



Avaliação da Representação Neuronal Implícita para Armazenamento de Dados em ADN

Moyo Kanivengidio

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática
(2^o ciclo de estudos)

Orientadora: Prof^a. Doutora Maria Manuela Areias da Costa Pereira de Sousa

Outubro de 2025.

Declaração de Integridade

Eu, Moyo Kanivengidio, estudante inscrito sob o número MI1420, do 2.º Ciclo em Engenharia Informática, declaro, sob compromisso de honra, que o presente trabalho foi integralmente desenvolvido e redigido por mim, em conformidade com o Código de Integridade Académica da Universidade da Beira Interior.

Asseguro que não incorri em qualquer forma de fraude académica, nomeadamente plágio, falsificação, conluio ou qualquer outro comportamento que viole os princípios de honestidade científica. Declaro, igualmente, que todas as citações, imagens, excertos e demais contributos de outros autores se encontram devidamente referenciados, assumindo na íntegra a responsabilidade pela autoria e originalidade do conteúdo apresentado.

Universidade da Beira Interior, Covilhã 10/10/2025



Moyo Kanivengidio

Resumo

A crescente demanda por soluções de armazenamento de alta densidade e longa duração tem colocado o ADN como uma alternativa promissora, graças à sua capacidade de armazenar grandes volumes de dados de forma compacta e durável. Entre os diversos tipos de dados, as imagens destacam-se pela sua importância em áreas como medicina, arte, comunicação, entre outras, tornando essencial o desenvolvimento de métodos eficientes para sua representação em ADN. A natureza complexa das imagens exige abordagens especializadas que considerem suas particularidades, como redundância espacial e correlação entre pixels, além das restrições bioquímicas do ADN.

Inicialmente, as abordagens de codificação em ADN tratavam todos os tipos de dados de forma homogênea, sem adaptações específicas para imagens. Com o tempo, codificadores clássicos de imagem, como JPEG, foram adaptados para atender às restrições do ADN, e métodos baseados em inteligência artificial, como o COOL-CHIC, ganharam destaque por sua eficiência na compressão e preservação da qualidade visual. Inspirado nesses avanços, o HiDNA surgiu como um codificador inovador, combinando técnicas de IA com as restrições bioquímicas do DNA para representar imagens de forma robusta e eficiente.

Neste trabalho, exploramos o uso de representações neuronais implícitas na codificação de imagens com vista ao seu armazenamento em Ácido Desoxirribonucleico (ADN), avaliando sua capacidade de compressão, robustez e fidelidade na recuperação de dados. Métodos como o COOL-CHIC demonstram ganhos significativos na relação taxa-distorção e na redução do custo computacional, apontando para o potencial de tornar o armazenamento genético escalável e sustentável. A nossa contribuição visa avançar nesta direção, adaptando codificadores, tal como efetuado pelo HiDNA, ainda mais adaptados às necessidades específicas das imagens, abrindo caminho para aplicações práticas e inovadoras no campo do armazenamento em ADN.

Palavras-chave

Armazenamento em ADN, Representação Neuronal Implícita, Compressão de Dados, Codec HiDNA, COOL-CHIC, Taxa-Distorção.

Abstract

The growing demand for high-density, long-term storage solutions has made DNA a promising alternative, thanks to its ability to store large volumes of data in a compact and durable way. Among the various types of data, images stand out for their importance in areas such as medicine, art, communication, among others, making it essential to develop efficient methods for their representation in DNA. The complex nature of images requires specialized approaches that consider their particularities, such as spatial redundancy and correlation between pixels, in addition to the biochemical constraints of DNA. The complex nature of images requires specialized approaches that consider their particularities, such as spatial redundancy and correlation between pixels, in addition to the biochemical constraints of DNA.

Initially, DNA encoding approaches treated all types of data homogeneously, without specific adaptations for images. Over time, classic image encoders, such as JPEG, were adapted to meet DNA constraints, and methods based on artificial intelligence, such as COOL-CHIC, gained prominence for their efficiency in compression and preservation of visual quality. Inspired by these advances, HiDNA emerged as an innovative encoder, combining AI techniques with the biochemical constraints of DNA to represent images robustly and efficiently.

In this work, we explore the use of implicit neural representations in image encoding for storage in deoxyribonucleic acid (DNA), evaluating their compression capacity, robustness, and fidelity in data retrieval. Methods such as HiDNA and COOL-CHIC have demonstrated significant gains in rate-distortion and reduced computational cost, pointing to the potential to make genetic storage scalable and sustainable. Our contribution aims to advance in this direction by developing encoders more adapted to the specific needs of images, paving the way for practical and innovative applications in the field of DNA storage.

Keywords

DNA Storage, Implicit Neural Representation, Data Compression, HiDNA Codec, COOL-CHIC, Rate-Distortion.

Índice

Resumo	iii
Abstract	v
Índice	vii
Lista de Figuras	ix
Lista de Tabelas	xi
1 Contexto	1
1.1 Volume de Dados Digitais e Armazenamento de Dados	1
1.2 Codificação de Imagens para Armazenamento em ADN	3
1.3 Declaração do Problema e Objetivos	3
1.3.1 Declaração do Problema	4
1.3.2 Objetivos	4
1.4 Organização do trabalho	5
2 Estado da Arte	7
2.1 Histórico	7
2.2 Métodos de Codificação de Imagens para Armazenamento em ADN	8
2.3 Conclusão	11
3 Representação Neuronal Implícita e o Codec HiDNA	13
3.1 O codec HiDNA	13
3.1.1 O codec COOL-CHIC	14
3.1.2 Modelo de síntese	16
3.1.3 Modelo autoregressivo	18
3.1.4 Codificador Entrópico	19
3.2 Conclusão	21

4	O Codec HiDNA adaptado ao COOL-CHIC 4.0	23
4.1	O COOL-CHIC 4.0	23
4.1.1	Descodificação entrópica dos espaços latentes	24
4.1.2	Reconstrução da Imagem a partir dos Espaços Latentes	25
4.1.3	Codificação Entrópica	25
4.2	HiDNA adaptado ao COOL-CHIC 4.0	26
4.3	Conclusões	26
5	Comparação entre codecs utilizados	27
5.1	Contexto dos codecs	27
5.1.1	COOL-CHIC 1.0 vs. COOL-CHIC 4.0	27
5.1.2	HiDNA	28
5.1.3	HiDNA adaptado ao COOL-CHIC 4.0	28
5.2	Ambiente da Avaliação	29
5.2.1	Ambiente Experimental	29
5.2.2	Parâmetros de Codificação	30
5.3	Métricas de Avaliação	30
5.4	Conjunto de Dados	31
5.5	Etapas Realizadas	32
5.6	Resultados obtidos	34
5.6.1	Resultados Quantitativos	34
5.6.2	Resultados Visuais	36
5.6.3	Robustez a Erros	37
5.6.4	Desempenho Computacional	38
5.6.5	Conclusão	39
6	Conclusão e Trabalho Futuro	41
	Bibliografia	43

Lista de Figuras

2.1	Codificação geral [1].	9
2.2	Algoritmo do PAIRCODE [2].	10
2.3	: Fluxo de trabalho do fluxo de trabalho JPEG modificado (JPEG-DNA) para atender às necessidades de codificação de ADN [2].	11
3.1	Esquema geral.	14
3.2	Funcionamento do modelo de síntese.	17
3.3	Funcionamento do modelo autoregressivo.	18
4.1	Arquitetura do decodificador Cool-chic 4 [3].	24
5.1	Conjunto de imagens	32
5.2	Relação NTTP vs PSNR-YUV para os codec HiDNA Adaptado e HiDNA. . .	35
5.3	Relação NTTP vs MS-SSIM para os codec HiDNA Adaptado e HiDNA. . .	35
5.4	Exemplos Visuais para 3 imagens de HiDNA Adaptado e HiDNA.	36

Lista de Tabelas

3.1	Tabela de Goldman	20
3.2	Tabela de transcodificação quaternária	21
5.1	Comparação entre versões do COOL-CHIC e HiDNA	28
5.2	Condições Experimentais	30
5.3	Conjunto de dados JPEG DNA - Imagens não comprimidas	32
5.4	Médias das métricas por codec e taxa de compressão (λ)	34
5.5	Diferenças percentuais entre HiDNA Adaptado e HiDNA	35
5.6	Métricas Médias de Robustez a Erros por Codec e Taxa de Compressão	38
5.7	Desempenho para imagens 00005 e 00010.	39

Acrónimos

ADN *Ácido Desoxirribonucleico*

ARN *Ácido Ribonucleico*

R, G, B *Red, Green, Blue*

DVD *Digital Volatil Disc*

COOL-CHIC *Coordinate-based Low Complexity Hierarchical Image Codec*

HTML *HyperText Markup Language*

DCT *Discrete Cosine Transform*

DWT *Discrete Wavelet Transform*

PSNR *Peak Signal-to-Noise Ratio*

SSIM *Structural Similarity Index*

JPEG *Joint Photographic Experts Group*

SFC4 *Space-Filling Curves 4*

INR *Implicit Neural Representation*

MB *Megabyte*

GB *Gigabyte*

MLP *Multi-Layer Perceptron*

MUX *Multiplexador*

Capítulo 1

Contexto

O armazenamento de dados é um campo em rápida evolução, impulsionado pela crescente necessidade de guardar grandes quantidades de informação geradas diariamente. Entre as várias abordagens emergentes, o armazenamento de dados em ADN é uma das mais promissoras. O ADN, a molécula que carrega a informação genética dos organismos vivos, tem demonstrado um potencial notável como meio de armazenamento de dados, devido à sua elevada densidade de informação e estabilidade a longo prazo. Este trabalho aborda o conceito de armazenamento de imagens em ADN sintético, destacando as suas vantagens, e propõe uma adaptação do COOL-CHIC 4.0 como codec de imagens para armazenamento em ADN comparando a sua eficiência em relação ao HiDNA, que é uma adaptação do Cool-chic 1.0, que permite o armazenamento de imagens em ADN.

1.1 Volume de Dados Digitais e Armazenamento de Dados

A expansão contínua da esfera global de dados desafia o uso dos meios de armazenamento tradicionais. Um dos principais problemas é o limite de densidade. Se os meios de comunicação convencionais continuarem a ser utilizados, em breve não haverá espaço físico suficiente para armazenar os dados. Para ilustrar este desafio, se quiséssemos armazenar os dados digitais do mundo em *Digital Volatil Disc* (DVD)s, precisaríamos de uma pilha de DVDs que poderia alcançar a Lua 23 vezes ou circundar a Terra 222 vezes [4]. Uma proposta de solução para este problema de armazenamento passa por armazenar toda essa informação em ADN. Recordemos que o ADN é, basicamente, o disco rígido da natureza. O ADN é o material genético essencial presente em todos os organismos vivos. É constituído uma molécula longa e helicoidal que contém as instruções genéticas necessárias ao desenvolvimento, funcionamento, crescimento e reprodução de um organismo [5]. O ADN é composto por quatro bases nitrogenadas, os nucleotídeos: adenina (A), timina (T), citosina (C) e guanina (G). Estas bases emparelham-se de maneira complementar, respeitando a regra de emparelhamento de bases complementares, conhecida como regra de Chargaff, (A - T e C - G) para formar a estrutura de dupla hélice do ADN. Esta sequência pode ser utilizada para codificar informações digitais, onde cada base representa uma unidade de dados.

O armazenamento de dados em ADN tem emergido como uma alternativa atrativa aos meios de armazenamento tradicionais, com o potencial de alcançar uma densidade de informação extremamente elevada e uma durabilidade prolongada [6]. Devido à sua alta

densidade de dados e estabilidade química, o armazenamento em ADN oferece uma solução promissora para o armazenamento digital. O ADN apresenta um tempo de retenção de informação muito superior em comparação com a memória eletrônica, com densidades de informação cerca de 1000 vezes superiores às da memória flash [6].

As principais propriedades do ADN que o tornam um candidato promissor para o armazenamento de dados são [4]:

- **Extrema densidade de armazenamento:** O ADN é um meio de armazenamento tri-dimensional extremamente denso, com a capacidade teórica de armazenar até 455 Exabytes em 1 grama, superando significativamente os dispositivos de armazenamento convencionais.
- **Longevidade:** O ADN pode durar vários séculos, mesmo em ambientes de armazenamento adversos, em contraste com discos rígidos e fitas magnéticas, que têm uma vida útil de cinco a vinte anos.
- **Facilidade de replicação:** A replicação *in vitro* do ADN é rápida, fácil e econômica, ao contrário da cópia de grandes arquivos de exabytes que pode levar horas ou dias com dispositivos convencionais.
- **Estabilidade da informação:** O ADN é o material de armazenamento de informações da vida, cuja composição principal nunca mudará, garantindo a estabilidade da informação ao longo do tempo.

Estas propriedades tornam o armazenamento de dados em ADN uma solução altamente promissora e eficaz para o armazenamento de grandes volumes de informações de forma duradoura e eficiente. O ADN é uma alternativa interessante, pois é mil vezes mais denso que a memória física e não consome energia depois de o polímero ser sintetizado.. Além disso, o ADN pode ser armazenado durante muito tempo sem se degradar. Uma chávena de café cheia de ADN poderia, teoricamente, armazenar todos os dados digitais do mundo. Se for conservado num local escuro e fresco pode durar centenas de anos. O mais interessante é que o armazenamento de dados em ADN tem potencial para ser muito mais ecológico do que continuar a produzir discos rígidos de metal e plástico. Entre as primeiras representações de dados em ADN, temos por exemplo Organick et al. [7] que demonstraram um avanço significativo no volume de dados armazenados em ADN, no acesso aleatório e nos esquemas de codificação/descodificação, contribuindo para uma visão de sistema completo para armazenar dados em ADN. Os autores conseguiram codificar e armazenar 200 Megabyte (MB) de dados em mais de 13 milhões de oligonucleótidos de ADN, com recuperação completa dos dados sem erros de bits. Em 2013, foi criado um algoritmo para transformar um código quaternário de A, T, C e G em binário (0 e 1). Quatro anos depois, os investigadores da Microsoft e da Universidade de Washington quebraram esse

recorde [8]. Estes autores usaram codificação binária para capturar impressionantes 200 MB de dados, incluindo a Declaração Universal dos Direitos Humanos e um videoclipe de alta definição, tudo com sequência de ADN.

1.2 Codificação de Imagens para Armazenamento em ADN

Atualmente, os avanços para armazenar dados em ADN demonstram o grande potencial desta tecnologia. Visto que a imagem constitui uma parte substancial dos dados a armazenar, têm surgido nos últimos tempos vários estudos relacionados com a codificação de imagem com vista ao seu armazenamento em ADN [9][10]. O processo básico de armazenar os dados em ADN envolve a codificação da informação digital em sequências de ADN, a síntese dessas sequências em moléculas de ADN, o armazenamento a longo prazo dessas moléculas e, posteriormente, a recuperação e decodificação dos dados [10]. Essa codificação digital para ADN permite armazenar enormes fluxos de informação, incluindo imagens, em um espaço físico muito reduzido. Além disso, as técnicas de acesso aleatório e reescrita da informação armazenada em ADN têm sido desenvolvidas. Essas inovações permitem armazenar as imagens de forma flexível e eficiente, facilitando o acesso e a manipulação das mesmas. Contudo, há desafios significativos que precisam ser ultrapassados, tais como a precisão e o custo da síntese de ADN e do sequenciamento, a velocidade de leitura e escrita, e a necessidade de criar métodos de codificação de dados que sejam compatíveis com as restrições bioquímicas do ADN. Um dos primeiros trabalhos que se focou em ultrapassar esses desafios no caso da codificação de imagem, mencionando o desenvolvimento de algoritmos específicos de codificação e compressão para armazenar dados em ADN, visando lidar com o ruído e otimizar a taxa de compressão, foi proposto por Dimopoulou et al.[1].

1.3 Declaração do Problema e Objetivos

O armazenamento de dados em ADN apresenta um enorme potencial, mas requer avanços adicionais em síntese em grande escala, acesso aleatório e desenho de esquemas de codificação/decodificação eficientes para se tornar uma alternativa viável aos métodos tradicionais de armazenamento.

Esta abordagem explora o potencial da representação neuronal implícita para enfrentar os desafios inerentes ao armazenamento de dados em ADN, com especial foco na codificação e decodificação de imagens e em estudar e adaptar métodos de representação de imagem que sejam simultaneamente eficientes, compactos e compatíveis com as restrições bioquímicas do meio. As representações implícitas, por descreverem os dados de forma contínua através de funções aprendidas, permitem uma compressão eficaz da informação visual, reduzindo a quantidade de dados a sintetizar em ADN e facilitando a sua recuperação posterior. Ao

aplicar estas representações no contexto da codificação para ADN, pretende-se melhorar a taxa de compressão, a robustez ao ruído e a fidelidade da reconstrução da imagem, tornando o processo mais viável do ponto de vista prático e biológico.

1.3.1 Declaração do Problema

Com o atual crescimento da geração de dados digitais, os métodos antigos de armazenamento enfrentam limitações significativas em capacidade, durabilidade e sustentabilidade ambiental. O ADN surge como uma alternativa promissora, devido à sua densidade de armazenamento e longevidade. No entanto, os métodos de codificação de imagem atualmente usados não são concebidos tendo em mente o seu armazenamento em ADN. A Representação Neuronal Implícita - *Implicit Neural Representation* (INR) tem-se mostrado eficiente para tarefas como compressão de dados e reconstrução de sinais, mas o seu potencial no contexto do armazenamento em ADN ainda é pouco explorado. A INR permite codificar dados digitais de maneira eficiente em termos de relação débito-qualidade. Espera-se que a aplicação de INR no armazenamento de dados em ADN melhore a fidelidade da recuperação dos dados em cenários com degradação das sequências genéticas. O modelo INR apresenta uma redução significativa no custo computacional para codificação e decodificação em relação a outras abordagens baseadas em redes neuronais. O uso de INR permite uma melhor adaptação a diferentes tipos de dados digitais, como imagens e textos e espera-se que possa garantir uma grande flexibilidade para armazenamento de imagem em ADN.

1.3.2 Objetivos

O principal objetivo deste trabalho é avaliar a eficácia e a viabilidade do uso das INR como uma abordagem inovadora na codificação para armazenamento e recuperação de imagem em ADN, considerando aspetos como compressão, robustez, eficiência computacional e fidelidade na reconstrução dos dados.

Especificamente, o trabalho tem como objetivos:

- Investigar e sintetizar os avanços recentes no armazenamento de dados em ADN e no uso de Representações Neuronais Implícitas, destacando as suas aplicações, desafios e vantagens.
- Propor uma adaptação do codificador COOL-CHIC 4.0 para um modelo de compressão para armazenamento de imagem em ADN, destacando a sua abordagem de INR.
- Avaliar a viabilidade e a eficiência do codificador proposto.
- Realizar simulações com imagens para avaliar o desempenho do codificador proposto

em relação a outras representações INR em termos de compressão e robustez a erros de síntese e de sequenciação.

- Analisar a eficácia dos métodos por meio de métricas de fidelidade na reconstrução (*Peak Signal-to-Noise Ratio* (PSNR) e *Structural Similarity Index* (SSIM)), eficiência de codificação e descodificação e resistência a degradações das sequências de ADN.

1.4 Organização do trabalho

O restante documento está organizado nos seguintes capítulos:

- **Capítulo 2 - Estado de arte:** Foca-se em métodos, técnicas e soluções propostas para a codificação de dados, e especialmente de imagem, para armazenamento em ADN. Estes métodos têm por objetivo reduzir a distorção enquanto maximizam a densidade de armazenamento e garantem elevada precisão na recuperação dos dados.
- **Capítulo 3 - Representação Neuronal Implícita para o Codec HiDNA:** Explica como as redes neuronais podem modelar informações de alta dimensionalidade que permitem a compressão e recuperação de dados em aplicações de armazenamento baseado em ADN.
- **Capítulo 4 - O Codec HiDNA adaptado ao COOL-CHIC 4:** Relata a adaptação do novo HiDNA adaptado ao COOL-CHIC 4.0 explicando as etapas empregadas.
- **Capítulo 5 - Comparação entre codecs utilizados:** Mostra os passos seguidos para avaliar o HiDNA adaptado ao COOL-CHIC 4.0 e o HiDNA adaptado ao COOL-CHIC 1.0. São também apresentados e discutidos os resultados obtidos.

Capítulo 2

Estado da Arte

2.1 Histórico

A ideia de armazenamento de dados digitais em ADN remonta a 1959, quando o físico Richard P. Feynman delineou as perspectivas gerais para a criação de objetos artificiais semelhantes a objetos do microscópicos, incluindo biológicos, e com capacidades semelhantes ou até mais extensas. De fato, a capacidade de armazenar enormes quantidades de informação em um espaço extremamente pequeno, é bem conhecido pelos biólogos e resolve um mistério que, até pouco tempo, parecia inexplicável. A questão era: "como seria possível, numa célula microscópica, armazenar toda a informação necessária para organizar um organismo complexo como o ser humano?" Essa vasta quantidade de informações como, por exemplo, a cor dos nossos olhos, a nossa capacidade de pensar, ou o desenvolvimento do maxilar no embrião, que começa com um pequeno orifício para permitir o crescimento de um nervo que está contida em uma fração minúscula da célula, na forma de longas cadeias de moléculas de ADN. Nelas, aproximadamente 50 átomos são usados para codificar um único bit de informação sobre a célula. [11].

Em 1964-65, Mikhail Samoilovich Neiman [12], físico soviético, publicou três artigos sobre microminiaturização em eletrônica no nível molecular-atômico, que apresentavam considerações gerais independentes e alguns cálculos sobre a possibilidade de gravação, armazenamento e recuperação de informações em ADN sintetizado e moléculas de Ácido Ribonucleico (ARN). No ano de 2007, na Universidade do Arizona, foi criado um dispositivo usando moléculas de endereçamento para codificar locais incompatíveis dentro de uma fita de ADN [13]. Em 2011, George Church, Sri Kosuri e Yuan Gao [14] realizaram uma experiência que codificaria um livro de 659 kb que foi co-escrito por Church. Em 2012, George Church e colegas da Universidade de Harvard publicaram um artigo [15] no qual o ADN foi codificado com informações digitais que incluíam um rascunho em HyperText Markup Language (HTML) de um livro de 53 400 palavras escrito pelo pesquisador principal, 11 imagens no formato JPG e um programa em JavaScript. Mais tarde foi desenvolvido um método escalável capaz de armazenar grandes quantidades de dados em ADN, onde se conseguiu codificar 739 kilobytes de informações e reconstruir os arquivos com toda precisão [9]. Os autores sugerem que, esta abordagem pode tornar-se viável para armazenamento digital de longo prazo e grande escala dentro de uma década, graças ao avanço da tecnologia e a redução dos custos de síntese de ADN. Em março de 2018 foram codificados e armazenados 35 arquivos distintos, totalizando mais de 200 MB de dados, em mais de 13 milhões de oligonucleotídeos de ADN, com a capacidade de

recuperar cada arquivo individualmente, sem erros, usando um método de acesso aleatório [9]. O desenvolvimento de uma biblioteca de uma sequência curta de nucleotídeo e um algoritmo de decodificação eficiente permitiu a recuperação de dados com menor cobertura de leitura, demonstrando a viabilidade de um sistema de armazenamento e recuperação de dados em ADN em grande escala. Em junho de 2019, os cientistas relataram que todos os 16 Gigabyte (GB) da Wikipedia em inglês foram codificados em ADN sintético [8]. O primeiro artigo descrevendo o armazenamento de dados em sequências nativas de ADN por meio de corte enzimático foi publicado em abril de 2020 [16]. Este trabalho demonstra um novo método de registro de informações no *backbone* de ADN que permite acesso aleatório bit a bit e computação na memória.

No que diz respeito à codificação de imagem para representação em ADN, um dos primeiros trabalhos resultou da colaboração entre o artista Joe Davis e investigadores de Harvard [14]. Neste trabalho é apresentado o processo de armazenamento de uma imagem numa sequência de ADN em *E.coli*, organizada em uma matriz 5×7 que, uma vez decodificada, formou a imagem de uma antiga runa germânica que representava a vida e a Terra feminina.

2.2 Métodos de Codificação de Imagens para Armazenamento em ADN

Nos primeiros trabalhos de codificação de imagem para armazenamento em ADN o processo de codificação clássico foi adaptado para a representação dos dados nos 4 nucleotídeos. Esta adaptação envolve elementos como compressão de imagem mas também exige lidar com restrições biológicas impostas pela síntese de ADN e procedimentos de sequenciamento (exemplo na Figura 2.1). Ademais, os métodos podem concentrar-se na otimização da quantidade de oligonucleotídeos empregues para a sequência codificada e na diminuição do custo do sequenciamento.

Em 2021 uma imagem foi convertida numa sequência de ADN utilizando uma abordagem baseada em blocos de Discrete Cosine Transform (DCT), onde a imagem foi representada por uma sequência de nucleotídeos A, T, C e G. Os oligos formatados sintetizados a partir da sequência de ADN são convertidos em moléculas de ADN, que são sequenciadas para recuperar os oligos originais para decodificação [17]. A síntese de ADN é um processo praticamente livre de erros quando a sequência a ser sintetizada em ADN não excede o comprimento de 150 nucleotídeos [1]. Aqui, destacam-se as limitações inerentes à codificação de dados em ADN, especialmente relacionadas à síntese e sequenciamento. A precisão da síntese varia com o comprimento da sequência a ser sintetizada. Neste trabalho utilizaram uma técnica que permite controlar a taxa de compressão através de uma alocação ótima de nucleotídeos e, consequentemente, controlar o custo de síntese de ADN, isto serviu para minimizar o custo de decodificação do ADN otimizando a quantidade de

oligo-nucleotídeos sintéticos necessários para um sequenciamento robusto.

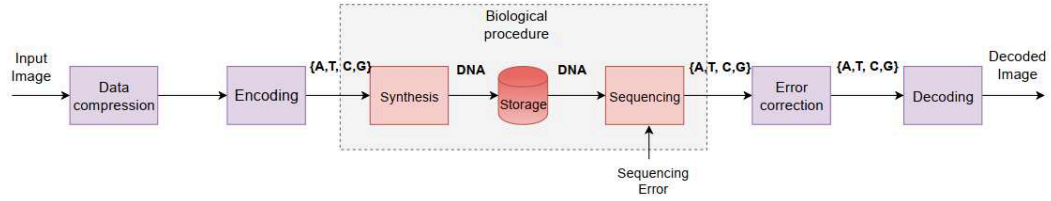


Figura 2.1: Codificação geral [1].

Esta técnica tem como objetivo principal armazenar os dados em ADN, codificando os dados de entrada usando um código quaternário composto pelas bases nitrogenadas A, T, C e G. Inicialmente, a imagem de entrada deve ser compactada utilizando um algoritmo de compactação de imagem com perdas. Em seguida, essas imagens foram codificadas em C, G, T e A para que posteriormente sejam sintetizadas em ADN (Figura 2.1). Mais tarde este processo foi melhorado e foi proposto o método *PAIRCODE* [17].

O método *PAIRCODE* garante que a informação seja guardada de forma eficiente e robusta enquanto minimiza a probabilidade de erros durante a síntese e a sequenciação do ADN. Este método permite construir códigos de comprimento fixo, podendo ser aplicado a qualquer tipo de dados de entrada, formando um par de nucleotídeos (como AT, AC, AG, etc.). O método pode ser seguido na Figura (2.2). O objetivo deste método é de transformar uma imagem comprimida (após quantização) numa sequência de ADN, evitando repetições de padrões que podem causar problemas. O método funciona da seguinte maneira:

1. É aplicada a transformada em wavelets discreta (Discrete Wavelet Transform (DWT)) para explorar as redundâncias espaciais da imagem, gerando sub-bandas que são posteriormente comprimidas através de um quantizador escalar uniforme \mathbf{Q} . O valor do passo de quantização \mathbf{q} é definido com base num algoritmo de alocação ótima de nucleótidos, o qual tem como objetivo minimizar a distorção da imagem comprimida, ao mesmo tempo que limita o comprimento da cadeia de nucleótidos gerada pelo codificador em cada sub-banda de wavelet.
2. São criados dois alfabetos:
 - D_1 : Pares de nucleotídeos (ex.: AT,AC,AG,TA,TC,TG,CA,CT,GA,GT).
 - D_2 : Nucleotídeos individuais (A, T, C, G).
3. Os valores quantizados são mapeados para sequências de ADN usando combinações de D_1 e D_2 , da seguinte forma:
 - Se a sequência de ADN for par, usa-se apenas pares de D_1 .
 - Se for ímpar, usa-se pares de D_1 mais um nucleotídeo de D_2 .

4. Para evitar sequências repetitivas (comum em imagens quantizadas), o algoritmo usa um mapeamento pseudo-aleatório:
 - Cada valor quantizado é associado a várias sequências de ADN possíveis.
 - Isso garante que não haja repetições longas de padrões.
5. A imagem quantizada é transformada numa sequência de ADN pronta para ser sintetizada e evitando repetições de forma a reduzir a probabilidade de erros de síntese e de sequenciação.

A escolha deste método prevê que a codificação evite homopolímeros e que o valor da percentagem de GC não ultrapassa o de AT para evitar os erros na reconstrução. Esta é uma segunda restrição que permite reduzir a probabilidade de erro durante a sequenciação.

Algorithm 1 Encoding Algorithm

- 1: Compute length l of codewords needed for encoding all k levels of quantization:
- 2: **if** $\log_{10} k$ not an integer **then**
- 3: **if** $10^{\lfloor \log_{10} k \rfloor} * 4 \leq k$ **then**
- 4: $l = \lfloor \log_{10} k \rfloor * 2 + 1$
- 5: **else** $l = \lceil \log_{10} k \rceil * 2$
- 6: **end if**
- 7: **else** $l = \log_{10} k * 2$
- 8: **end if**
- 9: Build code \mathcal{D} of L different codewords:
- 10: **if** l is even **then**
- 11: Construct all possible codewords of length l using $\frac{l}{2}$ choices from \mathcal{D}_1
- 12: **else if** l is odd **then**
- 13: Construct all possible codewords of length l by using $\frac{l}{2}$ choices from \mathcal{D}_1 adding one symbol from \mathcal{D}_2
- 14: **end if**
- 15: Mapping of index values of quantization to codewords from \mathcal{D}

Compute: $m = \lfloor \frac{l}{k} \rfloor$

Compute: $\Gamma(\hat{x}^i) = \mathcal{D}^*(i + rand(1, m - 1) * k)$

Figura 2.2: Algoritmo do PAIRCODE [2].

Neste trabalho o método proposto usa codificação de comprimento fixo. Os mesmos autores propuseram um novo codificador [2], desta vez de comprimento variável, inspirado no codificador *Joint Photographic Experts Group* (JPEG) e adaptado para armazenamento de imagens digitais em ADN. Com este método, os autores obtiveram uma solução inspirada no núcleo essencial da norma JPEG. No estudo, apresentaram o JPEG-DNA como uma variante adaptada do algoritmo JPEG para a codificação de uma imagem em uma representação quaternária restrita.

O JPEG clássico proporciona uma codificação ótima expressa numa representação binária, empregando duas técnicas principais de codificação. Esta codificação transforma cada valor em sua representação binária utilizando um número de bits predefinido pelo campo de categoria que o antecede. Assim, torna-se evidente que, para adaptar o algoritmo JPEG existente e fornecer uma representação quaternária, foi necessário substituir estas

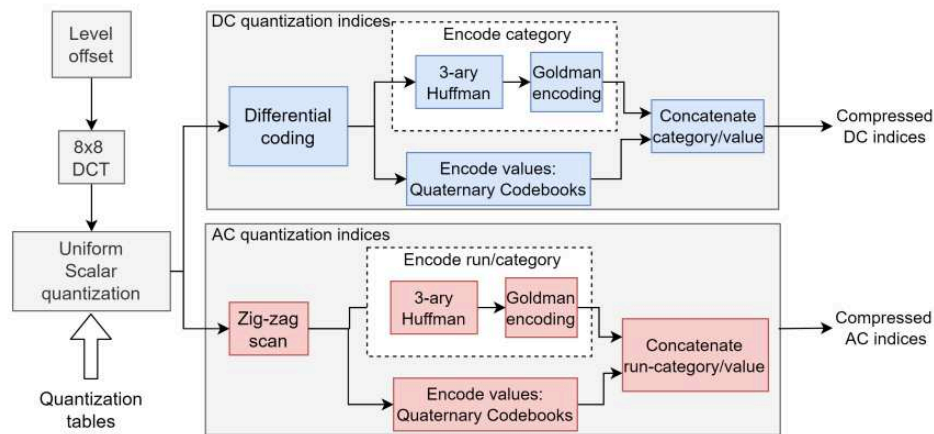


Figura 2.3: Fluxo de trabalho do fluxo de trabalho JPEG modificado (JPEG-DNA) para atender às necessidades de codificação de ADN [2].

duas codificações binárias por outras adequadas ao quaternário, respeitando as restrições de codificação do ADN (ver Figura 2.3). Mais especificamente, é efetuada uma codificação de Huffman ternária seguida do algoritmo de codificação de Goldman e por dicionários quaternários [9].

Os trabalhos mais recentes têm seguido a tendência dos codificadores de imagem clássicos e apostado na utilização de redes neurais para aumentar a eficiência da codificação. Em [18] os autores adaptaram o Codec Hierárquico de Imagem de Baixa Complexidade Baseado em Coordenadas *Coordinate-based Low Complexity Hierarchical Image Codec* (COOL-CHIC) [19] que oferece uma abordagem inovadora para a compressão de imagens, aproveitando representações neurais baseadas em coordenadas. Este método é valioso devido à sua baixa complexidade e desempenho competitivo comparando-se com os codecs existentes. O COOL-CHIC permite alcançar um nível de compressão próximo ao dos *codecs* modernos, como o HEVC. Em termos de eficiência, o codificador proposto por [18] superou os seus antecessores, inclusive os sistemas baseados em *autoencoders*[20]. Este método é a base do trabalho proposto nesta dissertação e será descrito detalhadamente no próximo capítulo.

2.3 Conclusão

Resumindo, as conclusões gerais dos estudos mostram progressos significativos nas metodologias usadas para armazenamento de informações genéticas. Os investigadores conseguiram codificar e guardar imagens digitais no ADN, demonstrando técnicas de compressão eficazes para diminuir os custos de síntese [4][21]. Essa nova abordagem possibilita a retenção de volumes consideráveis de informações de maneira compacta, com melhorias na segurança e a habilidade de ler arquivos em paralelo [1]. Adicionalmente, a criação de algoritmos de codificação de imagem inovadores especificamente desenhados para o arma-

zenamento em ADN resultou em avanços em comparação com as técnicas convencionais, assegurando uma recuperação de dados mais precisa e diminuindo os gastos com sequenciamento [22]. Em termos gerais, essas descobertas ressaltam o potencial do armazenamento de dados de ADN como uma alternativa escalável e acessível para preservar e administrar grandes quantidades de informações.

É possível controlar o custo da síntese de ADN de forma eficaz, comprimindo as imagens de entrada sem afetar significativamente a qualidade visual. Além disso, o custo de sequenciamento pode ser consideravelmente diminuído obtendo uma reconstrução impecável com apenas uma pequena parte dos dados sequenciados. A diminuição de custos é essencial, pois poderia tornar o armazenamento de ADN mais acessível e popular, representando um avanço importante na área do armazenamento de dados. Os resultados indicam um potencial otimista para a criação e análise de ADN com um bom custo-benefício, permitindo uma maior incorporação e aplicação de tecnologias de armazenamento de dados de ADN.

Para aprimorar a eficiência da codificação de imagens vamos explorar as novas abordagens na codificação de imagens baseadas em INR. Esta abordagem permite uma modelagem contínua e eficiente das imagens, que vai contribuir para uma reconstrução de alta fidelidade e um melhor aproveitamento dos dados armazenados. No capítulo 3, abordamos como uma INR foi integrada ao codec HiDNA, destacando os seus benefícios e implicações para a otimização do armazenamento de dados em ADN.

Capítulo 3

Representação Neuronal Implícita e o Codec HiDNA

A Representação Neuronal Implícita (INR) é uma abordagem inovadora, que permite representar imagens através de redes neurais, utilizando um espaço latente contínuo em vez de representações explícitas baseadas em pixels. Esta técnica tem sido eficaz em tarefas como compressão de dados e reconstrução de sinais, graças à sua capacidade de codificar imagens com alta fidelidade e baixa complexidade. O COOL-CHIC (*Coordinate-based Low Complexity Hierarchical Image Codec*) é um exemplo de INR aplicada à codificação de imagens [23]. Utiliza uma arquitetura baseada em *Multi-Layer Perceptrons* (MLPs) para aprender representações compactas e permitir a reconstrução de imagens a partir dessas representações. Este codec combina a capacidade de modelação de um *Multi-Layer Perceptron* (MLP) (para o modelo de síntese) com um modelo autoregressivo, proporcionando um sistema que codifica as imagens de forma eficiente. A compressão obtida é competitiva com codecs modernos como o HEVC, com a vantagem de maior flexibilidade e menor complexidade computacional [24]. Com base neste modelo, foi desenvolvido o HiDNA, um codec adaptado às características específicas do armazenamento em ADN e às restrições bioquímicas do armazenamento em ADN [18]. O HiDNA integra os princípios do COOL-CHIC com mecanismos de mapeamento quaternário compatíveis com sequenciação e síntese de ADN, permitindo gerar sequências de nucleotídeos e produzindo um ficheiro FASTA (.fasta) contendo um conjunto de oligos que verificam as restrições bioquímicas de adequadas ao processo biológico. A figura 3.1 ilustra o funcionamento geral do codificador HiDNA, ilustrando o fluxo desde a entrada da imagem até a produção da sequência de ADN. Este será descrito detalhadamente nas secções que se seguem.

3.1 O codec HiDNA

Na codificação com perda de uma imagem com n canais de cor, o objetivo é obter uma representação comprimida \hat{x} que reduza a taxa de bits necessária para armazenamento, minimizando ao mesmo tempo a distorção em relação à imagem original x . Esta codificação com perda tem como objetivo minimizar tanto a taxa necessária para transmitir \hat{x} quanto a distorção entre x e \hat{x} e pode ser representada pela função de custo:

$$\min D(x, \hat{x}) + \lambda R(\hat{x}) \quad \text{ou} \quad C_t = D - \lambda \times T \quad (3.1)$$

onde

- C_t : é o custo total,

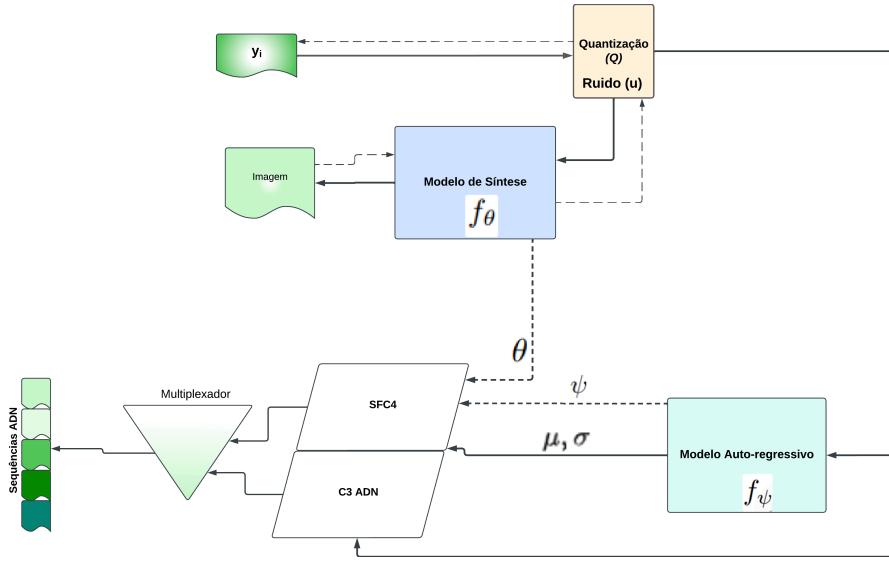


Figura 3.1: Esquema geral.

- D : distorção (pode usar por exemplo, métrica de distorção PSNR ou MSE),
- T ou R : taxa em bits por pixel,
- λ : é o hiperparâmetro de regularização que controla o compromisso entre qualidade e compressão.

Esta função de custo será usada para minimizar a distorção entre a imagem original e a imagem reconstruída, e minimizar a taxa de compressão necessária para armazenar a representação latente calculada no modelo de síntese. A taxa estimada de bits necessária para codificar os mapas latentes é calculada via o modelo autoregressivo. Visto que estes dois modelos são baseados no codec COOL-CHIC, este será descrito na secção 3.1.1. Os modelos de síntese e autoregressivo serão detalhados nas secções 3.1.2 e 3.1.3. Finalmente o codificador entrópico composto pelos codificadores C3 ADN e SFC4 serão descritos na secção 3.1.4.

Após a codificação entrópica o Multiplexador (MUX) combina os diferentes fluxos de dados numa única saída como uma sequência organizada e armazenada num ficheiro de formato comum para sequências biológicas, FASTA.

3.1.1 O codec COOL-CHIC

O COOL-CHIC é um codec de imagem baseado em representações neuronais implícitas, cuja arquitetura combina dois componentes principais: um modelo de síntese e um modelo de probabilidade autoregressivo. Esta combinação permite representar e comprimir imagens de forma eficiente, equilibrando qualidade de reconstrução, taxa de compressão e complexidade computacional. Durante o treino da rede, a função de custo (equação 3.1) considera dois fatores: a distorção entre a imagem original e a reconstruída a partir da

MLP de síntese; e, a taxa estimada de bits necessária para codificar os mapas latentes via o modelo autoregressivo.

O modelo de síntese do COOL-CHIC baseia-se numa representação latente hierárquica da imagem. Durante a codificação, a imagem original (x) é transformada num conjunto de mapas latentes discretos (\hat{y}) de diferentes resoluções espaciais, desde mapas de baixa resolução, que capturam informação estrutural global, até mapas de alta resolução, responsáveis por detalhes finos. Cada um destes mapas latentes é redimensionado (via interpolação bicúbica) para a resolução original da imagem. Os mapas interpolados são então concatenados formando um tensor \hat{z} com a mesma resolução espacial da imagem, mas com múltiplos canais. Para cada pixel (i, j) , o vetor $\hat{z}_{i,j}$ resultante é introduzido numa MLP de síntese, que estima os valores de intensidade (RGB) do pixel correspondente. Este processo transforma a imagem (x) num conjunto de vetores latentes discretos (\hat{y}) que serão posteriormente armazenados ou transmitidos ($\hat{x}_{i,j} = f_{\theta}(\hat{y}_{i,j}) = (R, G, B)$). Este processo será detalhado na secção 3.1.2.

A compressão resulta, assim, na substituição da imagem original pelos mapas latentes e pelos pesos da MLP de síntese. Antes de ser armazenada ou transmitida, a informação passa por um processo de quantização (\mathcal{Q}), onde valores contínuos são discretizados para reduzir a quantidade de dados necessários. Como não é possível otimizar diretamente uma variável discreta através de uma otimização baseada num gradiente descendente, a operação de quantização é substituída pela adição de ruído u durante o treino.

Para garantir compressão eficiente, é necessário codificar os mapas latentes discretos com uma taxa próxima do limite teórico de entropia. Para isso, o COOL-CHIC utiliza um modelo de probabilidade autoregressivo, responsável por estimar a distribuição estatística dos valores latentes. Este modelo aprende a prever a probabilidade de cada vetor latente $\hat{y}_{i,j}$, condicionado ao seu contexto causal, isto é, aos valores de vetores vizinhos já codificados. A estimativa é feita através de uma MLP probabilística, que modela os vetores latentes como amostras discretizadas de uma distribuição de Laplace, cujos parâmetros (média e desvio padrão) dependem do contexto espacial. Esta etapa será detalhado na secção 3.1.3. Com esta estimativa probabilística, os vetores latentes \hat{y} , e os parâmetros ψ , θ , podem ser codificados com codificação entrópica, reduzindo o número de bits necessários para representar a imagem.

A codificação entrópica baseia-se numa estimativa p da distribuição de probabilidade (desconhecida) do sinal q [25]. A dimensão da distribuição latente $p(\hat{y})$ é várias ordens de grandeza superior à dos parâmetros MLP θ e ψ . A taxa associada aos parâmetros MLP θ e ψ não é levado em conta para essa função objetivo. A taxa pode assim ser calculada usando a equação:

$$R(\hat{y}) = \mathbb{E}_{\hat{y} \sim q} [-\log_2 p(\hat{y})] \quad (3.2)$$

Considerando o modelo de probabilidade autoregressivo $p_{\psi}(\hat{y})$ implementado pela MLP f_{ψ} .

podemos reescrever a equação (3.1) para expor mais claramente as quantidades otimizadas:

$$\min_{y, \theta, \psi} [D(x, \hat{x})] - \lambda [\log_2 p_\psi(\hat{y})] \quad (3.3)$$

Uma vez a otimização concluída a codificação entrópica é realizada.

A decodificação de uma imagem, usando o COOL-CHIC, consiste em três etapas principais [19]:

- os parâmetros de dois MLPs (f_θ e f_ψ) são recuperados do fluxo de bits.
- a f_ψ é utilizado para decodificar \hat{y} , um conjunto de L variáveis latentes discretas bidimensionais.
- o valor *Red*, *Green*, *Blue* (R, G, B) de cada pixel \hat{x}_{ij} da imagem comprimida é calculado alimentando a representação latente densa para o MLP de síntese:

$$\hat{x}_{ij} = f_\theta(\hat{z}_{ij}) \quad (3.4)$$

O modelo de síntese e o modelo autoregressivo desempenham um papel importante no processo de codificação de imagens, considerando o codec HiDNA. Dada a sua importância para o codec HiDNA, irão ser apresentados detalhadamente.

3.1.2 Modelo de síntese

O modelo de síntese é responsável pela reconstrução da imagem a partir do espaço latente utilizando uma rede neuronal treinada que reduz a diferença entre a imagem original e a imagem decodificada [18]. Para o seu funcionamento existem passos a seguir, tais como: Entrada do espaço latente, processamento através da rede neuronal, comparação com a imagem original para calcular as perdas e por fim, atualização dos pesos da rede para melhorar a precisão da imagem reconstruída.

O modelo de síntese, que pode ser visto na Figura 3.5, segue os passos seguintes:

- O modelo recebe como entrada uma representação \hat{y} compacta da imagem armazenada no espaço latente. Esta representação é obtida através do processo de codificação prévia da imagem. Onde:

$$\hat{y} = \{\hat{y}_k \in \mathbb{Z}^{H_k \times W_k}, k = 0, \dots, L - 1\} \quad (3.5)$$

com $H_k = \frac{H}{2^k}$, $W_k = \frac{W}{2^k}$, representando resoluções hierárquicas.

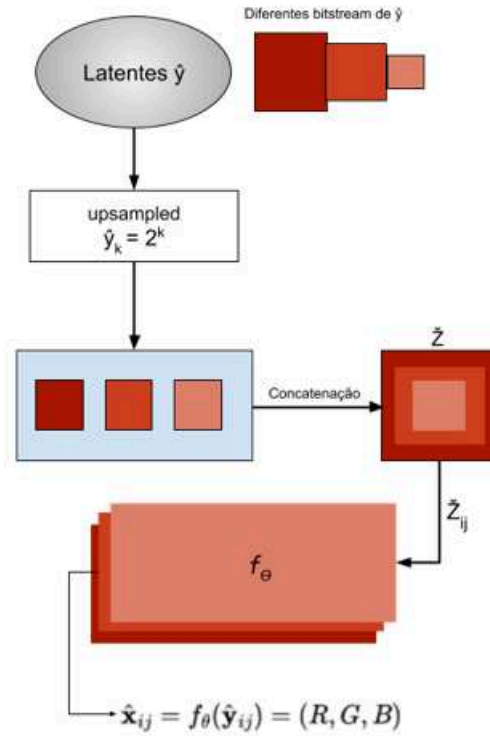


Figura 3.2: Funcionamento do modelo de síntese.

- O processamento para este modelo é uma rede neuronal que se comporta como um MLP f_θ treinada para reconstruir a imagem original a partir do espaço latente.

$$\hat{x}_{ij} = f_\theta(z_{ij}), \quad z_{ij} = \{z_{ij,k}, k = 0, \dots, L - 1\} \quad (3.6)$$

onde $z_{ij,k}$ é a representação do espaço latente densa para a posição (i, j) e f_θ é a MLP que realiza a síntese da imagem.

Para tal, existe uma função de ativação que é utilizada de acordo com a arquitetura específica.

- A saída do modelo é a imagem reconstruída, que é comparada com a original para calcular as perdas. A função de perda mais comum é o MSE.

$$L = MSE(x, \hat{x}) = \frac{1}{N} \sum_{i=1}^N \|x_i - \hat{x}_i\|^2 \quad (3.7)$$

- A atualização dos pesos da rede durante o treino é feita por *backpropagation* utilizando

$$\theta \leftarrow \theta - \eta \frac{\partial L}{\partial \theta} \quad (3.8)$$

onde η é a taxa de aprendizagem.

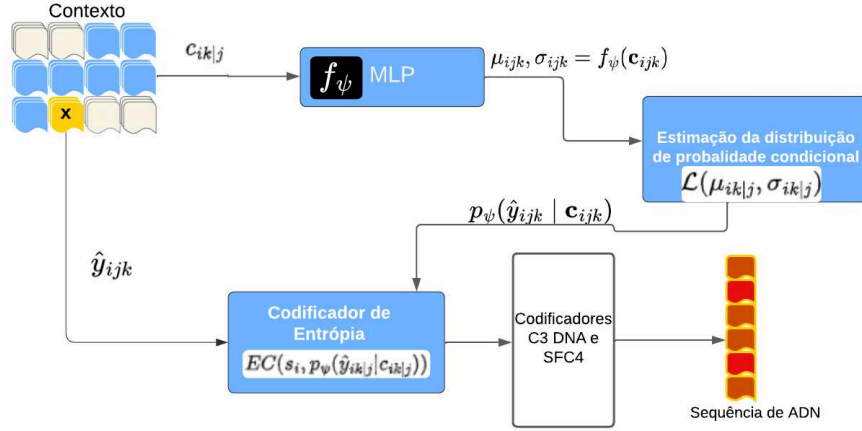


Figura 3.3: Funcionamento do modelo autoregressivo.

- Após o treino, o modelo f_θ consegue reconstruir imagens a partir de novas representações latentes \hat{y} , gerando:

$$\hat{x} = f_\theta(\hat{y}). \quad (3.9)$$

3.1.3 Modelo autoregressivo

O modelo autoregressivo (f_ψ) tem a responsabilidade de estimar a distribuição dos objetos no espaço latente com base nos seus vizinhos, utilizando um contexto de elementos anteriores. Para que este modelo funcione, existem passos que incluem entrar no espaço latente, contextualizar os itens, estimar a distribuição, codificar o item atual e iterar o processo. As funções matemáticas aplicáveis incluem a modelação de distribuição Laplaciana, o cálculo de parâmetros e a codificação de entropia. A figura 3.3 ilustra o funcionamento do modelo.

Este modelo desempenha um papel importante na estimativa da distribuição dos elementos do espaço latente utilizando a informação contextual dos elementos da vizinhança. Comporta-se da seguinte forma:

- Este modelo recebe como entrada os elementos quantizados do espaço latente \hat{y}_{ik} no nível k . Cada elemento a ser codificado será influenciado pelos elementos anteriores $c_{ik} \in \mathbb{Z}^C$ que formam um contexto com a seguinte relação:

$$p_\psi(\hat{y}) = \prod_{i,k} p_\psi(\hat{y}_{ik}|c_{ik}), \quad (3.10)$$

onde p_ψ é a distribuição estimada pelo modelo autoregressivo parametrizado por ψ .

- Para cada elemento do espaço latente, o modelo utiliza um contexto composto por 12 elementos previamente quantizados. Esta noção permite que o modelo assuma dependências temporárias e especiais entre os elementos.
- Para cada elemento \hat{y}_{ik} , a distribuição é modelada como Laplaciana, com parâmetros de expectativa μ_{ik} e escala σ_{ik} :

$$p_{\psi}(\hat{y}_{ik}|c_{ik}) = \int_{\hat{y}_{ik}-0.5}^{\hat{y}_{ik}+0.5} \mathcal{L}(y; \mu_{ik}, \sigma_{ik}) dy, \quad (3.11)$$

onde $\mathcal{L}(y; \mu, \sigma)$ representa a função densidade de uma distribuição Laplaciana.

- Os parâmetros estimados μ_{ik} e σ_{ik} da distribuição Laplaciana são utilizados diretamente num codificador entrópico (*Space-Filling Curves 4* (SFC4)) para codificar o elemento atual e, por fim, o codificador transforma a representação da distribuição numa sequência de bits. A equação 3.12 calcula o débito.

$$R(\hat{y}) = -\log_2 p_{\psi}(\hat{y}) = -\sum_{i,k} \log_2 p_{\psi}(\hat{y}_{ik}|c_{ik}) \quad (3.12)$$

3.1.4 Codificador Entrópico

Calculada a estimativa probabilística, os vetores latentes \hat{y} , e os parâmetros ψ , θ , podem ser codificados com codificação entrópica.

Os vetores latentes \hat{y} são passados pelo codificador entrópico C3 DNA [26], que os transforma em sequências de nucleótidos para que as sequências sejam devidamente armazenadas no ADN. Este codificador é descrito na secção 3.1.4.1.

Os modelos de Síntese e Autorregressivo, θ e ψ , respetivamente, são ambos codificados usando um codificador de entropia quaternária restrita denominado SFC4 [22]. Este codificador é descrito na secção 3.1.4.2.

3.1.4.1 Codificador C3 ADN

O codificador C3 ADN [27, 20] é um sofisticado sistema de codificação projetado para converter dados binários em sequências de DNA sintético, garantindo que as restrições bioquímicas sejam rigorosamente respeitadas [26]. Inspirado no codificador entrópico MQ coder do JPEG 2000, este método foi desenvolvido para evitar padrões indesejáveis que poderiam comprometer a estabilidade e a leitura dos dados armazenados no DNA, como por exemplo os homopolímeros (base que aparece repetidamente mais do que 3 vezes consecutivas - ex: AAAA).

O codificador C3 ADN funciona da seguinte forma:

1. É definido um código quaternário restrito, que usa palavras de 3 nucleotídeos (triple-

tos) cuidadosamente escolhidas de modo a evitar essas repetições problemáticas. O conjunto de palavras válidas forma um dicionário C_3 com 48 tripletos válidos, como por exemplo: $\{AAT, AAC, AAG, ATA, \dots GCT, GCG, GGA\}$. Cada palavra no dicionário é associada a um símbolo numérico $k \in [0, 47]$, formando um mapeamento direto entre números e cadeias de DNA seguras.

2. Em vez de codificar os intervalos aritméticos em binário (como tradicionalmente é efetuado pelo MQ coder do JPEG 2000), eles são codificados em base 48.
3. Cada elemento em base 48, é mapeado num dos 48 símbolos definidos no dicionário C_3 , garantindo que a conversão seja consistente e bioquimicamente válida.

3.1.4.2 O Codificador SFC4

O codificador SFC4 [22] é um codificador baseado em árvore quaternária que impõe restrições em posições específicas da árvore, de forma a manter o equilíbrio entre eficiência de codificação e validade bioquímica. A árvore quaternária é uma versão adaptada do algoritmo Shannon-Fano, que constrói a árvore da raiz para as folhas e permite controlar a profundidade de cada nó durante a recursão.

O codificador funciona com dois algoritmos:

- Algoritmo 1: inicia a construção da árvore a partir da raiz.
- Algoritmo 2: realiza a divisão recursiva do conjunto de símbolos, decidindo se a ramificação naquele nível será ternária (3 filhos) ou quaternária (4 filhos) com base na profundidade atual d e no parâmetro global $maxhl$ (comprimento máximo permitido de homopolímeros). A árvore é ternária nos níveis em que $d \equiv 0(\text{mod } maxhl)$, e quaternária nos restantes.

Cada símbolo codificado gera uma sequência com dígitos 0, 1, 2, ou 3. Essas sequências são depois codificadas para DNA, utilizando duas tabelas distintas. Quando o símbolo está numa posição ternária (restrita), usa-se a tabela de Goldman, Tabela 3.1 [9]. Caso contrário, usa-se a tabela de transcodificação quaternária, Tabela 3.2.

Nucleotídeo anterior	0	1	2
A	T	C	G
T	A	C	G
C	A	T	G
G	A	T	C

Tabela 3.1: Tabela de Goldman

Este método garante que as cadeias geradas respeitam a restrição de homopolímeros, mantendo boa capacidade de compressão.

Símbolo	0	1	2	3
Nucleotídeo	A	T	C	G

Tabela 3.2: Tabela de transcodificação quaternária

3.2 Conclusão

Neste capítulo, foi explorado o uso de Representações Neurais Implícitas (INR) para compressão e armazenamento de imagens, com ênfase no codec COOL-CHIC e na sua adaptação ao contexto do armazenamento em ADN, designada por HiDNA. A INR demonstrou-se uma abordagem promissora, ao permitir representar imagens através de modelos neuronais compactos que operam num espaço latente contínuo, reduzindo significativamente a complexidade computacional e mantendo altos níveis de fidelidade visual.

O codec COOL-CHIC constitui um exemplo bem-sucedido desta técnica, combinando um modelo de síntese com um modelo autoregressivo probabilístico para gerar representações comprimidas de imagens. A sua arquitetura hierárquica e o uso de MLPs tornam possível alcançar uma elevada eficiência de codificação, com desempenho competitivo face a codecs tradicionais.

Partindo desta base, o HiDNA foi desenvolvido para permitir a codificação direta de imagens comprimidas em sequências de nucleótidos compatíveis com as restrições bioquímicas do ADN sintético. A integração dos codificadores entrópicos C3 ADN e SFC4 garante que as sequências resultantes evitam erros comuns de síntese e leitura, como homopolímeros excessivos, mantendo uma boa taxa de compressão.

Com esta abordagem, é possível armazenar imagens digitais no ADN de forma mais robusta, eficiente e biologicamente segura, abrindo caminho para futuras aplicações de Representações Neurais Implícitas em sistemas de armazenamento molecular.

No próximo capítulo, será apresentado o codificador adaptado do HiDNA, que utiliza a nova versão, mais eficiente, do codificador COOL-CHIC onde a MLP no modelo de síntese é substituída por redes neuronais convolucionais. Posteriormente estes dois codificadores, HiDNA e HiDNA Adaptado, serão comparados em termos de desempenho nas tarefas de compressão e recuperação de imagens a partir de sequências genéticas.

Capítulo 4

O Codec HiDNA adaptado ao COOL-CHIC 4.0

O COOL-CHIC 4.0 (*Coordinate-based Low Complexity Hierarchical Image Codec*), desenvolvido pela *Orange Open Source*, é um codec de compressão de imagem baseado em redes neurais que combina alta eficiência com baixa complexidade computacional. Neste capítulo são descritos os passos realizados para integrar o COOL-CHIC 4.0 no contexto da compressão de imagens, com uma adaptação inovadora que permite a conversão do fluxo de bits em sequências de ADN, explorando o potencial do ADN como meio de armazenamento de dados [10, 28].

A versão 4.0 do COOL-CHIC apresenta avanços significativos que expandem as suas capacidades e aplicabilidade. As melhorias na eficiência da codificação incluem algoritmos otimizados que reduzem a taxa de bits por pixel (bpp) sem comprometer a qualidade perceptiva, resultando em ganhos de até 15% em relação às versões anteriores, como avaliado em conjuntos de dados como o UVG [28]. O suporte de diferentes resoluções permite que o codec processe vídeos de baixa resolução (por exemplo, 360p) a ultra-alta definição (por exemplo, 4K), ajustando dinamicamente a arquitetura neuronal para manter a eficiência em cada caso. Além disso, a otimização para dispositivos com recursos limitados foi conseguida através da redução da complexidade computacional utilizando técnicas como a poda de redes neurais e a quantização de pesos, tornando o COOL-CHIC viável para execução em dispositivos *IoT*, *smartphones* e outros sistemas com restrições de energia e memória [28].

Neste trabalho foi integrada a codificação em ADN no COOL-CHIC 4.0. Após esta adaptação foi avaliado o desempenho em termos de nucleotídeos por pixel (ntpp) e qualidade usando as métricas PSNR-YUV, VMAF, etc. Neste capítulo apresentam-se as diferenças do Cool-Chic 4 em relação ao descrito no capítulo anterior e detalha-se a adaptação efetuada. As experiências conduzidas com os dados fornecidos e os resultados obtidos serão apresentados no capítulo seguinte.

4.1 O COOL-CHIC 4.0

O COOL-CHIC 4.0 é um codec de compressão de imagens baseado em redes neurais, concebido para oferecer uma elevada capacidade com baixas taxas de bits e baixa complexidade computacional [28, 29]. Utiliza uma abordagem hierárquica coordenada, na qual a imagem é representada por uma rede neuronal que modela os pixels num espaço contínuo. A seguir apresentam-se as principais características do COOL-CHIC 4.0, conforme

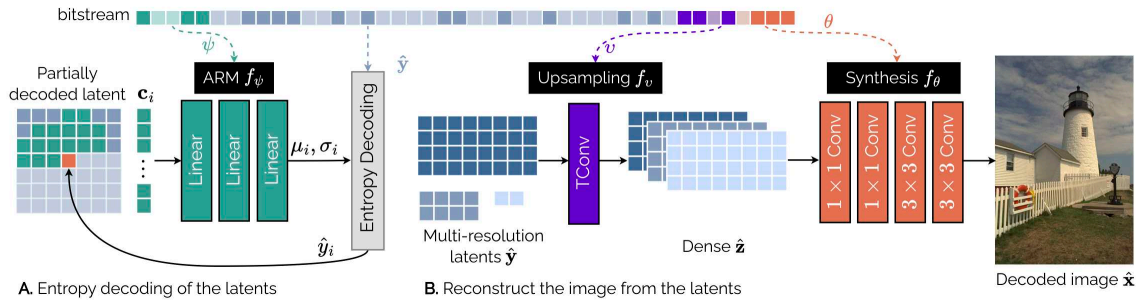


Figura 4.1: Arquitetura do decodificador Cool-chic 4 [3].

descritas na documentação oficial [28, 30, 31].

A arquitetura desta versão consiste numa rede neuronal de baixa complexidade que gera representações latentes compactas. Ao contrário dos codecs tradicionais, opera em um espaço estruturado, permitindo a representação em multi-resolução da imagem [3].

O sistema de decodificação eficiente para compressão de dados está dividido em duas etapas principais que serão descritas de seguida (ver Figura 4.1): decodificação entrópica dos espaços latentes e reconstrução da imagem a partir desses espaços.

4.1.1 Descodificação entrópica dos espaços latentes

Os espaços latentes são recuperados do ficheiro comprimido e organizados numa estrutura hierárquica. Segue-se então a aplicação do modelo autoregressivo com parâmetros ψ , ARM f_ψ , idêntico ao da versão 1 e detalhado na secção 3.1.3. Este modelo é responsável por decodificar os espaços latentes c_i numa representação hierárquica \hat{y}_ℓ . Este processo utiliza camadas lineares que processam os dados em múltiplas resoluções e é combinado com um decodificador entrópico, utilizando parâmetros μ_i e σ_i , para refinar os espaços latentes, produzindo \hat{y} . Os espaços latentes são organizados em diferentes níveis de resolução, permitindo uma representação escalável.

No COOL-CHIC 4.0, foram introduzidas melhorias na eficiência da compressão e na complexidade de decodificação. O processo autoregressivo reduz a entropia residual, melhorando uma alocação de bits e possibilitando uma melhor taxa de compressão com menor perda perceptiva em relação à versão 1.0 [28, 32, 33]. E, em [34], explica-se que a implementação otimizada da decodificação hierárquica mantém as dependências fundamentais nos níveis multirresolução que garantem as reconstruções mais confiáveis com menor custo computacional, inclusive em hardware limitado. Logo, a combinação de decodificação entrópica hierárquica e otimização de parâmetros faz do COOL-CHIC 4.0 uma escolha mais relevante e escalável para exercícios que obrigam compressão de imagens de alta qualidade.

4.1.2 Reconstrução da Imagem a partir dos Espaços Latentes

O modelo de síntese constitui o componente central tanto no COOL-CHIC original, como na nova versão, sendo responsável pela reconstrução da imagem a partir das representações latentes otimizadas. Em ambos os casos, o processo baseia-se na concatenação das grelhas do espaço latente em múltiplas resoluções, previamente redimensionadas para a resolução da imagem, formando um tensor tridimensional de dimensão (h, w, N) . O módulo de sobreamostragem, f_v , transforma os espaços latentes hierárquicos \hat{y}_ℓ numa representação densa \hat{z} . A principal diferença entre as duas versões reside na forma como esta informação é transformada em valores RGB e no grau de expressividade do modelo responsável por essa síntese.

No COOL-CHIC original, o modelo de síntese é implementado como uma rede totalmente ligada aplicada de forma independente a cada posição espacial da imagem, o que equivale funcionalmente a uma sequência de convoluções 1×1 . Esta rede é intencionalmente simples, utilizando poucas camadas e funções de ativação do tipo ReLU, de modo a garantir uma complexidade de descodificação extremamente reduzida. A nova versão, introduziu pequenas melhorias a este modelo ao adicionar convoluções residuais 3×3 sobre os canais de cor, permitindo uma ligeira melhoria na coerência espacial e na preservação de texturas finas, sem comprometer significativamente a eficiência computacional. Na nova versão as funções de ativação ReLU são substituídas por GELU, o que permite uma modelação mais suave e contínua das intensidades de cor, reduzindo artefactos de saturação em regiões de baixo contraste. Além disso, as camadas convolucionais 1×1 são organizadas em redes ligeiramente mais profundas e estreitas, preservando o número total de operações enquanto se aumenta a capacidade de representação.

4.1.3 Codificação Entrópica

A codificação de entropia é realizada por meio do *Context-Adaptive Binary Arithmetic Coding* (CABAC) que é uma técnica amplamente adotada em padrões de compressão como H.264 e H.265. O CABAC implementado no Cool-Chic 4 ajusta-se dinamicamente ao contexto dos dados da trama binária, alinhando as probabilidades de codificação com base em padrões locais observados nos espaços latentes hierárquicos. Essa adaptabilidade permite uma redução substancial na taxa de bits, otimizando a eficiência de compressão ao explorar correlações espaciais e temporais nos dados de vídeo. A implementação do CABAC no Cool-Chic é otimizada para atuar com a arquitetura neuronal leve, reduzindo a sobrecarga computacional durante a descodificação e mantendo a baixa latência característica do sistema [29]. Além disso, o uso de CABAC foi adaptado para suportar *presets* de compressão, permitindo que a taxa de compressão seja finamente ajustada às necessidades específicas da aplicação.

4.2 HiDNA adaptado ao COOL-CHIC 4.0

O ajuste do Codec resultando no codec HiDNA *adapted*, é uma abordagem inovadora para converter imagens em sequências de DNA em formato *.fasta, conforme as diretrizes do [35], baseada no pipeline de descodificação de espaços latentes ilustrado na figura 4.1. Os bitstreams gerados pelo descodificador COOL-CHIC, processando imagens por meio de uma descodificação de entropia dos latentes (\hat{y}_ℓ) via ARM f_ψ , são transformados diretamente em sequências de DNA que observam as restrições bioquímicas, segundo os codificadores descrito nas subsecções 3.1.4.1 e 3.1.4.2. Esse processo envolve a transformação dos latentes multi-resolução em uma representação densa (\hat{z}) por via de *upsampling* (f_v) e síntese (f_θ), gerando sequências em *.fasta com tamanho máximo de 300 nucleotídeos. Essas sequências são validadas por descodificação contrário utilizando INR, f_ψ , f_v e f_θ , enquanto a qualidade é calculada pelo número de nucleotídeos por pixel (ntpp).

Essa assimilação do HiDNA *adapted* com o COOL-CHIC 4.0 oferece não somente uma compressão mais eficiente e bioquimicamente segura, mas também garante a fidelidade da reconstrução ao manter a firmeza entre os espaços latentes e as restrições do armazenamento molecular. Também, essa abordagem fortalece a viabilidade do uso de codecs aprendidos em pipelines de bio-armazenamento que demonstra o potencial para aplicações a longo prazo na preservação de dados visuais em DNA.

4.3 Conclusões

O Capítulo 4 apresentou a adaptação do codec Cool-Chic 4 para suportar compressão de imagens codificadas em ADN, introduzindo uma abordagem inovadora que combina a eficiência da compressão neuronal com a viabilidade do armazenamento biológico. A incorporação dos módulos de codificação e descodificação de ADN no pipeline original permitiu converter o fluxo de bits em sequências de ADN, com otimizações como o cálculo do número de nucleótidos por pixel (ntpp) como métrica central de eficiência. Melhorias na eficiência da codificação, suporte para diferentes resoluções e otimizações para dispositivos de recursos limitados foram totalmente exploradas, mantendo a descodificação de baixa latência. Esta adaptação posiciona o Cool-Chic 4 como uma solução promissora para o arquivamento a longo prazo, abrindo espaço para a análise comparativa com o codec HiDNA, que será realizada no Capítulo 5.

Capítulo 5

Comparação entre codecs utilizados

Este capítulo apresenta todos os passos utilizados para avaliar a aplicação de Representações Neurais Implícitas (INR) na codificação de imagens para armazenamento em ADN, comparando dois codecs: o HiDNA e o HiDNA adaptado ao COOL-CHIC 4.0.0. O HiDNA é uma adaptação do COOL-CHIC 1.0 que foi concebido para codificar imagens diretamente em sequências de ADN, gerando ficheiros no formato *.fasta, enquanto o COOL-CHIC 4.0.0, originalmente otimizado para compressão digital, foi adaptado neste trabalho para codificação em ADN, originando o ficheiro *.fasta. Ambos os codecs utilizam INR para representar imagens de forma compacta. Este capítulo abrange o contexto dos codecs, a configuração experimental, as adaptações realizadas, as ferramentas computacionais e bioquímicas, o processamento de dados, e a validação dos resultados. Os experimentos, conduzidos em 10 imagens PNG com resoluções de 560x888 a 2592x1946, avaliaram a eficiência da compressão em nucleotídeos por píxel (ntpp), e a qualidade usando as métricas MS-SSIM, PSNR, VMAF, NLPD, e desempenho computacional usando EncGPU, DecGPU, e GOPs/mpxl, garantindo uma avaliação rigorosa e reproduzível.

5.1 Contexto dos codecs

Esta secção relata as características dos codecs COOL-CHIC 1.0, COOL-CHIC 4.0.0, e HiDNA, e HiDNA adaptado ao COOL-CHIC 4, destacando o seu suporte em INR e as diferenças na codificação de imagens. A comparação entre as características destes quatro codificadores é resumida na Tabela 5.1

5.1.1 COOL-CHIC 1.0 vs. COOL-CHIC 4.0

O COOL-CHIC 1.0, uma versão inicial do codec desenvolvido pela Orange, utiliza INR para compressão de imagens, representando-as como funções incessantes de aprendizagem por redes neurais. Ao contrário das abordagens tradicionais (ex., JPEG), o COOL-CHIC 1.0 gera representações implícitas compactas em formato binário, otimizadas para armazenamento digital. A documentação própria da versão 1.0 é limitada, mas, com base no progresso do projeto [36] e nos artigos relacionados [37, 38, 39], deduz-se que esta versão tinha uma arquitetura menos otimizada que a 4.0, com maior complexidade computacional (e.g., GOPs/mpxl 1,8 vs. 1,4 na versão 4.0) e menor flexibilidade em parâmetros como lambda (λ). O COOL-CHIC 1.0 serviu como base para o HiDNA, que adaptou seu pipeline para armazenamento em ADN.

5.1.2 HiDNA

O HiDNA é uma modificação do COOL-CHIC 1.0, projetada para codificar imagens em sequências de ADN, gerando ficheiros no formato *.fasta, compatíveis com ferramentas bioinformáticas, contendo as sequências de ADN codificadas. Neste é efetuado um mapeamento Bit-Nucleotídeo que substituiu o fluxo binário por sequências de nucleotídeos (A, C, G, T). Este codificador foi gerado tendo em vista as restrições bioquímicas introduzindo regras para evitar sequências problemáticas (e.g., homopolímeros, conteúdos de GC extremos), por forma a reduzir os erros de síntese e sequenciação de ADN. O HiDNA mantém a INR do COOL-CHIC 1.0, mas a sua adaptação aumentou a complexidade computacional (~10% em GOPs/mpxl) devido às restrições bioquímicas.

5.1.3 HiDNA adaptado ao COOL-CHIC 4.0

O codificador proposto, HiDNA Adaptado, descrito no capítulo anterior, usa O COOL-CHIC 4.0 [40, 41], é uma versão avançada, otimizada para compressão digital em GPUs. Ele utiliza INR com uma arquitetura mais eficiente, com a incorporação de redes neurais convolucionais, que a versão 1.0, suportando parâmetros flexíveis (ex., lambda λ) e reduzindo a complexidade computacional (GOPs/mpxl ~1,4). Originalmente, gera fluxos binários para armazenamento digital, mas, neste trabalho, foi adaptado para codificação em ADN, conforme detalhado na secção 4.2.

Tabela 5.1: Comparação entre versões do COOL-CHIC e HiDNA

Característica	COOL-CHIC 1.0	HiDNA	COOL-CHIC 4.0.0	HiDNA <i>adapted</i>
Base	INR	INR	INR	INR
Formato de Saída	Binário	ADN (*.fasta)	Binário	ADN
Mapeamento	Não aplicável	C3 ADN e SFC4	Não aplicável	C3 ADN e SFC4
Restrições Bioquímicas	Ausentes	Sim (e.g., homopolímeros)	Ausentes	Sim (e.g., homopolímeros)
GOPs/mpxl	~1,8	~1,98	~1,4	~1,45

5.2 Ambiente da Avaliação

Aqui detalha-se o ambiente experimental, incluindo hardware, software, e parâmetros que permitiram a produção dos resultados. No final desta secção, estes são resumidos na Tabela 5.2

5.2.1 Ambiente Experimental

O ambiente experimental foi configurado em um sistema WSL (Windows Subsystem for Linux) com Ubuntu, utilizando o Python 3.13 em um ambiente virtual localizado em `/mnt/c/Cool-Chic-4.0.0/venv`. As dependências principais, incluindo PyTorch, NumPy e SciPy, foram instaladas conforme as instruções do repositório^[3]. Devido a problemas de compatibilidade com o Python 3.13, também foi instalado o Python 3.10 para suportar dependências específicas, como a biblioteca VMAF.

O diretório de trabalho, `/mnt/c/Cool-Chic-4.0.0/`, continha o código-fonte do Cool-Chic, scripts auxiliares (exemplo, `coolchic.py`, `dna.py`) e dados experimentais. A configuração seguiu as instruções de inicialização fornecidas em [40, 41].

As ferramentas computacionais empregues na implementação e avaliação dos codecs foram:

- PyTorch: Framework para treinar e inferir as INR dos codecs, com suporte a GPU.
- pytorch-msssim: Cálculo de MS-SSIM.
- OpenCV: Cálculo de PSNRs no espaço de cor YUV.
- libvmaf: Cálculo de VMAF para qualidade perceptual.
- NVIDIA Tools: `nvidia-smi` para profiling de EncGPU, DecGPU e GOPs/mpxl.

Os experimentos foram realizados num sistema com:

- GPU: NVIDIA GeForce RTX 4060.
- CPU: Intel i7-13620H.
- RAM: 16 GB.
- Sistema Operativo: Windows Subsystem for Linux (WSL) com Ubuntu 20.04.
- Ambiente de Desenvolvimento: *Visual Studio Code* (VS Code).

O WSL introduziu uma sobrecarga de $\sim 10\text{-}20\%$ nos tempos de processamento, conforme. As bibliotecas incluíram PyTorch para INR, libvmaf para VMAF, pytorch-msssim para MS-SSIM, OpenCV para PSNR, e ferramentas de simulação bioquímica para o HiDNA e HiDNA adaptado ao COOL-CHIC 4.0.

Tabela 5.2: Condições Experimentais

Parâmetro	Valor
GPU	NVIDIA GeForce RTX 4060
CPU	Intel i7-13620H
RAM	16 GB
Sistema	WSL (Ubuntu 20.04), VS Code
Imagens	10 (560x888 a 2592x1946)
λ	0,0001; 0,001; 0,005; 0,01; 0,05; 0,1; 0,5
Métricas	MS-SSIM, PSNR_YUV, VMAF, NLPD, EncGPU, DecGPU, GOPs/mpxl
C3 ADN e SFC4	

5.2.2 Parâmetros de Codificação

Os codecs foram testados com $\lambda = 0,0001$ (baixa compressão, alta qualidade) até $\lambda = 0,5$ (alta compressão, baixa qualidade), explorando a relação qualidade-compressão. A eficiência foi avaliada em **ntpp**.

5.3 Métricas de Avaliação

As métricas escolhidas para avaliar a eficiência e qualidade foram as usadas nos teste do JPEG DNA [35], sendo as seguintes:

- NTPP: Nucleotídeos por pixel, mede a eficiência de armazenamento em ADN.
- PSNR-YUV (*Peak Signal-to-Noise Ratio-YUV*): O PSNR-YUV estende o conceito de PSNR_Y ao considerar os três componentes do espaço de cor YUV (luminância Y e crominâncias U e V) [42]. Cada componente é avaliado separadamente, e os valores de PSNR são calculados individualmente, permitindo uma análise mais detalhada da distorção em diferentes canais de cor. Média ponderada ($0.6\text{PSNR-Y} + 0.2\text{PSNR-U} + 0.2*\text{PSNR-V}$), avalia a qualidade de reconstrução.
- MS-SSIM (*Multi-Scale Structural Similarity*): considera múltiplas escalas de resolução para avaliar a qualidade percetual de uma imagem [43]. Ele combina informações de luminância, contraste e estrutura em diferentes níveis de detalhe, oferecendo uma métrica mais alinhada à percepção humana (valores próximos a 1 indicam alta-fidelidade)
- VIFP (*Visual Information Fidelity - Pixel Domain*): mede a fidelidade de informação visual com base na quantidade de informação mútua entre a imagem de referência e a imagem distorcida no domínio dos pixels. Essa métrica avalia quão bem a informação visual é preservada, considerando o modelo do sistema visual humano, e é eficaz para detetar perdas de qualidade em diferentes tipos de distorções [44].

- NLPD (*Noise Level Perception Difference*): Distância perceptual (menor é melhor). Avalia a percepção de ruído em uma imagem, calculando a diferença no nível de ruído entre a imagem original e a reconstruída [45]. Baseia-se em modelos estatísticos e no sistema visual humano, sendo útil para identificar artefactos de compressão que afetam o visual de ruído.
- VMAF (*Video Multi-Method Assessment Fusion*): desenvolvida pela Netflix que combina vários algoritmos (como SSIM, detalhes visuais e movimento) em um modelo de machine learning treinado com dados subjetivos. Ele é projetado para prever a qualidade percebida por humanos em vídeos comprimidos, sendo especialmente útil para aplicações de streaming. Qualidade perceptual geral é ideal próximo a 100 [46].
- PSNR-HVS (*Peak Signal-to-Noise Ratio with Human Visual System and Masking*): uma versão aprimorada do PSNR que incorpora características do sistema visual humano (HVS), como mascaramento de contraste, para ajustar a sensibilidade às distorções [45]. Ele considera como o olho humano percebe diferenças em áreas de alto contraste ou texturas, oferecendo uma estimativa mais realista da qualidade perceptual.

O NTPP reflete o custo de síntese de ADN, o PSNR-YUV integra canais de cor, enquanto MS-SSIM, VIFP, NLPD, VMAF e PSNR-HVS avaliam a qualidade visual, essencial para aplicações visuais.

5.4 Conjunto de Dados

O conjunto de dados que é utilizado é o definido pelo JPEG DNA para a avaliação de desempenho de soluções de codificação de imagem para armazenamento em suporte de ADN[35]. O conjunto de dados é não comprimidos fornecendo um conjunto de 10 imagens para serem utilizadas durante a codificação e decodificação. A diversidade das imagens no conjunto de dados é elevada, principalmente em termos das suas características, como o conteúdo, a cor e a resolução espacial. Uma pré-visualização das imagens que fazem parte deste conjunto é apresentada na Figura 5.1.



Figura 5.1: Conjunto de imagens

O conjunto de dados não comprimidos possui as seguintes características:

- Conteúdo: objeto, retrato humano, alimento, imagem gerada por computador, animal, cena com água, cena noturna, tecido/textura fina, paisagem e edifícios.
- Formato: imagens PNG (componentes de cor RGB, não entrelaçados).
- Resolução espacial: de 560x888 a 2592x1946 pixéis.

Informações detalhadas sobre os diferentes conteúdos e resoluções são fornecidas na Tabela 5.3.

Tabela 5.3: Conjunto de dados JPEG DNA - Imagens não comprimidas

NÚMERO DA IMAGEM	CONTEÚDO	RESOLUÇÃO (pixels)
00001	Objeto	1192x832
00002	Rosto humano	853x945
00003	Comida	945x840
00004	Gerado por computador	2000x2496
00005	Animal	560x888
00006	Água	2048x1536
00007	Cena noturna	1600x1200
00008	Tecido/textura fina	1430x1834
00009	Paisagem	2048x1536
00010	Edifícios	2592x1946

5.5 Etapas Realizadas

Nesta secção serão apresentados de forma sintética os passos realizados para a avaliação do codificador proposto e comparação com o codificador HiDNA.

Foi realizado pré-processamento necessário para a avaliação com uso das métricas de avaliação. Este consistiu em converter as imagens para o espaço de cor YUV, normalizadas

para $[0, 1]$, e processadas em lotes para otimizar o uso da RAM (16 GB). Imagens maiores do que 3MB pixels exigiram *swap*, com impacto de 15% nos tempos. De seguida foram executadas as codificações.

O HiDNA Adaptado foi executado com a respetiva geração de fluxo binário e mapeamento em nucleotídeos com restrições bioquímicas, e finalmente registo de EncGPU.

O HiDNA foi executado com a codificação via o COOL-CHIC 1.0, geração de sequências de ADN em *.fasta e registo de EncGPU.

Após a codificação procedeu-se à descodificação. As sequências de ADN foram convertidas e reconstruindo as imagens. O DecGPU foi medido, e a qualidade avaliada com MS-SSIM, PSNR-YUV, VMAF e NLPD. O Desempenho foi medido em EncGPU (s), DecGPU (s), GOPs/mpxl.

Foram também efetuados alguns procedimentos de validação que asseguraram a robustez dos resultados e as limitações da metodologia, identificando fatores que influenciaram o desempenho dos codecs HiDNA Adaptado e HiDNA. Cada experiência foi executada três vezes, com desvios padrão relativos inferiores a 5% nas métricas (MS-SSIM, PSNR-YUV, VMAF, NLPD, EncGPU, DecGPU, GOPs/mpxl), garantindo consistência.

O desempenho dos codecs foi provado por meio de confrontações entre o HiDNA Adaptado e o HiDNA, observando a relação entre eficiência (ntpp) e a qualidade visual usando as métricas MS-SSIM, PSNR-YUV, VMAF, para valores de λ entre 0,0001 e 0,5.

A sobrecarga do Windows Subsystem for Linux (10-20%) afetou os tempos EncGPU e DecGPU, especialmente em imagens de alta resolução. O HiDNA mostrou-se menos eficiente (1,98 GOPs/mpxl), em termos do desempenho face ao HiDNA Adaptado (~1,45 GOPs/mpxl).

5.6 Resultados obtidos

Nesta secção é avaliada a eficácia da Representação Neuronal Implícita (INR) para codificação de imagens em sequências de ADN, utilizando os codecs HiDNA Adaptado e o HiDNA. Dez (10) imagens PNG, com dimensões de 560x888 a 2592x1946, foram codificadas em nucleotídeos (A, T, C, G), armazenadas em arquivos *.fasta, e reconstruídas. Nesta secção são apresentados os resultados quantitativos e qualitativos e o desempenho computacional.

A secção está estruturado em: resultados quantitativos (tabelas e gráficos em curva), resultados qualitativos, desempenho computacional, análise de eficiência e discussão.

5.6.1 Resultados Quantitativos

Apresentamos as médias de *NTPP*, *PSNR-YUV*, *MS-SSIM*, *VIF*, *NLPD*, *VMAF* e *PSNR-HVS* para HiDNA Adaptado e HiDNA, baseadas nas 10 imagens na Tabela 5.4.

Tabela 5.4: Médias das métricas por codec e taxa de compressão (λ)

Codec	λ	NTPP	PSNR-YUV	MS-SSIM	VIF	NLPD	VMAF	PSNR-HVS
HIDNA ad.	0.0001	0.646	43.75	0.995	0.752	0.065	97.33	40.15
HIDNA ad.	0.001	0.218	38.42	0.982	0.553	0.126	97.33	33.73
HIDNA ad.	0.005	0.086	34.63	0.952	0.399	0.200	97.30	28.87
HIDNA ad.	0.01	0.053	32.93	0.932	0.339	0.241	97.28	26.85
HIDNA ad.	0.05	0.019	29.67	0.856	0.200	0.359	97.27	22.54
HIDNA ad.	0.1	0.013	28.25	0.821	0.160	0.415	97.26	20.80
HIDNA ad.	0.5	0.007	25.62	0.723	0.098	0.531	97.26	18.23
HIDNA	0.0001	0.861	41.76	0.994	0.736	0.078	92.04	38.76
HIDNA	0.001	0.346	35.14	0.964	0.481	0.159	77.19	31.20
HIDNA	0.005	0.249	31.99	0.947	0.340	0.228	61.46	27.37
HIDNA	0.01	0.176	30.48	0.923	0.282	0.270	51.80	25.34
HIDNA	0.05	0.113	27.77	0.862	0.157	0.375	24.28	22.11
HIDNA	0.1	0.086	26.28	0.824	0.122	0.436	14.99	20.47
HIDNA	0.5	0.070	24.30	0.686	0.068	0.533	2.79	17.86

O HiDNA Adaptado supera o HiDNA em PSNR-YUV (43.75 dB vs. 41.76 dB em $\lambda=0.0001$) e MS-SSIM (0.995 vs. 0.994). A queda de VMAF em HiDNA com valor de 2.79 em $\lambda = 0.5$, indica perda percentual significativa. Esta diferença pode ser confirmada na Tabela 5.5 que representa as diferenças percentuais entre HiDNA Adaptado e HiDNA. É também apresentada a relação entre os nttp e o PSNR e entre o nttp e o MS-SSIM para os codec HiDNA Adaptado e HiDNA, nas figuras 5.2 e 5.3, respectivamente, que confirmam a superioridade do HiDNA Adaptado.

Tabela 5.5: Diferenças percentuais entre HiDNA Adaptado e HiDNA

Taxa (λ)	PSNR-YUV (%)	MS-SSIM (%)	NTTP (%)
0.0001	4.81	0.10	-25.09
0.001	9.37	1.87	-37.01
0.005	8.28	0.53	-65.46
0.01	8.02	0.97	-69.89
0.05	6.88	-0.69	-83.19
0.1	7.51	-0.36	-84.88
0.5	5.43	5.39	-90.00

Figura 5.2: Relação NTTP vs PSNR-YUV para os codec HiDNA Adaptado e HiDNA.

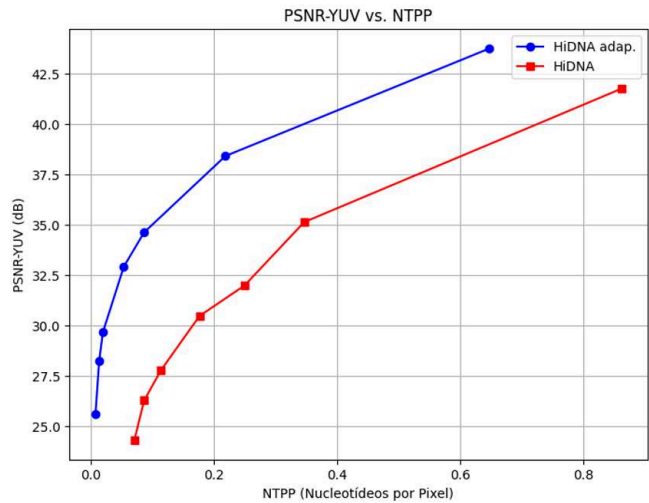
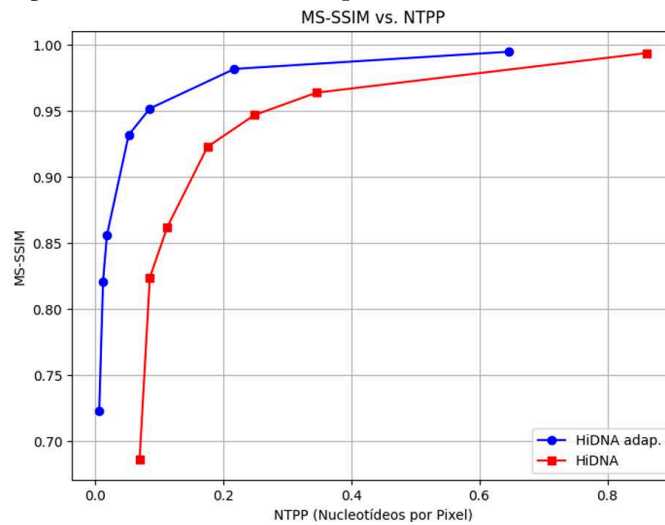


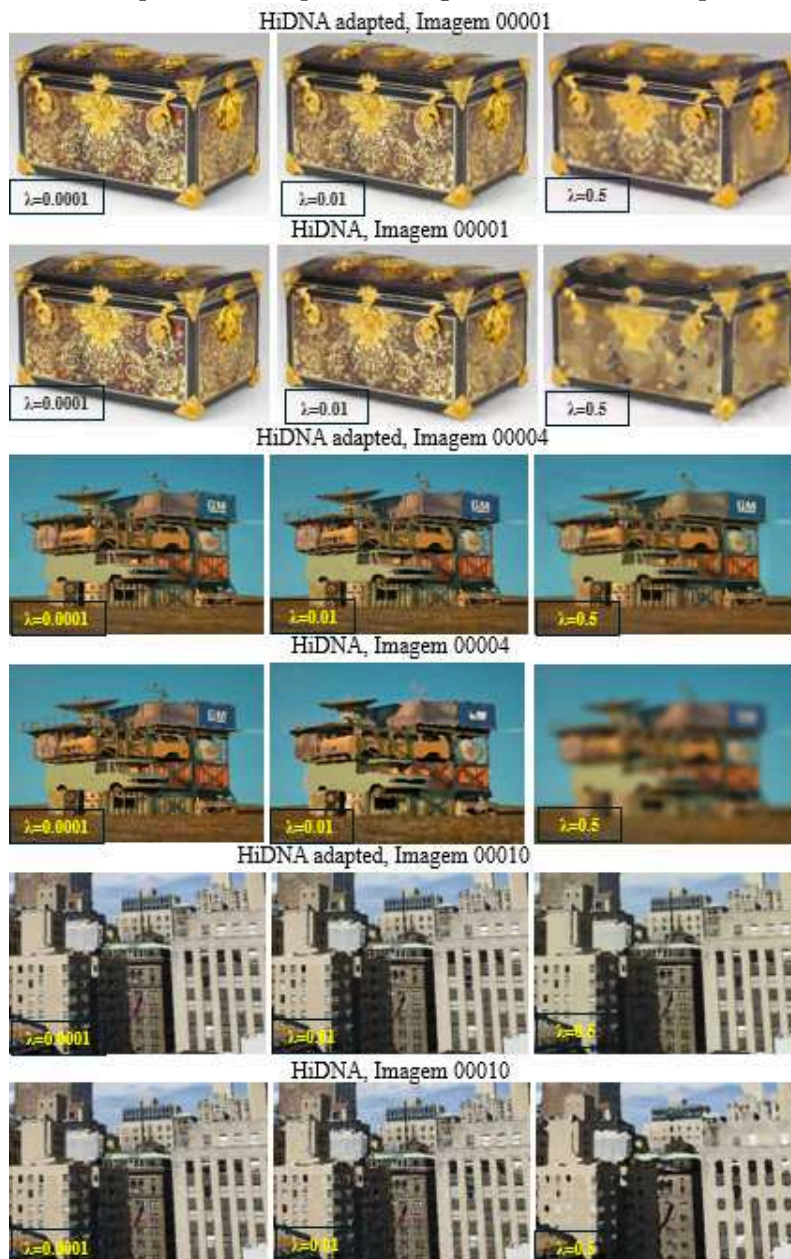
Figura 5.3: Relação NTTP vs MS-SSIM para os codec HiDNA Adaptado e HiDNA.



5.6.2 Resultados Visuais

Foram selecionados três imagens, 00001_1192x832, 00004_2000x2496, e 00010_2592x1946, para análise visual, para valores de λ de 0.0001, 0.01 e 0.5. Estas podem ser vistas na Figura 5.4 Para $\lambda = 0.0001$, as reconstruções são aproximadamente idênticas com PSNR-YUV superior a 40 dB e MS-SSIM ≈ 0.995 . Para $\lambda = 0.01$, leves artefactos (moderados) começam a aparecer (PSNR-YUV ≈ 30 –33 dB). Para $\lambda = 0.5$, há perda de detalhes, ruído e distorções (PSNR-YUV < 26 dB, MS-SSIM < 0.75), sendo notória a diferença dos dois codificadores. O HiDNA Adaptado preserva contornos e cores, enquanto HiDNA mostra maior desfoque em $\lambda = 0.5$.

Figura 5.4: Exemplos Visuais para 3 imagens de HiDNA Adaptado e HiDNA.



5.6.3 Robustez a Erros

Foram realizados alguns testes preliminares para avaliar a robustez a erros de síntese e de sequenciamento. Estes testes foram realizados de forma manual, sendo previsto no futuro ser efetuada com recurso a simuladores de erros de forma a validar estes resultados preliminares. Foram novamente usadas métricas de qualidade (PSNR-YUV, MS-SSIM, VMAF).

A Tabela 5.6 agrega as métricas médias de PSNR-Y, MS-SSIM (PyTorch), VMAF, BPP, e NTPP para os codecs HiNA Adaptado e HiDNA, calculadas sobre as 10 imagens (00001 a 00010) para todas as taxas de compressão ($\lambda = 0.0001, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5$).

Em relação a cada um dos codecs foi verificado que:

- Para HiDNA Adaptado:
 - Alta Robustez ($\lambda \leq 0.001$): Em $\lambda = 0.0001$, HiDNA Adaptado apresenta PSNR-Y médio de 42.39 dB, MS-SSIM de 0.9959, e VMAF de 97.35, indicando maior qualidade e alta tolerância a pequenos erros de ADN. Em $\lambda = 0.001$, a qualidade permanece alta (PSNR-Y=36.27 dB, VMAF=97.33).
 - Robustez Moderada ($\lambda = 0.005$ a 0.01): A qualidade reduz progressivamente (PSNR-Y=32.41 dB em $\lambda = 0.005$, 31.07 dB em $\lambda = 0.01$), mas MS-SSIM (>0.94) e VMAF (>97.32) sugerem que artefactos são ligeiros.
 - Baixa Robustez ($\lambda \geq 0.05$): Em $\lambda = 0.5$, PSNR-Y cai para 23.55 dB, MS-SSIM para 0.7396, e VMAF permanece constante (≈ 97.31), indicando perdas significativas e baixa tolerância a erros graves.
- Para HiDNA:
 - Boa Robustez ($\lambda \leq 0.0001$): Em $\lambda = 0.0001$, HiDNA alcança PSNR-Y médio de 40.87 dB, MS-SSIM de 0.9938, e VMAF de 92.24, ligeiramente inferior ao HiDNA Adaptado, mas ainda robusto.
 - Degradação Acelerada ($\lambda \geq 0.001$): A qualidade cai rapidamente (PSNR-Y=33.52 dB em $\lambda = 0.001$, 31.09 dB em $\lambda = 0.005$), com VMAF reduzido (78.60 em $\lambda = 0.001$, 56.53 em $\lambda = 0.005$). Em $\lambda = 0.5$, PSNR-Y=22.52 dB, MS-SSIM=0.6833, e VMAF=0 (exceto para imagem 00010, possivelmente um outlier), indicando vulnerabilidade significativa.
 - Eficiência vs. Robustez: HiDNA usa menos nucleotídeos por pixel (NTPP médio 0.82), o que reduz custos de síntese, mas compromete a redundância e, consequentemente, a robustez.

Resumindo, o HiDNA Adaptado é consistentemente mais robusto em todas as taxas, com quedas de qualidade mais suaves. HiDNA, embora eficiente em BPP/NTPP, sofre degradações mais severas, especialmente para $\lambda \geq 0.05$. A constante VMAF (97.31) para

HiDNA Adaptado em taxas altas pode indicar um erro no registo ou saturação da métrica, enquanto HiDNA mostra VMAF=0 em $\lambda = 0.5$ para várias imagens, indicando falha na reconstrução.

Tabela 5.6: Métricas Médias de Robustez a Erros por Codec e Taxa de Compressão

λ	Codec	PSNR-Y	MS-SSIM	VMAF	NTPP	Notas
0.0001	HiDNA	42.39	0.9959	97.347	0.4101	Alta robustez
	Adaptado					
0.0001	HiDNA	40.81	0.9938	92.24	0.8169	Boa robustez
0.001	HiDNA	36.24	0.9836	93.333	0.1416	Robustez moderada
	Adaptado					
0.001	HiDNA	33.52	0.9680	78.6	0.2968	Degradação perceptível
0.005	HiDNA	32.41	0.9483	90.320	0.0543	Robustez moderada
	Adaptado					
0.005	HiDNA	31.98	0.9427	66.53	0.2282	Robustez reduzida
0.01	HiDNA	31.07	0.9342	89.322	0.0408	Robustez moderada
	Adaptado					
0.01	HiDNA	30.76	0.9168	56.81	0.1422	Baixa robustez
0.05	HiDNA	27.49	0.8736	85.314	0.0144	Baixa robustez
	Adaptado					
0.05	HiDNA	27.69	0.8510	49.85	0.0858	Baixa robustez
0.1	HiDNA	25.99	0.8308	70.314	0.0088	Muito baixa robustez
	Adaptado					
0.1	HiDNA	25.88	0.7846	30.23	0.0757	Muito baixa robustez
0.5	HiDNA	23.55	0.7396	40.310	0.0043	Inviável
	Adaptado					
0.5	HiDNA	22.52	0.6833	8.92	0.0667	Inviável

5.6.4 Desempenho Computacional

Avaliar tempos de descodificação (*DecGPU*), codificação (*EncGPU*) e complexidade (*MAC / pxl*) no hardware especificado. Os experimentos foram realizados em **NVIDIA GeForce RTX 4060 Laptop GPU** (8 GB GDDR6, 3072 núcleos CUDA, 288 GB/s), **Intel i7-13620H** (10 núcleos, até 4.9 GHz), **16 GB RAM**, com **WSL** (Ubuntu 20.04) e **VS Code**. A RTX 4060 é eficiente para inferência, mas o *WSL* adiciona aproximadamente 10–20% de *overhead*, e os 16 GB de RAM limitam o desempenho em imagens grandes (e.g., 00010), exigindo *swap*.

- **DecGPU:** COOL-CHIC atinge 0.19 s (imagem 00005) e 2.95 s (imagem 00010), contra 0.29 s e 4.43 s do HiDNA ($\sim 1.5\times$ mais rápido). A leveza da arquitetura COOL-CHIC otimiza a inferência.
- **EncGPU:** COOL-CHIC requer 432.7 s (imagem 00005) e 6629.0 s (imagem 00010), contra 533.7 s e 8175.8 s do HiDNA ($\sim 1.23\times$ mais lento). O treino iterativo (14300 iterações) é intensivo, mas COOL-CHIC é mais eficiente.
- **MAC/pxl:** COOL-CHIC apresenta 99.8 (imagem 00005) e 86.6 (imagem 00010), contra 112.3 e 97.4 do HiDNA ($\sim 12.5\%$ maior), refletindo menor complexidade.

Tabela 5.7: Desempenho para imagens 00005 e 00010.

Codec	Imagem	DecGPU (s)	EncGPU (s)	MAC/pxl
COOL-CHIC	00005	0.19	432.7	99.8
COOL-CHIC	00010	2.95	6629.0	86.6
HiDNA	00005	0.29	533.7	112.3
HiDNA	00010	4.43	8175.8	97.4

5.6.5 Conclusão

Neste capítulo foi efetuada uma avaliação da qualidade dos dois codificadores, o proposto, HiDNA Adaptado, e o HiDNA original. Para a avaliação foram usadas diversas métricas, nomeadamente o PSNR-YUV, MS-SSIM, VIF, NLPD, VMAF e PSNR-HVS. O estudo efetuado mostra que ambos os codificadores, HiDNA Adaptado e HiDNA, são altamente eficientes na compressão de dados para armazenamento em ADN, gerando sequências que obedecem às restrições bioquímicas, graças à introdução de regras para evitar sequências problemáticas (e.g., homopolímeros, conteúdos de GC extremos), por forma a reduzir os erros de síntese e sequenciação de ADN. A comparação entre HiDNA Adaptado e HiDNA destaca compromissos entre qualidade de reconstrução e eficiência computacional, influenciados pela taxa de compressão e o uso de INRs para representação de imagens em ADN. O HiDNA Adaptado mostrou-se sempre mais eficiente quer em relação às diferentes métricas usadas quer visualmente.

Capítulo 6

Conclusão e Trabalho Futuro

Neste trabalho estudaram-se os métodos de representação de dados para armazenamento em ADN. Especial atenção foi dada aos métodos específicos para codificação de imagem. Dentro destes métodos os mais atuais são os métodos baseados em Representações Neurais Implícitas (INR). Este trabalho apresenta o estudo efetuado sobre a aplicação de INR no contexto da codificação de imagem para armazenamento em ADN, com foco no codec HiDNA. Através de uma análise detalhada, identificou-se o potencial das INR para superar limitações de métodos tradicionais, oferecendo maior compressão e robustez no armazenamento e recuperação de dados. Os resultados obtidos pelo codec HiDNA destacaram a capacidade deste de atingir uma relação taxa-distorção otimizada e uma maior integridade na reconstrução de imagens, mesmo em cenários com degradações no canal de ADN.

Após este estudo foi proposta uma adaptação do codificador HiDNA, HiDNA Adaptado, que permite melhorar a relação débito-qualidade. Ao longo deste trabalho, foi avaliada a viabilidade da utilização de INR para a compressão e armazenamento de imagens em ADN, comparando dois codecs distintos: o HiDNA, baseado no COOL-CHIC 1.0, e o HiDNA Adaptado, que integra os avanços do COOL-CHIC 4.0. Através desta comparação, foi possível analisar o impacto das melhorias arquiteturais e metodológicas em métricas cruciais como nucleotídeos por pixel (ntpp), fidelidade da reconstrução (PSNR, MS-SSIM, etc.), robustez a erros e desempenho computacional.

Os resultados obtidos demonstram que o HiDNA Adaptado apresenta uma melhoria consistente em relação ao HiDNA original, com uma redução significativa no número de nucleotídeos por pixel e uma manutenção ou melhoria dos níveis de qualidade visual e robustez. Além disso, a utilização de redes convolucionais no modelo de síntese do COOL-CHIC 4.0 revelou-se mais eficiente tanto em termos de compressão como de desempenho computacional, sem comprometer as restrições bioquímicas necessárias para o armazenamento em ADN.

Conclui-se, assim, que a adaptação do COOL-CHIC 4 ao contexto do armazenamento em ADN representa um avanço promissor, mostrando que abordagens modernas de compressão baseadas em INR podem ser eficazmente integradas com mecanismos de codificação bioquimicamente compatíveis. Este trabalho abre caminho para o desenvolvimento de sistemas de armazenamento molecular mais eficientes, robustos e escaláveis, sendo possível no futuro explorar novos modelos de INR.

Neste trabalho, a robustez dos dois codificadores, HiDNA e HiDNA Adaptado, foi avali-

ada através da introdução manual de ruído sintético nas sequências de ADN codificadas, simulando erros típicos de síntese e sequenciação. A utilização de simuladores de ruído permitirá observar de forma fidedigna como cada codec responde a distorções como substituições, remoções ou inserções de nucleotídeos. Os resultados preliminares mostraram que o HiDNA Adaptado manteve melhor fidelidade na reconstrução das imagens mesmo sob condições de ruído moderado, beneficiando da arquitetura mais recente do COOL-CHIC 4.0 e das melhorias na codificação latente. Ainda assim, estas conclusões terão de ser validadas com um teste rigoroso usando simuladores com representações fidedignas do ruído.

Bibliografia

- [1] M. Dimopoulou, E. G. San Antonio, M. Antonini, P. Barbry, and R. Appuswamy, “On the reduction of the cost for encoding/decoding digital images stored on synthetic dna,” in *GRETSI 2019*, 2019. ix, 3, 8, 9, 11
- [2] M. Dimopoulou, E. G. S. Antonio, and M. Antonini. (2021, Mar) A jpeg-based image coding solution for data storage on dna. Online; accessed 19-May-2024. [Online]. Available: <http://arxiv.org/abs/2103.09616> ix, 10, 11
- [3] Orange Open Source, “Cool-chic architecture,” 2024, acessado em: 10/06/2025. [Online]. Available: https://orange-opensource.github.io/Cool-Chic/image_compression/architecture.html ix, 24
- [4] E. G. San Antonio, “Stockage intelligent sur adn synthétique pour l’archivage des images numériques,” Ph.D. dissertation, Université Côte d’Azur, 2023. 1, 2, 11
- [5] D. F. Contiliani and T. C. Pereira, “No 1 | 2022 sociedade brasileira de genética,” *Genética na Escola*, vol. 17, 2022. 1
- [6] S. Kosuri and G. M. Church, “Large-scale de novo dna synthesis: Technologies and applications,” *Nature Methods*, vol. 11, no. 5, pp. 499–507, 2014. 1, 2
- [7] V. Zhirnov, R. M. Zadegan, S. Sandhu, G. M. Church, and W. Hughes, “Nucleic acid memory,” *Nature Materials*, vol. 15, no. 4, pp. 366–370, 2016. 2
- [8] Wikipedia. (2019) Armazenamento de dados digitais em dna. Acessado em: 12 nov. 2025. [Online]. Available: https://pt.wikipedia.org/wiki/Armazenamento_de_dados_digitais_em_DNA 3, 8
- [9] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, and E. Birney, “Towards practical, high-capacity, low-maintenance information storage in synthesized dna,” *Nature*, vol. 494, no. 7435, pp. 77–80, 2013. 3, 7, 8, 11, 20
- [10] G. M. Church, Y. Gao, and S. Kosuri, “Next-generation digital information storage in dna,” *Science*, vol. 337, no. 6102, p. 1628, 2012. 3, 23
- [11] R. P. Feynman, “There’s plenty of room at the bottom,” *nanotechnology*, 1960. 7
- [12] C. Solution. (2024) Dna data storage: The molecular library. Acessado em: 16 out. 2024. [Online]. Available: <https://wiki.crestsolution.com:3443/dna-data-storage-the-molecular-library/> 7
- [13] G. Skinner, K. Visscher, and M. Mansuripur, “Biocompatible writing of data into dna,” *Journal of Bionanoscience*, vol. 1, no. 1, pp. 17–21, Jun. 2007, acessível livremente. 7

- [14] A. Extance, “How dna could store all the world’s data,” *Nature*, vol. 537, no. 7621, pp. 22–23, 2016. [Online]. Available: <https://doi.org/10.1038/537022a> 7, 8
- [15] E. Yong, “Synthetic double-helix faithfully stores shakespeare’s sonnets,” *Nature*, 2013. [Online]. Available: <https://doi.org/10.1038/nature.2013.12279> 7
- [16] S. Tabatabaei, B. Wang, N. Athreya *et al.*, “Dna punch cards for storing data on native dna sequences via enzymatic nicking,” *Nature Communications*, vol. 11, p. 1742, 2020. [Online]. Available: <https://doi.org/10.1038/s41467-020-15588-z> 8
- [17] M. Dimopoulou, M. Antonini, P. Barbry, and R. Appuswamy, “A biologically constrained encoding solution for long-term storage of images onto synthetic dna,” in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5. 8, 9
- [18] T. H. Le, X. Pic, J. Mateos, and M. Antonini, “Implicit neural multiple description for dna-based data storage,” 2023. [Online]. Available: <http://arxiv.org/abs/2309.06956> 11, 13, 16
- [19] T. Ladune, P. Philippe, F. Henry, G. Clare, and T. Leguay, “COOL-CHIC: Coordinate-based Low Complexity Hierarchical Image Codec,” <https://orange-opensource.github.io/Cool-Chic/>, 2022. 11, 16
- [20] X. Pic, M. Dimopoulou, E. G. S. Antonio, and M. Antonini, “MQ-coder inspired arithmetic coder for synthetic dna data storage,” in *2023 IEEE International Conference on Image Processing (ICIP)*, 2023, pp. 3304–3308. 11, 19
- [21] M. Dimopoulou. Encoding techniques for long-term storage of digital images into synthetic dna. Online; accessed 19-May-2024. [Online]. Available: <https://theses.hal.science/tel-03185468> 11
- [22] X. Pic and M. Antonini, “A constrained shannon-fano entropy coder for image storage in synthetic dna,” in *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 1367–1371. 12, 19, 20
- [23] T. Ladune, P. Philippe, F. Henry, G. Clare, and T. Leguay, “Cool-chic: Coordinate-based low complexity hierarchical image codec,” 2023. 13
- [24] Y. Strumpler, J. Postels, R. Yang, L. V. Gool, and F. Tombari, “Implicit neural representations for image compression,” in *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVI*, 2022. 13
- [25] T. Ladune, P. Philippe, F. Henry, G. Clare, and T. Leguay, “Cool-chic: Coordinate-based low complexity hierarchical image codec,” *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10377686> 15

- [26] X. Pic, M. Dimopoulou, E. G. S. Antonio, and M. Antonini, “MQ-coder inspired arithmetic coder for synthetic dna data storage,” in *2023 IEEE International Conference on Image Processing (ICIP)*, 2023, pp. 3304–3308. 19
- [27] J. Jpeg, X. Pic, M. Dimopoulou, E. Gil, S. Antonio, and M. Antonini, “Coding of still pictures. technical description of the hidna submission to the jpeg dna cfp,” Tech. Rep., 2021. 19
- [28] H. Kim, M. Bauer, L. Theis, J. R. Schwarz, and E. Dupont, “C3: High-performance and low-complexity neural compression from a single image or video,” 2023. [Online]. Available: <https://arxiv.org/abs/2312.02753> 23, 24
- [29] Orange Open Source, “Cool-chic: Coordinate-based low complexity hierarchical image codec,” 2024, acessado em: 10/06/2025. [Online]. Available: <https://orange-opensource.github.io/Cool-Chic/> 23, 25
- [30] —, “Cool-chic documentation,” 2024, acessado em: 10/06/2025. [Online]. Available: <https://orange-opensource.github.io/Cool-Chic/> 24
- [31] —, “Decoding images with cool-chic,” 2024, acessado em: 10/06/2025. [Online]. Available: https://orange-opensource.github.io/Cool-Chic/image_compression/decoding_images.html 24
- [32] T. Leguay *et al.*, “Cool-Chic Video: A Low-Complexity Learned Video Codec,” <https://arxiv.org/abs/2402.03179>, 2024. 24
- [33] —, “Improved Encoding for Cool-Chic Video,” <https://arxiv.org/abs/2501.16976>, 2025. 24
- [34] B. Ladune *et al.*, “Low-Complexity Overfitted Neural Image Codec with Efficient Sub-Pixel Motion Compensation,” https://www.researchgate.net/publication/372583968_Low-complexity_Overfitted_Neural_Image_Codec, 2025. 24
- [35] M. Antonini and T. Ebrahimi, “Jpeg dna common test conditions version 2.0,” ISO/IEC JTC 1/SC 29/WG 1 (JPEG), Online, Standard WG1N100517, April 2023, documento apresentado na 99th JPEG Meeting, 24-28 April 2023. [Online]. Available: <https://jpeg.org/jpegdna/> 26, 30, 31
- [36] Orange OpenSource Team (Hyunjik Kim, Matthias Bauer, Lucas Theis, Jonathan Richard Schwarz, Emilien Dupont et al.), “Cool-chic: Low-complexity neural image and video codec,” Orange OpenSource. <https://orange-opensource.github.io/Cool-Chic/>, 2023, acessado em: 13 de Outubro de 2025. 27
- [37] T. Ladune, P. Philippe, G. Clare, F. Henry, and T. Leguay, “Cool-chic: Perceptually tuned low complexity overfitted image coder,” *arXiv preprint arXiv:2401.02156*, 2024. 27

- [38] T. Leguay, T. Ladune, P. Philippe, G. Clare, and F. Henry, “Low-complexity overfitted neural image codec,” *arXiv preprint arXiv:2307.12706*, 2023. 27
- [39] T. Blard, T. Ladune, P. Philippe, G. Clare, X. Jiang, and O. Déforges, “Overfitted image coding at reduced complexity,” *arXiv preprint arXiv:2403.11651*, 2024. 27
- [40] “Cool-chic: Presets for encoder and decoder configuration,” https://orange-opensource.github.io/Cool-Chic/image_compression/preset.html, Orange OpenSource, acessado em: 13 de Outubro de 2025. 28, 29
- [41] “Cool-chic: Decoding images guide,” https://orange-opensource.github.io/Cool-Chic/image_compression/decoding_images.html, Orange OpenSource, acessado em: 13 de Outubro de 2025. 28, 29
- [42] M. Antonini and T. Ebrahimi, “JPEG DNA Common Test Conditions, Version 2.0,” ISO/IEC JTC 1/SC 29/WG 1 (Joint Photographic Experts Group), 99th JPEG Meeting, Online, Standard Document WG1N100517, April 2023, available at the JPEG official document repository. 30
- [43] Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Multiscale structural similarity for image quality assessment,” in *Thirty-Seventh Asilomar Conference on Signals, Systems & Computers*, Pacific Grove, CA, USA, November 2003. 30
- [44] H. R. Sheikh and A. C. Bovik, “Image information and visual quality,” *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 430–444, Feb. 2006. 30
- [45] N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, J. Astola, and V. Lukin, “On between-coefficient contrast masking of dct basis functions,” in *Proceedings of the Third International Workshop on Video Processing and Quality Metrics*, vol. 4, 2007. 31
- [46] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, “Toward a practical perceptual video quality metric,” 2020, [Online]. Available: <https://github.com/Netflix/vmaf>. 31