina, timina, citosina ou guanina), um açúcar pentose (a desoxirribose) e um grupo fosfato.

A elevada densidade da molécula sugeria a existência de mais de uma cadeia, hipótese confirmada pelo modelo estrutural. Assim, demonstrou-se que o DNA é formado por duas **fitas complementares**, unidas por pontes de hidrogênio entre as bases nitrogenadas, adenina pareando com timina (A–T) e citosina com guanina (C–G). As duas cadeias apresentam orientações opostas.

Na figura 4, a seguir, as duas fitas simbolizam as cadeias de fosfato e açúcar, enquanto as hastes horizontais representam os pares de bases que mantêm as cadeias unidas. A linha vertical indica o eixo da fibra.

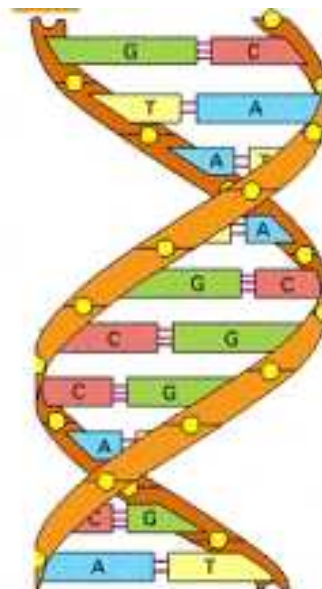


Figura 4 – Representação diagramática do modelo de dupla hélice do DNA explicitando as bases nitrogenadas. Fonte: Adaptada de *Site - Só Biologia*

No DNA, temos quatro bases nitrogenadas:

- A - Adenina
- T - Timina

- C - Citosina
- G - Guanina

Já no RNA, a única alteração é que a Timina é substituída pela uracila (U). Uma sequência de três nucleotídeos é codifica um aminoácido. Sendo assim o DNA composto por duas fitas de sequência de bases passam pelo processo de **transcrição**, ou seja, uma fita é lida apenas, conhecida como fita molde e a partir dela cria uma fita complementar substituindo Timina por Uracila levando no RNA-mensageiro leva a informação até os ribossomos que funcionam como "fabricas" de proteínas, nesse momento o RNA é lido e cada códon (trinca de nucleotídeos) que correspondem a um aminoácido específico, assim passando pela fase da **tradução** e leva a uma dada proteína.

Nos debruçamos sobre variados trabalhos com diferentes abordagens na previsão de mutações, dentre elas temos:

(MIZAS et al., 2008)

- Modelos estatísticos e probabilísticos, como modelos de substituição (Markov e modelos evolutivos) que estimam taxas de substituição de nucleotídeos/aminoácidos com base em sequências alinhadas e suas histórias evolutivas.
- Modelos baseados em contexto (neighbour dependent), nesse caso, as mutações podem depender das bases vizinhas, daí tal modelagem incorpora o contexto local, dessa maneira levando a probabilidades condicionais.
- Modelos de aprendizado de máquina (Supervisionado), utilizando de redes neurais e treinos a partir de base de dados conhecendo os efeitos, no caso biológicos de tal estrutura provocada pela mutação. Uma desvantagem desse método é que exigem um conjunto vasto de dados rotulados.
- Abordagem híbrida e simulativa foca na simulação do processo usando regras determinísticas e calibram os parâmetros com dados reais.

Autômatos celulares têm sido aplicados em bioinformática, por exemplo, para prever estruturas de proteínas e analisar sequências de DNA (XIAO; WANG; CHOU, 2011). Nesses modelos, os aminoácidos ou nucleotídeos são representados como células, e as regras de transição capturam iterações locais que influenciam a estrutura tridimensional ou função biológica.

4 Resultados e Discussões

Ao realizarmos um estudo sobre perturbações nas regras de Wolfram com o objetivo de analisar a estabilidade dessas regras, considerando que as regras de Wolfram variam de 0 a 255 e são representadas por oito dígitos binários (0 ou 1), na base 2, e foram realizadas perturbações de tamanho $1/8$, isto é, modificações em apenas um desses dígitos. Desta forma, escrevemos um código em linguagem de programação Python, apresentado no Apêndice A.

4.1 Primeira Fase

A partir das perturbações, analisamos o comportamento das regras alteradas com base na classificação de (WOLFRAM, 2002). Por meio da definição de uma regra como estável quando, após a perturbação de tamanho $1/8$ que gera oito novas regras possíveis, mais da metade dessas regras (ou seja, pelo menos cinco - 62.5%) pertencem à mesma classe da regra original. Caso contrário, a regra foi classificada como não estável. Os resultados que obtivemos estão disponíveis em: [Tabela de Classificação das Regras de Wolfram e Perturbações \(TCRWP\)](#), devido ao tamanho do arquivo, não foi possível colocar na íntegra. Podemos observar na tabela que todas as regras que são estáveis seguindo a nossa definição adotada, são regras de classe II. 21

Teorema 3. *A classe I não tem regra estável.*

Demonstração 1. *Conforme é visível na tabela TCRWP, o estudo relacionado às regra da primeira classe, que estão em vermelho, mostra que todas as perturbações não pertencem classe I, daí todas as regras de tal classe foram consideradas instáveis.*

Teorema 4. *A classe II possui 176 regras estáveis das 192 regras existentes na classe II de Wolfram, que corresponde a 91,6% das regras da classe II.*

Demonstração 2. *Conforme é visível na tabela TCRWP, o estudo relacionados de regra da segunda classe cuja estão em amarelo, mostra que 176 das perturbações de total 192 regras, pertencem a classe II, daí 176 regras de tal classe foram consideradas estáveis.*

Teorema 5. *A classe III não tem regra estável.*

Demonstração 3. *Conforme é visível na tabela TCRWP, o estudo relacionados das regras da classe III que estão em azul, mostra que todas as perturbações não pertencem a classe III, daí todas as regras de tal classe foram consideradas instáveis.*

Teorema 6. *A classe IV não tem regra estável.*

Demonstração 4. *Conforme é visível na tabela TCRWP, o estudo relacionados das regras da quarta classe que estão em verde, mostra que todas as perturbações não pertencem a classe IV, daí todas as regras de tal classe foram consideradas inestáveis.*

Corolário 1. *As regras de **Autômatos Celulares Áureos** são 56 regras ao todo, em que 38 dessas são estáveis, ou seja aproximadamente 68%*

Regra	Perturbações
R'_1	{0, 8, 64, 72, 128, 136, 192, 200}
R'_2	{4, 12, 68, 76, 132, 140, 196, 204}
R'_3	{2, 10, 66, 74, 130, 138, 194, 202}
R'_4	{16, 24, 80, 88, 144, 152, 208, 216}
R'_5	{32, 40, 96, 104, 160, 168, 224, 232}
R'_6	{48, 56, 112, 120, 176, 184, 240, 248}
R'_7	{34, 42, 98, 106, 162, 170, 226, 234}

Tabela 4 – Extensões das regras R'_1 a R'_7 de Autômatos Celulares Áureos para todas as 8 combinações possíveis dentro das regras de Wolfram. (GOMES, 2025b)

4.2 Segunda Fase

Posteriormente, fizemos outra simulação, usando autômatos para representar o DNA, visto que os autômatos têm sido usados para idealização de modelos físicos. Além disso, vemos que na biologia não é diferente, conforme apresentado no trabalho de (SIRAKOULIS et al., 2003), onde temos a modelagem de DNA, utilizando autômatos celulares unidimensionais, a qual associa a cada base nitrogenada um número correspondente ($A \rightarrow 0, C \rightarrow 1, T \rightarrow 2, G \rightarrow 3$). Em razão disso, consideramos o sistema de base 4, para que a regra de evolução considere as interações entre vizinhos imediatos, dessa maneira, definimos: $C_i \in \{0, 1, 2, 3\}$, e na forma matricial $A' = MA$, onde M a matriz de evolução composta por 0 e 1, A é a configuração da grade no tempo t , daí a resultante A' é uma nova grade de células considerando $t + 1$, para isso utilizou-se módulo 4.

$$\begin{bmatrix} \vdots \\ c_{i-1}^{t+1} \\ c_i^{t+1} \\ c_{i+1}^{t+1} \\ \vdots \end{bmatrix} = M \begin{bmatrix} \vdots \\ c_{i-1}^t \\ c_i^t \\ c_{i+1}^t \\ \vdots \end{bmatrix}$$

Vale destacar que, embora o DNA possua duas fitas complementares, considerou-se apenas uma delas, pois a outra é totalmente determinada pela complementaridade das bases.

A partir daí, construímos o código para realizar a evolução no tempo, simulamos as gerações e a alterações no DNA, o qual o código está disponível com mais detalhes no Apêndice B. A seguir são apresentados alguns resultados dessas simulações.

4.2.0.1 Passo I

Foram consideradas três estruturas distintas, cada uma representando uma fita de comprimento $L = 20$, cujas células podem assumir um dos quatro estados possíveis, associados às bases nitrogenadas. As simulações foram realizadas com sete regras diferentes, gerando ao todo 21 imagens.

A primeira regra, seguindo (SIRAKOULIS et al., 2003), tem como regra a seguinte matriz M_1 :

```
matriz_regra = [
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
]
```

Figura 5 – Representação matricial da regra de transição do autômato celular. Fonte: Autor

As estruturas escolhidas, ou seja, as fitas são: TAAGTAGGTACATCAAGGCG; TGACAGTAAATCATCTGTGG e AGTAGTGATCTGTGTGTTGATA, em que foram aplicadas a regra em cada fita respectivamente, gerando as seguintes imagens:

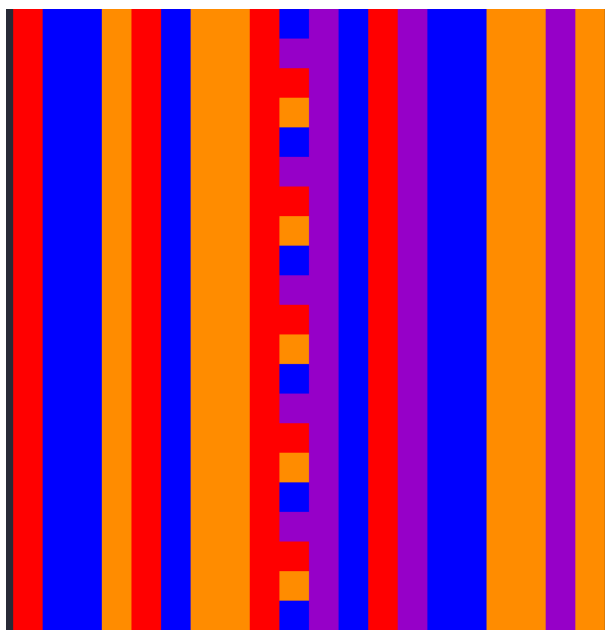


Figura 6 – 20 gerações considerando a Configuração Inicial:
 20032033201021003313 - TAAGTAGGTACATCAAGGCG. Fonte: Autor

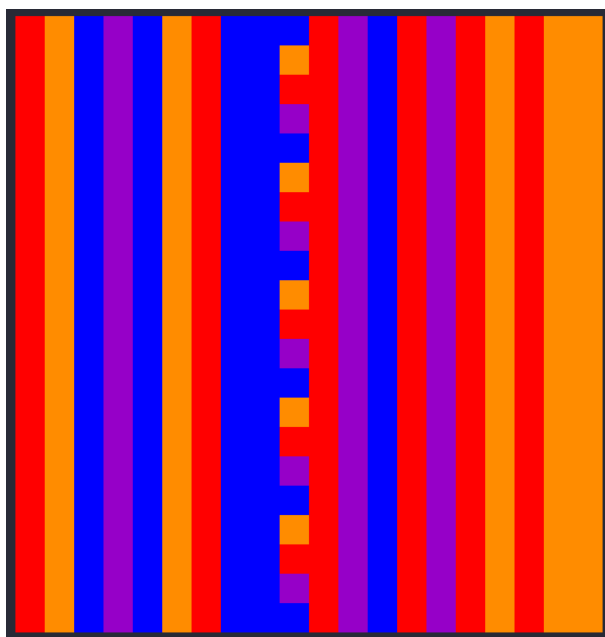


Figura 7 – 20 gerações considerando a Configuração Inicial:
 23010320002102123233 - TGACAGTAAATCATCTGTGG. Fonte: Autor

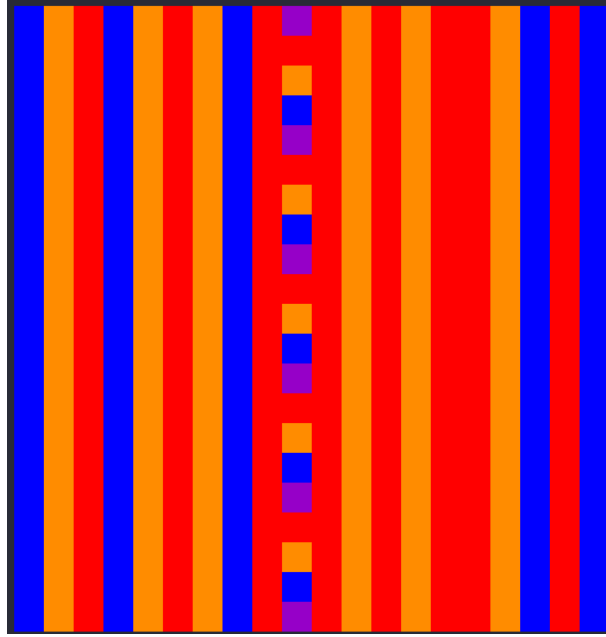


Figura 8 – 20 gerações considerando a Configuração Inicial:
AGTAGTGATCTGTGTTGATA - 03203230212323223020. Fonte: Autor

A segunda regra, seguindo (SIRAKOULIS et al., 2003), tem como regra a seguinte matriz M_2 :

```
matriz_regra = [
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
]
```

Figura 9 – Representação matricial da regra de transição do autômato celular. Fonte: Autor

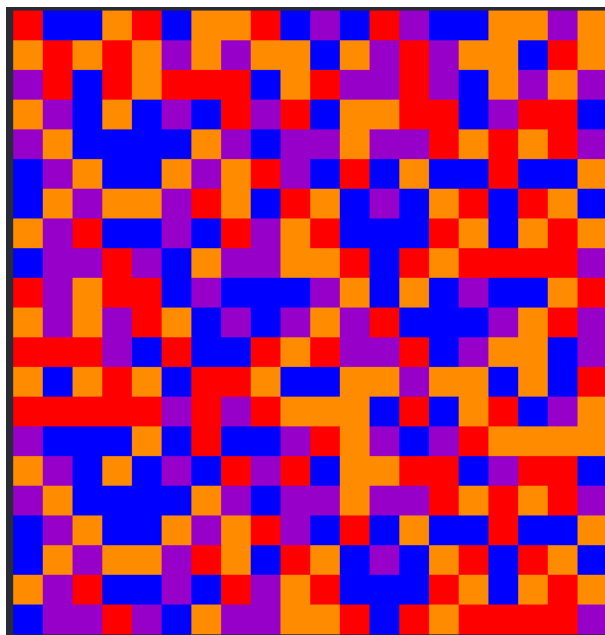


Figura 10 – 20 gerações considerando a Configuração Inicial:
20032033201021003313 - TAAGTAGGTACATCAAGGCG. Fonte: Autor

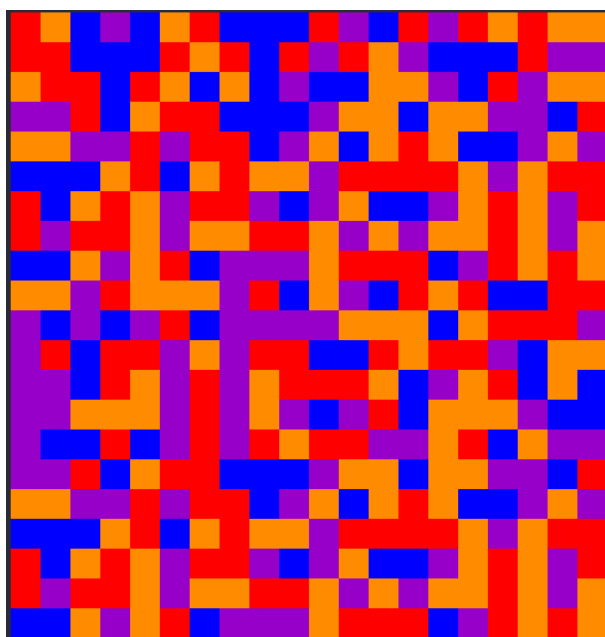


Figura 11 – 20 gerações considerando a Configuração Inicial:
23010320002102123233 - TGACAGTAAATCATCTGTGG. Fonte: Autor

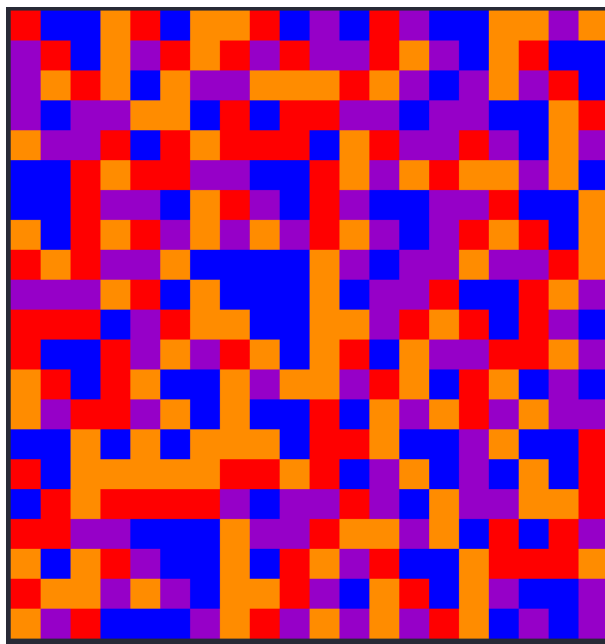


Figura 14 – 20 gerações considerando a Configuração Inicial:
20032033201021003313 - TAAGTAGGTACATCAAGGCG. Fonte: Autor

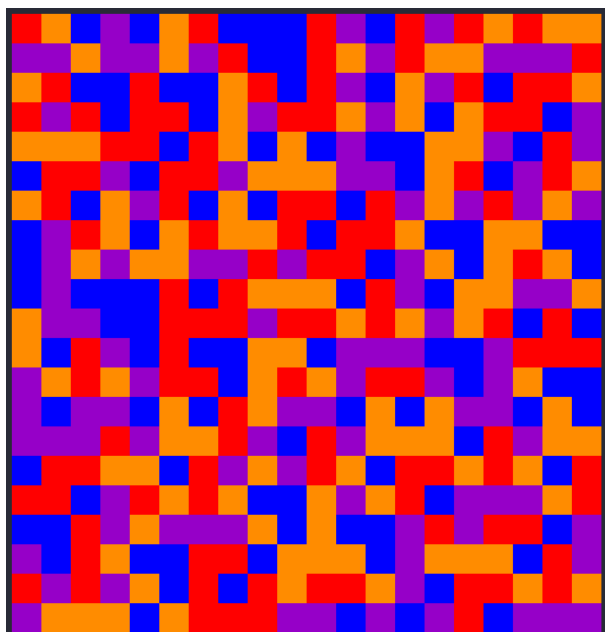


Figura 15 – 20 gerações considerando a Configuração Inicial:
23010320002102123233 - TGACAGTAAATCATCTGTGG. Fonte: Autor

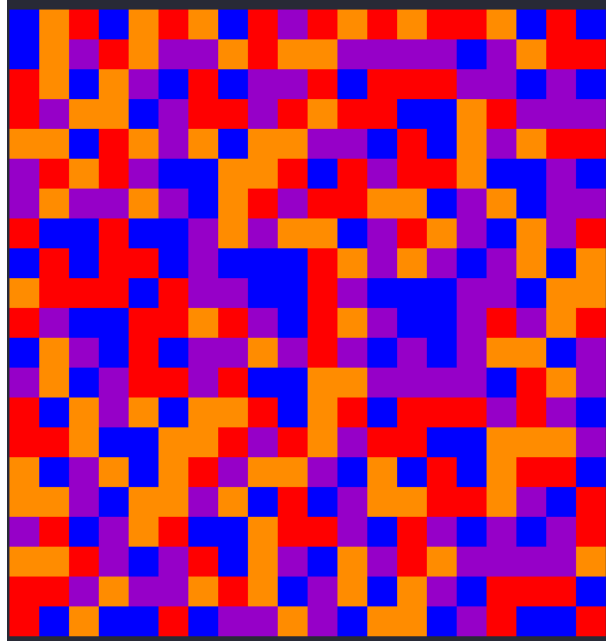


Figura 16 – 20 gerações considerando a Configuração Inicial:
AGTAGTGATCTGTGTTGATA - 03203230212323223020. Fonte: Autor

A sexta regra, seguindo (SIRAKOULIS et al., 2003), tem como regra a seguinte matriz M_6 :

```
matriz_regra = [
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
]
```

Figura 17 – Representação matricial da regra de transição do autômato celular. Fonte: Autor

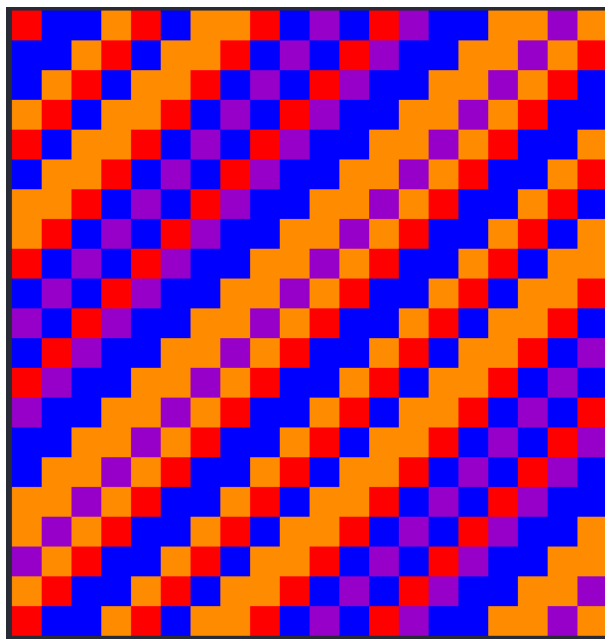


Figura 18 – 20 gerações considerando a Configuração Inicial:
20032033201021003313 - TAAGTAGGTACATCAAGGCG. Fonte: Autor

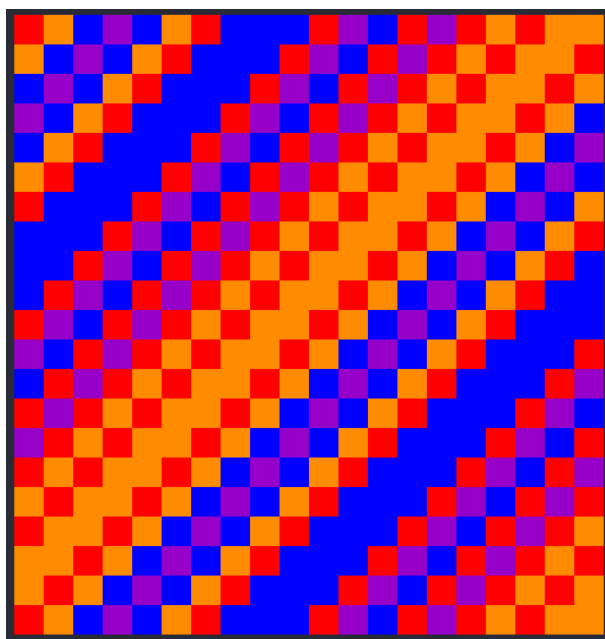


Figura 19 – 20 gerações considerando a Configuração Inicial:
23010320002102123233 - TGACAGTAAATCATCTGTGG. Fonte: Autor

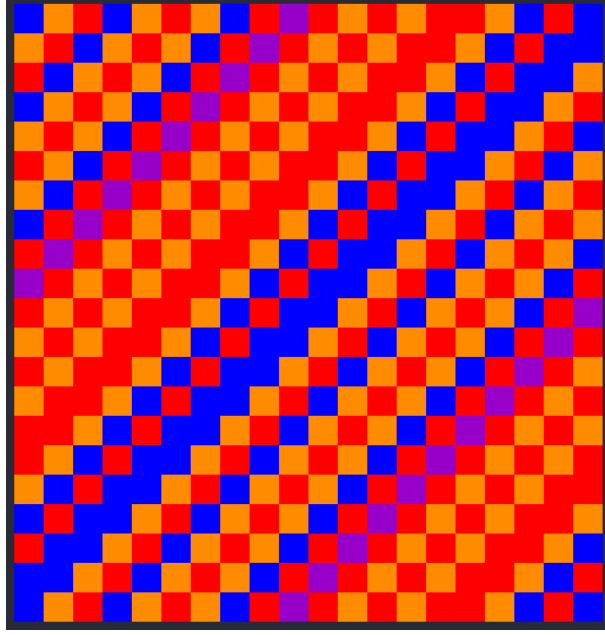


Figura 20 – 20 gerações considerando a Configuração Inicial:
AGTAGTGATCTGTGTTGATA - 03203230212323223020. Fonte: Autor

Observe que assim como anteriormente trabalhado, usamos nessas simulações regras como: M_i com $i \in \{1, 2, 3, 4, 5, 6, 7\}$, considerando que apenas as vizinhanças a anterior, atual e posterior influenciam nas próximas interações, alterando assim o central, assim como em (GOMES, 2025a).

4.2.1 Passo II

Nesse passo, apenas retomando o que já vínhamos trabalhando como a regra funciona alterando a estrutura do DNA, a mudança de uma regra para outra chamaremos de mutação, classificando assim:

- a regra $M_3 = M'_2$ uma perturbação da M_2 ;
- a regra $M_6 = M'_4$ uma perturbação da regra M_4 ;
- a regra $M_7 = M'_5$ uma perturbação da regra M_5 .

Todas essas perturbações citadas acima têm $DH = \frac{1}{20}$, ou seja, $DH(M_2, M_3) = DH(M_4, M_6) = DH(M_5, M_7) = \frac{1}{20}$.

A terceira regra, seguindo (SIRAKOULIS et al., 2003), tem como regra a seguinte matriz M_3 :

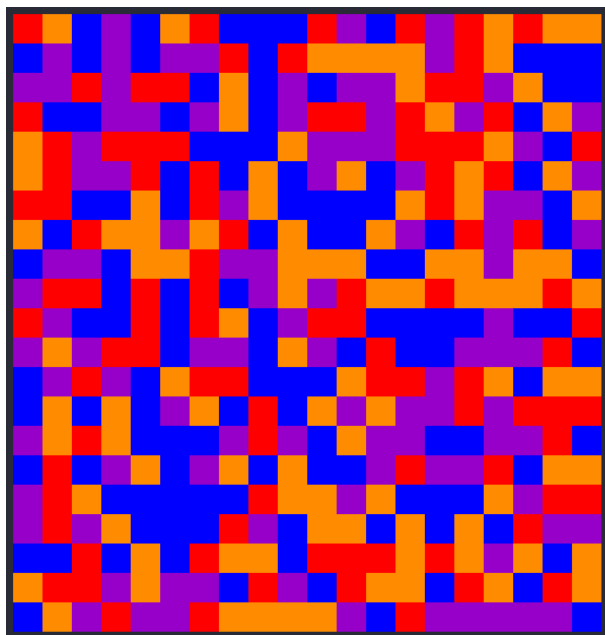


Figura 23 – 20 gerações considerando a Configuração Inicial:
 23010320002102123233 - TGACAGTAAATCATCTGTGG. Fonte: Autor

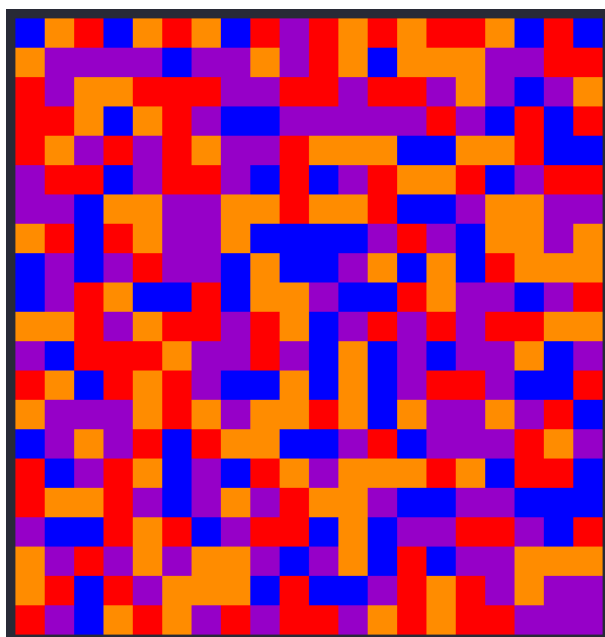


Figura 24 – 20 gerações considerando a Configuração Inicial:
 AGTAGTGATCTGTGTTGATA - 03203230212323223020. Fonte: Autor

A quarta regra, seguindo (SIRAKOULIS et al., 2003), tem como regra a seguinte matriz M_4 :

```

matriz_regra = [
[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
]

```

Figura 25 – Representação matricial da regra de transição do autômato celular. Fonte: Autor

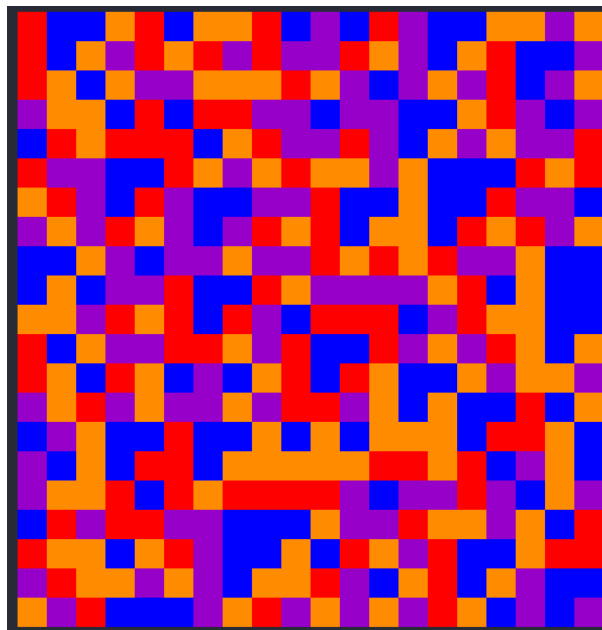


Figura 26 – 20 gerações considerando a Configuração Inicial: 20032033201021003313 - TAAGTAGGTACATCAAGGCG. Fonte: Autor

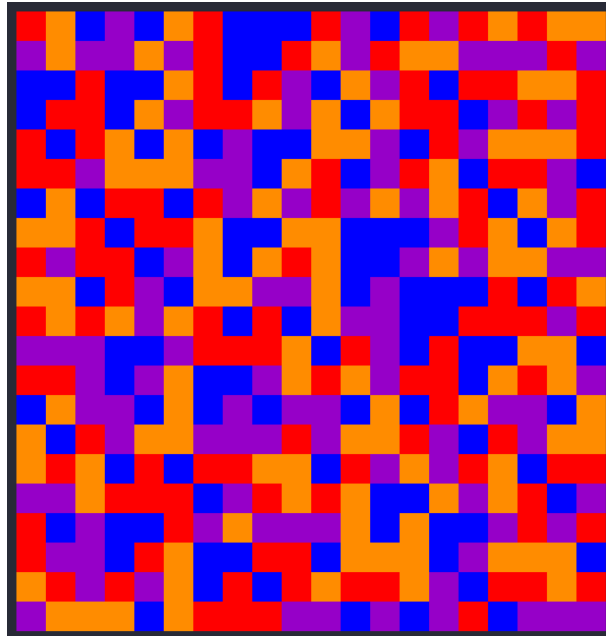


Figura 27 – 20 gerações considerando a Configuração Inicial:
 23010320002102123233 - TGACAGTAAATCATCTGTGG. Fonte: Autor

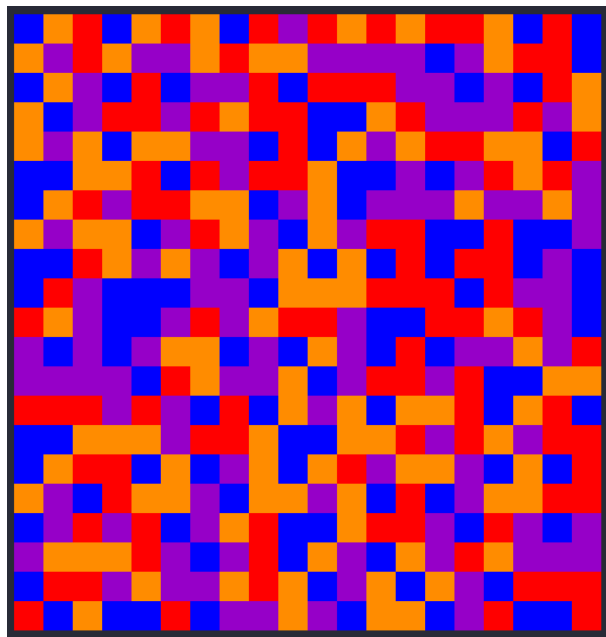


Figura 28 – 20 gerações considerando a Configuração Inicial:
 AGTAGTGATCTGTGTTGATA - 03203230212323223020. Fonte: Autor

A sétima regra, seguindo (SIRAKOULIS et al., 2003), tem como regra a seguinte matriz M_7 :

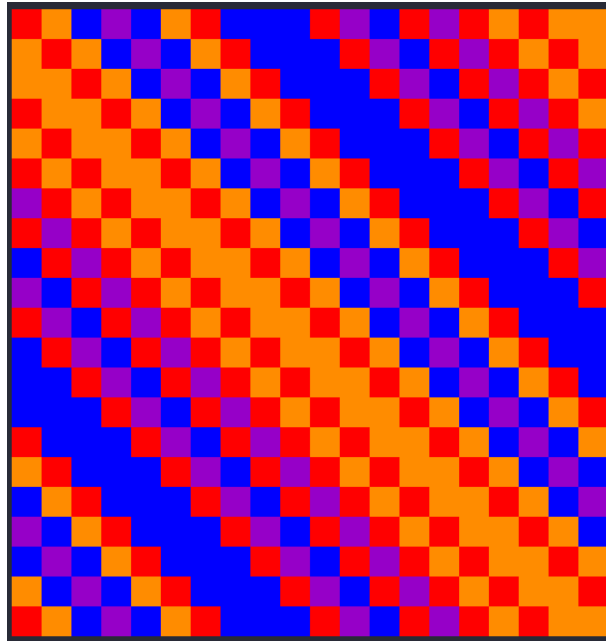


Figura 31 – 20 gerações considerando a Configuração Inicial:
 23010320002102123233 - TGACAGTAAATCATCTGTGG. Fonte: Autor

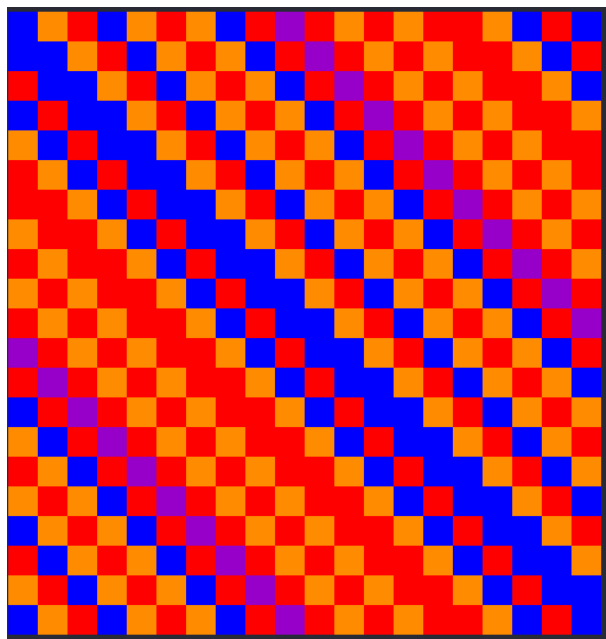
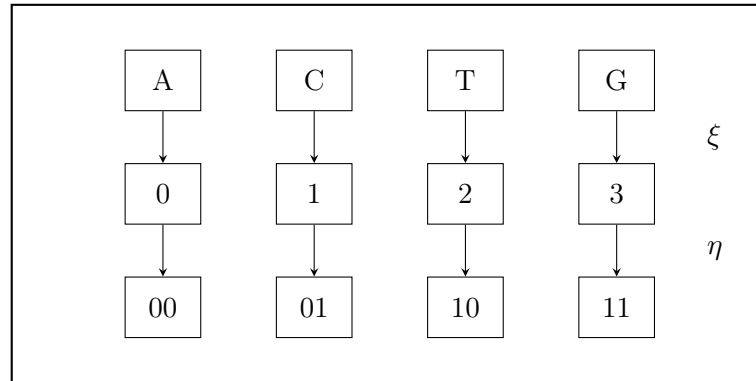


Figura 32 – 20 gerações considerando a Configuração Inicial:
 AGTAGTGATCTGTGTTGATA - 03203230212323223020. Fonte: Autor

4.3 Fase Final

Já em fase final, consideramos os seguintes mapas bijetores, ξ, η e sua composição, para daí utilizarmos Autômatos Celulares Áureos 3.1.7, dessa forma consideramos o 11 bloco proibido, ou seja, a partir da bijeção, estamos levando em conta estruturas de DNA danificado, cujo não tem a presença da Guanina(G). Considere o seguinte mapa contínuo:



Fonte: Elaborado pelos autores.

$$g := g(x) = \xi(\eta(x)).$$

$$g : \{A, C, T, G\} \longrightarrow \{00, 01, 10, 11\}$$

por

$$g(A) = 00, \quad g(C) = 01, \quad g(T) = 10, \quad g(G) = 11.$$

A aplicação g é bijetora, pois cada elemento do domínio possui imagem distinta e todo bloco binário de dois bits é atingido. Assim, sua inversa é bem definida:

$$g^{-1} : \{00, 01, 10, 11\} \longrightarrow \{A, C, T, G\}$$

com

$$g^{-1}(00) = A, \quad g^{-1}(01) = C, \quad g^{-1}(10) = T, \quad g^{-1}(11) = G.$$

Observação 2. Embora, nesta simulação, tenhamos relacionado o nucleotídeo G ao bloco proibido 11 por meio de uma bijeção, é importante ressaltar que, do ponto de vista da dinâmica simbólica e da comparação entre dinâmicas, existem restrições que impedem a conjugação completa entre eles. Pretende-se, em trabalhos futuros, desenvolver e adaptar a estrutura dos códigos de DNA de modo a permitir uma relação direta, seja por conjugação ou semi-conjugação topológica, com autômatos celulares do tipo áureo, de forma a explorar melhor os resultados de estabilidade obtidos.

4.3.1 Passo I

Na parte final do estudo, tratamos a ausência de um código correspondente a um aminoácido específico como um bloco proibido dentro da sequência de DNA, estabelecendo uma relação com autômatos celulares do tipo áureo. Em seguida, aplicamos uma perturbação na matriz (ou regra local), interpretada como um tratamento destinado a corrigir essa ausência. Com essa intervenção, o código anteriormente faltante pôde ser recuperado e corretamente reproduzido nas simulações. Como já estávamos trabalhando $L = 20$ e temos 4 estados possíveis, daí o total de configurações possíveis é $4^{20} = 1099511627776$, mas daí olhando apenas para as configurações que não possui o 3, ou seja, Guanina, temos um total de $3^{20} = 3486784401$ configurações, dessa forma, a chance de ocorrer uma fita sem 3 é $\left(\frac{3}{4}\right)^{20} \approx 0,317\%$ de todas as fitas possíveis não usam o estado proibido.

Ao acaso, retiramos três fitas sem 3, que são as seguintes: 01202210212122221020; 01120011111002221212; 11111111000002100021.

Nesse primeiro momento aplicamos em todas essas três fitas a regra associada à matriz M_7

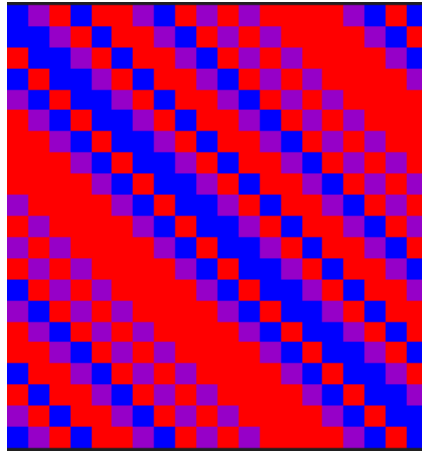


Figura 33 – 20 gerações considerando a Configuração Inicial (sem 3):

ACTATTCATCTCTTTTCATA - 01202210212122221020. Fonte: Autor

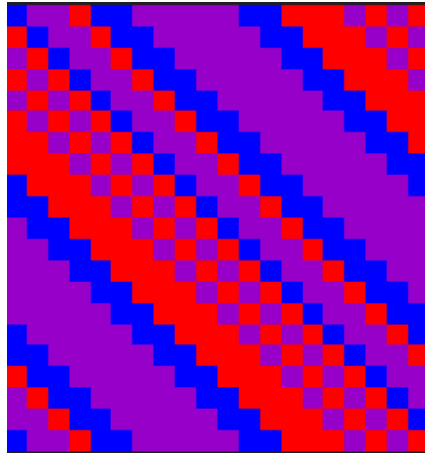


Figura 34 – 20 gerações considerando a Configuração Inicial (sem 3):
 ACCTAACCCCAATTTCTCT - 01120011111002221212. Fonte: Autor

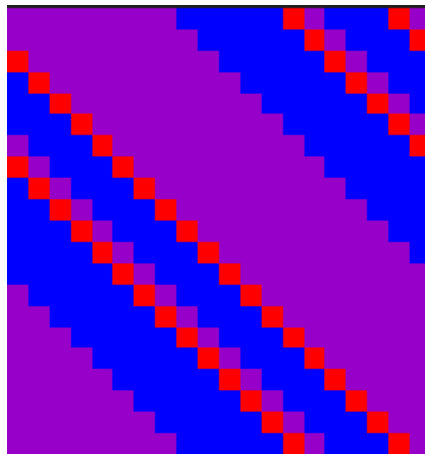


Figura 35 – 20 gerações considerando a Configuração Inicial (sem 3):
 CCCCCC AAAAATCAAATC - 11111111000002100021. Fonte: Autor

4.3.2 Passo II

Num segundo momento, aplicamos a regra M_5 como uma perturbação ou mutuação de regra M_7 de $DH(M_7, M_5) = \frac{1}{20}$ aplicadas nas três fitas anteriores, resultando nos seguintes configurações abaixo apresentadas:

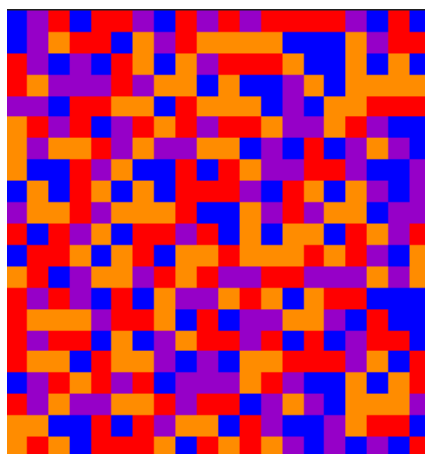


Figura 36 – 20 gerações considerando a Configuração Inicial (sem 3):
 ACTATTCATCTCTTTTCATA - 01202210212122221020. Fonte: Autor

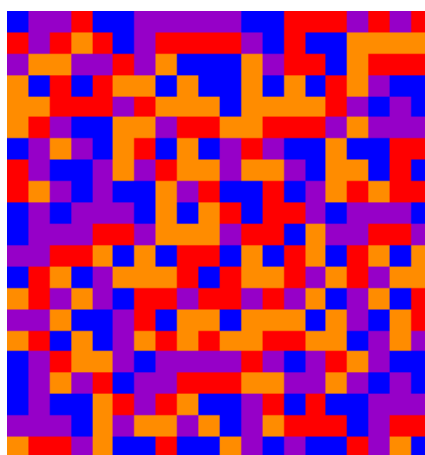


Figura 37 – 20 gerações considerando a Configuração Inicial (sem 3):
 ACCTAACCCCAATTTCTCT - 01120011111002221212. Fonte: Autor

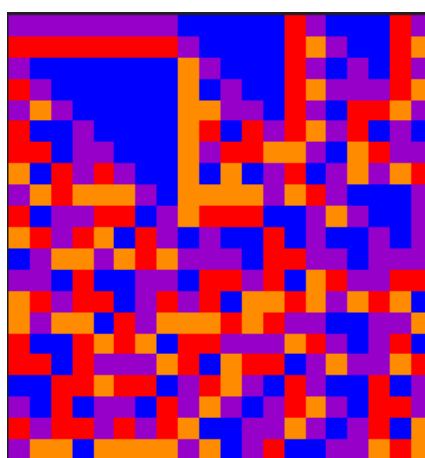


Figura 38 – 20 gerações considerando a Configuração Inicial (sem 3):
 CCCCCCAAAAATCAAATC - 11111111000002100021. Fonte: Autor

A regra M_7 só faz um deslocamento na estrutura, dessa forma independente da fita que não tem 3, vai continuar sem ter 3 independente do número de iterações. Já na regra M_5 pelas simulações é possível notar que o 3 aparece, mesmo a configuração inicial não tendo.

5 Considerações Finais

Consideramos que ainda são necessários mais estudos voltados à modelagem por meio de autômatos celulares, uma vez que essa abordagem, embora possua fundamentos estabelecidos desde a década de 1940 com John von Neumann e posteriormente popularizada na década de 1970 por John Conway 3.2.3, continua sendo relativamente recente e pouco explorada em diversas áreas. Apesar disso, observa-se um crescente interesse em sua aplicação, especialmente no campo da biologia, como demonstrado em trabalhos recentes, por exemplo, em (GOMES; MEYER; BARROS, 2021).

Embora diversos fenômenos naturais possam ser descritos por meio de expressões matemáticas, em muitos casos o método baseado em equações não é aplicável ou apresenta limitações significativas. Para lidar com sistemas de alta complexidade, recorre-se a abordagens alternativas, como a dinâmica simbólica e a teoria dos sistemas dinâmicos como descrito por (ANTONIO, 2013).

Nesse contexto, os autômatos celulares surgem como uma ferramenta promissora para representar e compreender determinados fenômenos naturais.

Referências

- ANTONIO, C. *Aplicação de Autômatos Celulares para Modelagem de Variáveis Regionalizadas na Mineração*. Dissertação (Dissertação (Mestrado em Engenharia Mineral)) — Universidade de São Paulo (USP), São Paulo, 2013.
- BRIN, M.; STUCK, G. *Introduction to Dynamical Systems*. Cambridge: Cambridge University Press, 2002. (Cambridge University Press). ISBN 9780521587501.
- CARDOSO, J. P. M. *Algoritmos genéticos para famílias de autômatos celulares em dimensão 1*. Dissertação (Mestrado) — Universidade de Évora, 2023.
- DEVANEY, R. L. *An Introduction to Chaotic Dynamical Systems*. 2nd. ed. Boulder, CO: Westview Press, 2003. Capítulo 1, Seção 1.5.
- GOMES, F. A. F.; MEYER, J. F. d. C. A.; BARROS, L. C. Simulação dos impactos de um evento super espalhador em um modelo tipo sirs por autômatos celulares epidêmicos. 2021. Trabalho acadêmico.
- GOMES, R. d. L. *Relatório de Iniciação Científica: Caos e Fractais I*. Viçosa, MG, 2024.
- GOMES, R. d. L. *Caos e Fractais II*. [S.l.], 2025.
- GOMES, R. d. L. *Relatório de Iniciação Científica: Caos e Fractais II*. Viçosa, MG, 2025.
- LANGTON, C. G. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, v. 42, n. 1-3, p. 12–37, 1990.
- LIMA, E. L. *Espaços Métricos*. 5. ed. Rio de Janeiro: Instituto de Matemática Pura e Aplicada (IMPA), 2009. (Projeto Euclides).
- LIND, D.; MARCUS, B. *An introduction to symbolic dynamics and coding*. [S.l.]: Cambridge University Press, 1995.
- LIND, D.; MARCUS, B. *An introduction to symbolic dynamics and coding*. 1st. ed.. ed. New York: Cambridge University Press, 1995. ISBN 0521551242.
- MEHDIPOUR, P.; GIBRIM, P. T. M.; ALBUQUERQUE, L. C. A. Modelagem computacional e análise de configurações periódicas de autômatos celulares na classe i de wolfram. In: *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*. Porto de Galinhas, PE: SBMAC, 2025. v. 11, n. 1, p. 010428–1–010428–7. Trabalho apresentado no XLIII CNMAC, Centro de Convenções do Armação Resort - Porto de Galinhas - PE, 2024. Disponível em: <<http://dx.doi.org/10.5540/03.2025.011.01.0428>>.
- MELOTTI, G. *Aplicação de autômatos celulares em sistemas complexos: Um estudo de caso em espalhamento de epidemias*. [S.l.]: UFMG, 2009.
- MIZAS, C. et al. Reconstruction of dna sequences using genetic algorithms and cellular automata: Towards mutation prediction *Biosystems*, v. 92, n. 1, p. 61–68, 2008.

MOHAPEL, J. M. G.; REGO, A. C. Doença de huntington: Uma revisão dos aspectos fisiopatológicos. *Revista Neurociências*, v. 19, n. 4, p. 724–734, 2011. Disponível em: <<https://periodicos.unifesp.br/index.php/neurociencias/article/view/8332>>.

MUNKRES, J. R. *Topology*. [S.l.]: Prentice Hall, 2000.

PALIS, J.; MELO, W. de. *Geometric Theory of Dynamical Systems: An Introduction*. New York: Springer-Verlag, 1982. Clássica introdução à teoria geométrica dos sistemas dinâmicos, incluindo estabilidade estrutural e bifurcações. ISBN 978-0-387-90717-8.

RIBEIRO, M. C. M. *Genética Molecular*. 1. ed. Florianópolis: Biologia/EaD/UFSC, 2014. 158 p. ISBN 978-85-61485-12-2.

SILVA, W. N.; RAMALHO, A. S. Aspectos moleculares da anemia falciforme. *Jornal Brasileiro de Patologia e Medicina Laboratorial*, v. 39, n. 1, p. 11–17, 2003. Disponível em: <<https://doi.org/10.1590/S1676-24442003000100011>>.

SIRAKOULIS, G. C. et al. A cellular automaton model for the study of DNA sequence evolution. *International Journal of Modern Physics C*, v. 14, n. 6, p. 761–771, 2003.

WOLFRAM, S. *A New Kind of Science*. [S.l.]: Wolfram Media, 2002.

XIAO, X.; WANG, P.; CHOU, K.-C. Cellular automata and its applications in protein bioinformatics. *Current Protein and Peptide Science*, v. 12, n. 6, p. 508–519, 2011.

6 Apêndice

Apêndice A - Geração de Regras e Perturbações

```
1 \section*{Tópico 1  Geração de Regras e Perturbações}
2
3 \begin{lstlisting}[language=Python, style=pythonstyle, caption={Função para gerar
↔ regras e perturbações de autômatos celulares.}]
4 def gerar_regra_e_perturbacoes(input_regra: str):
5
6     try:
7         numero_regra = int(input_regra.split('_')[1])
8     except (IndexError, ValueError):
9         raise ValueError("Use o formato 'regra_X', onde X é um número de 0 a 255.")
10
11     if not (0 <= numero_regra <= 255):
12         raise ValueError("A regra deve estar entre 0 e 255.")
13
14     def gerar_codigo_da_regra(numero):
15
16         binario = f"{numero:08b}"
17         padroes = ["111", "110", "101", "100", "011", "010", "001", "000"]
18
19         linhas = [f"def regra_{numero}(anterior, atual, sucessor):"]
20
21         for padrao, resultado in zip(padroes, binario):
22             if resultado == '1':
23                 condicao = " and ".join([f"{bit_nome} == '{bit_valor}'"
24                                         for bit_nome, bit_valor in zip(['anterior',
25                                         ↔ 'atual', 'sucessor'], padrao)])
26
27                 linhas.append(f"    if {condicao}:")
28                 linhas.append(f"        return '{resultado}'")
29
30             linhas.append("    return '0'")
31         return "\n".join(linhas)
32
33     def gerar_perturbacoes_de_regra(regra: int):
34
35         binario = f"{regra:08b}"
36         perturbacoes = []
37         for i in range(8):
38             bits = list(binario)
39             bits[i] = '1' if bits[i] == '0' else '0'
40             perturbacoes.append(int(''.join(bits), 2))
41         return perturbacoes
```

```

39
40     codigo_total = [f"# Funções geradas automaticamente para {input_regra}\n"]
41     codigo_total.append(gerar_codigo_da_regra(numero_regra))
42     codigo_total.append("")
43
44     perturbacoes = gerar_perturbacoes_de_regra(numero_regra)
45     for idx, regra_perturbada in enumerate(perturbacoes, start=1):
46         codigo_total.append(gerar_codigo_da_regra(regra_perturbada))
47         codigo_total.append("")
48
49     nome_arquivo = f"{input_regra}_e_perturbacoes.py"
50     with open(nome_arquivo, "w", encoding="utf-8") as f:
51         f.write("\n".join(codigo_total))
52
53     print(f" Arquivo '{nome_arquivo}' gerado com {len(perturbacoes)+1} funções
54     ↪ (regra base + perturbações).")
\end{lstlisting}

```

Apêndice B - Geração de Imagens Coloridas

```
1 \begin{lstlisting}[language=Python, style=pythonstyle, caption={Função para gerar
↔ imagem colorida de autômatos celulares.}]
2 def gerar_imagem_colorida(matriz_de_fitras, fita_inicial, iteracoes, nome_regra=''):
3     # Verifica se há alguma fita
4     if not matriz_de_fitras:
5         print("Nenhuma fita gerada - imagem não criada.")
6         return
7
8     largura = max(len(fita) for fita in matriz_de_fitras)
9     altura = len(matriz_de_fitras)
10
11     fitas_uniformes = [list(fita) + ['0'] * (largura - len(fita))
12                        for fita in matriz_de_fitras]
13
14     array = np.array(fitas_uniformes, dtype=int)
15
16     cores = {
17         0: (0, 0, 255),
18         1: (150, 0, 200),
19         2: (255, 0, 0),
20         3: (255, 140, 0)
21     }
22
23     imagem_rgb = np.zeros((altura, largura, 3), dtype=np.uint8)
24
25     for valor, cor in cores.items():
26         imagem_rgb[array == valor] = cor
27
28     imagem = Image.fromarray(imagem_rgb, mode='RGB')
29
30     nome_imagem = f"{nome_regra}_n_{iteracoes}_fita_{fita_inicial}.png"
31     imagem.save(nome_imagem)
32     print(f"Imagem salva com sucesso: {nome_imagem}")
33 \end{lstlisting}
```