Pontifícia Universidade Católica
do Rio de Janeiro

Diogo Munaro Vieira

Uma Nova Abordagem em Camadas para Representação de Dados Biológicos e suas Aplicações em Comparação de Sequências

Tese de Doutorado

Tese apresentada como requisito parcial para a obtenção do grau de Doutor pelo Programa de Pós–graduação em Informática, do Departamento de Informática da PUC-Rio.

Orientador: Prof. Sérgio Lifschitz

Rio de Janeiro
Setembro de 2023

Pontifícia Universidade Católica
do Rio de Janeiro

Diogo Munaro Vieira

Uma Nova Abordagem em Camadas para Representação de Dados Biológicos e suas Aplicações em Comparação de Sequências

Tese apresentada como requisito parcial para a obtenção do grau de Doutor pelo Programa de Pós–graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo:

Prof. Sérgio Lifschitz
Orientador
Departamento de Informática – PUC-Rio

Prof. Sérgio Colcher
Departamento de Informática – PUC-Rio

Prof. Edward Hermann Haeusler
Departamento de Informática – PUC-Rio

Prof. Rafael Dias Mesquita
UFRJ

Prof. João Carlos Setubal
USP

Rio de Janeiro, 22 de Setembro de 2023

Diogo Munaro Vieira

Graduado em Ciências Biológicas - Biofísica e mestre em Informática pela Universidade Federal do Rio de Janeiro.

Ficha Catalográfica

Munaro Vieira, Diogo

Uma Nova Abordagem em Camadas para Representação de Dados Biológicos e suas Aplicações em Comparação de Sequências / Diogo Munaro Vieira; orientador: Sérgio Lifschitz. – 2023.

110 f: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2023.

Inclui bibliografia

1. Informática – Teses. 2. Modelagem de Dados. 3. Biologia Molecular. 4. Proteínas Homólogas. 5. Representação de Características. 6. Visão Computacional. 7. Privacidade em Dados. 8. Aprendizado de Máquina. I. Lifschitz, Sérgio. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

A minha esposa, família, amigos e grupo de pesquisa
pelo apoio e encorajamento.

Agradecimentos

Ao meu orientador Professor Sérgio Lifschitz pelo estímulo e parceria para a realização deste trabalho. Esteve sempre presente, não me deixou desistir e com certeza atuou muito além do que era esperado como orientador.

Ao grupo de pesquisa BioBD, em especial aos Professores Antônio Miranda e Marcos Catanho, da Fiocruz, pela paciência e atenção, respondendo mensagens, mesmo nos fins de semana e ajudando com o contexto biológico.

Ao professor Edward Hermann por se juntar ao grupo de pesquisa, colaborando com formalismo matemático e discussões ricas sobre visão computacional.

Ao CNPq e à PUC-Rio, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

A minha orientadora de graduação Marília Guimarães da UFRJ, que me apresentou ao Professor Carlos Lucena e a pós-graduação da PUC-Rio.

Ao Professor Carlos Lucena e ao programa de pós-graduação, que me contemplaram com bolsa e possibilitaram meu aprendizado na PUC-Rio.

A minha esposa Vivi, por todos os fins de semana e noites me ajudando com a família e com a casa, se privando de muitas coisas para que eu pudesse desenvolver o trabalho.

Aos meus pais, pela educação, atenção e carinho de todas as horas. Também ao meu irmão Vinícius e meus cachorros que estavam sempre ao lado me confortando quando eu precisava.

A todos os profissionais da saúde que ajudaram na pandemia, em especial, minha psicóloga Veruska, que tem me ajudado a trilhar um novo caminho.

Aos meus colegas da PUC-Rio, Globo, OLX e PicPay que me apoiaram durante toda minha trajetória, em especial, meu gestor atual Raphael Dayan.

Aos professores que participaram da Comissão examinadora.

A todos os funcionários do Departamento pelos ensinamentos e ajuda.

A todos os amigos e familiares que, de uma forma ou de outra, me estimularam.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

Munaro Vieira, Diogo; Lifschitz, Sérgio. **Uma Nova Abordagem em Camadas para Representação de Dados Biológicos e suas Aplicações em Comparação de Sequências**. Rio de Janeiro, 2023. 110p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A identificação e categorização de proteínas homólogas são tarefas fundamentais no campo da biologia, que dependem de ferramentas que analisam sequências de nucleotídeos ou aminoácidos. No entanto, a detecção automatizada de padrões evolutivos, assim como outras características, usando métodos tradicionais, ainda apresenta desafios científicos. Neste estudo, propomos uma nova abordagem de representação de dados em camadas, que permite explorar padrões evolutivos e outras características de sequências na busca por similaridades, classificação e agrupamento. Utiliza-se um processo livre de alinhamento e são propostos novos algoritmos de similaridade que permitem aprimorar a eficácia dessa abordagem. Esses algoritmos utilizam técnicas inspiradas na percepção humana para capturar similaridades dentro das representações de moléculas biológicas. Avaliações experimentais demonstram bom desempenho e alta precisão em comparação com abordagens propostas anteriormente. Essa representação em camadas se mostra promissora na identificação de proteínas similares, principalmente com características de homólogas distantes. Além disso, sugere-se também o desenvolvimento de novos métodos e algoritmos de aprendizado de máquina em bioinformática que envolvam a privacidade e segurança de dados biológicos.

Palavras-chave

Modelagem de Dados; Biologia Molecular; Proteínas Homólogas; Representação de Características; Visão Computacional; Privacidade em Dados; Aprendizado de Máquina.

Abstract

Munaro Vieira, Diogo; Lifschitz, Sérgio (Advisor). **A New Layered Approach to Biological Data Representation and its Applications Comparing Sequences**. Rio de Janeiro, 2023. 110p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The identification and categorization of homologous proteins are fundamental tasks in the field of biology, relying on tools that analyze nucleotide or amino acid sequences. However, automated detection of evolutionary patterns and additional attributes using traditional methods still presents research challenges. In this study, we propose a novel layered data representation approach that allows us to explore evolutionary patterns and other sequence features in similarity searching, classification, and clustering. It employs an alignment-free process, and we introduce new similarity algorithms to enhance the effectiveness of this approach. These algorithms leverage techniques inspired by human perception to capture subtle similarities within biological molecules representations. Experimental evaluations demonstrate good performance and high accuracy compared to previously proposed approaches. This layered representation shows promise in identifying similar proteins, especially with distant homologs characteristics. Furthermore, it also suggests the development of new methods and machine learning (ML) algorithms in bioinformatics that address the privacy and security of biological data.

Keywords

Data Modeling; Molecular Biology; Homologous Protein; Feature Representation; Computer Vision; Data Privacy; Machine Learning.

Sumário

Lista de figuras

Lista de tabelas

Lista de algoritmos

Lista de Abreviaturas

ML – *Machine Learning*

PTN – Proteína

PDB – *Protein Data Bank*

LLM – *Large Language Model*

NLP – *Natural Language Processing*

PPI – *Protein Protein Interaction*

CGR – *Chaos Game Representation*

FGCR – *Frequency CGR*

SSIM – *Structural Similarity Index Measure*

XAI – *Explanaible Artificial Intelligence*

GDPR – *General Data Protection and Regulation*

LGPD – Lei Geral de Proteção de Dados

LCS – *Longest Common Subsequence*

pb – pares de base

aa – aminoácidos

MSA – *Multiple Sequence Alignment*

mtDNA – Genoma Mitocondrial

DL – *Deep Learning*

ANN – *Approximate Nearest Neighbors*

UQI – *Universal Quality Index*

MS-SSIM – *MultiScale SSIM*

MAP – *Mean Average Precision*

RF – Robinson-Foulds

BCM – *Branch Congurence Measure*

BIM – *Branch Incongurence Measure*

SW – Smith-Waterman

NW – Needleman-Wunsch

NJ – *Neighbor Joining*

EC2 – *Elastic Compute Cloud*

AWS – *Amazon Web Services*

G – Guanina, Glicina (contexto DNA ou PTN respectivamente)

C – Citosina, Cisteína (contexto DNA ou PTN respectivamente)

T – Timina, Treonina (contexto DNA ou PTN respectivamente)

A – Adenina, Alanina (contexto DNA ou PTN respectivamente)

R – Arginina

D – Aspartato

N – Asparagina

F – Fenilalanina

E – Glutamato

Q – Glutamina

H – Histidina

I – Isoleucina

L – Leucina

K – Lisina

M – Metionina

P – Prolina

S – Serina

Y – Tirosina

W – Triptofano

V – Valina

R-SSIM – *Resized SSIM*

RMS-SSIM – *Resized MS-SSIM*

WMS-SSIM – *Windowed MS-SSIM*

GS-SSIM – *Greedy Sliced SSIM*

US-SSIM – *Unrestricted Sliced SSIM*

What man sees depends both upon what he looks at and also upon what his previous visual-conception experience has taught him to see.

Thomas S. Kuhn, *The Structure of Scientific Revolutions.*

1
Introdução

Dados biológicos referem-se às representações de informações coletadas de diversas áreas da biologia, abrangendo moléculas biológicas, estruturas celulares, expressão gênica, características genéticas, fenótipos e interações moleculares (WOOLEY; LIN, 2005). Os dados são obtidos por meio de experimentos e técnicas específicas, como sequenciamento de DNA e RNA, microscopia e análises bioquímicas. Esses dados são essenciais para compreender os processos biológicos, identificar padrões e relacionamentos, além de impulsionar avanços na pesquisa científica e no desenvolvimento de tratamentos e terapias. Este trabalho possui maior foco em dados de sequências biológicas, mas pode ser extensível a outros dados de bioinformática.

A bioinformática é o campo da ciência que realiza pesquisas computacionais através de análise e processamento de grandes quantidades de dados biológicos, provenientes de diversas origens. Para cada origem específica, utilizamos formatos de arquivos previamente padronizados para interoperabilidade entre programas. Uma das representações de dados mais amplamente utilizada na bioinformática envolve arquivos FASTA (MILLS, 2014). Trata-se de um formato textual que fornece uma maneira simples e prática de armazenar informações de sequências e anotações de moléculas de DNA, RNA ou proteínas (PTN), permitindo visualização em qualquer aplicativo de edição de texto e facilitando o processamento por parte dos programas utilizados. Além disso, informações, como alinhamentos de sequências de nucleotídeos ou aminoácidos (ex.: ClustalW, MEGA, MSF) e dendrogramas filogenéticos (ex.: Nexus, Newick, Philip)(LEONARD; LITTLEJOHN; BAXEVANIS, 2006), podem ser armazenadas em formatos de texto semelhantes. Para dados de estrutura molecular 3D, o formato PDB (Protein Data Bank)(BERNSTEIN et al., 1977) é comumente utilizado, permitindo o armazenamento e compartilhamento de informações detalhadas sobre a estrutura tridimensional de PTNs.

A variedade de formatos de arquivo, em formato de texto, facilita a integração e análise de dados biológicos em diferentes contextos de pesquisa, contribuindo para avanços significativos em nossa compreensão dos processos biológicos. Ao mesmo tempo que essa variedade simplifica o uso dos dados para aplicações específicas, complica em uma visão geral e ecossistêmica de todo o processo biológico, ao passo que são formatos específicos somente pensados para uma determinada origem de dados. Já existem tentativas de padronização e modelagem de dados biológicos como: BioMart (KASPRZYK,

2011), Gene Ontology (ASHBURNER et al., 2000), BioModels (NOVÈRE et al., 2006), SEEK (WOLSTENCROFT et al., 2015) e recentemente o BioModelsML (TIWARI et al., 2023). Todos eles se preocupam com a padronização e disponibilidade dessa informação, que já é um problema muito relevante, mas sem se preocupar com a integração das informações para o uso na resolução de tarefas da bioinformática. Com isso, essas informações continuam sendo disponibilizadas, por exemplo, em arquivos FASTA ou PDB, que são as representações tradicionais. Nesse caso, quando é preciso utilizar informações de sequência de aminoácidos e estrutura 3D da molécula, é necessário lidar com arquivos FASTA e PDB separadamente.

As técnicas aplicadas em bioinformática, incluem também o uso de algoritmos de aprendizado de máquina. Existe um encapsulamento dos algoritmos de *machine learning* (ML) nos programas de bioinformática para cada fim, visto que há dificuldade do uso de todos os dados biológicos por conta da complexidade multifacetada dos processos biológicos (XU; JACKSON, 2019). Em biologia molecular, por exemplo, os programas de bioinformática costumam priorizar o uso de sequências das moléculas biológicas como entrada para tentar resolver vários dos problemas de bioinformática. O problema é que os dados de moléculas biológicas vão além de sequências, pois podem influenciar processos como transcrição, tradução, metabolismo celular, entre outros, que são aspectos importantes para a caracterização completa de uma molécula. Essa diversidade de aspectos torna a utilização dos dados biológicos uma tarefa complexa, exigindo o desenvolvimento de abordagens de ML cada vez mais adaptadas e sofisticadas.

Um exemplo da dificuldade associada à utilização dos dados biológicos foi o desenvolvimento do modelo AlphaFold (JUMPER et al., 2021; JONES; THORNTON, 2022) para predição da estrutura 3D de PTNs através de técnicas de ML. Os pesquisadores que desenvolveram o algoritmo precisaram utilizar como variáveis de entrada do modelo arquivos FASTA de PTNs homólogas a proteína para a qual queriam predizer a estrutura, em conjunto com um mapa par a par de coevolução de PTNs homólogas, além de outras variáveis dos aminoácidos da proteína-alvo. O processo de transformação dessas variáveis para algo que seja utilizável no modelo foi descrito de forma extensiva em seu material suplementar. Nele, os pesquisadores tentaram sumarizar os dados de diversas fontes para resolver especificamente o problema de predição estrutural de PTNs, mas precisaram criar diferentes tratamentos no algoritmo para isso, sendo complexo de reaproveitar em outros algoritmos diferentes no futuro.

Mesmo elevando o grau de sofisticação para a utilização de modelos de

ML recentes, como os *Large Language Models* (LLMs), o mesmo problema enfrentado persiste dado que é preciso preparar a informação biológica de forma que se adapte com a tecnologia. No caso específico de genômica, é possível citar o Nucleotide Transformer (DALLA-TORRE et al., 2023) e o DNAGPT (ZHANG et al., 2023) que são LLMs desenvolvidos especificamente para resolver tarefas da genômica, como predizer marcadores epigenéticos, ou promotores, ou recriar partes do genoma. Por serem modelos baseados em *Natural Language Processing* (NLP), eles necessitam de palavras como entrada e o genoma é estruturado como uma única palavra gigante com vários nucleotídeos como letras. Para lidar com isso, quebram o genoma em sequências menores, "inventando" palavras, sendo o Nucleotide Transformer de seis letras (nucleotídeos), conhecido na literatura como abordagem em *k-mers*, onde $k = 6$. O número seis foi escolhido, por ser o que deu melhores resultados, mas o aspecto biológico se perde nessa afirmação por estarmos agrupando os nucleotídeos sem entender a semântica de ter aqueles nucleotídeos agrupados. Essa abordagem acaba por forçar ao algoritmo que entenda o genoma a cada seis palavras, podendo quebrar partes importantes de um genoma, perdendo parte do contexto de outros nucleotídeos em volta dele. No DNAGPT os autores comentam sobre esse mesmo problema do Nucleotide Transformer e usam, além do *k-mer* de palavras, outras características para descreverem mais informações sobre a sequência e tentar amenizar o problema identificando regiões importantes das sequências. Assim como no AlphaFold, lidam com cada uma dessas características de formas diferentes no modelo, tornando complexa a adição de novas informações e comparações com outros modelos.

Essa abordagem baseada em *k-mers* é só um dos exemplos em que tentamos forçar o encaixe de moléculas biológicas como DNA, RNA e PTNs, que são de cunho molecular, em uma representação adaptada para textos, a fim de facilitar seu manuseio e comparação. Outro exemplo mais comum de simplificação dessas moléculas para a comparação de sequências é feito através de *multiple sequence alignment* (MSA) (RANWEZ; CHANTRET, 2020). O processo de alinhamento de sequências busca encontrar similaridade entre as sequências otimizando para que o máximo de caracteres estejam alinhados entre as sequências como mostra a Fig 1.1. O número de caracteres alinhados que são iguais equivale à identidade do alinhamento mas, no caso da Seq1 com a Seq3 nesse exemplo, os nucleotídeos A e T não vão contar para a identidade. Essa forma de comparação pode gerar *gaps* onde o algoritmo prefere pular o caractere para alinhar melhor com os outros como acontece na Seq2. Estes métodos de comparação de sequências, em si, não são um problema mas, novamente, eles só consideram a abstração sequencial da molécula, quando, na

verdade, ela é uma entidade que interage com outras moléculas, desempenha funções no sistema que se encontra e sofre pressões evolutivas, dependendo de suas características e do ambiente que está inserida.

```
Seq1  A  T  A  G  C
      |  |     |  |
Seq2  A  T  -  G  C
         |     |  |
Seq3  -  T  T  G  C
```

Figura 1.1: MSA entre três sequências (Seq1, Seq2 e Seq3) com *gaps*.

Também na Fig 1.1, aparece um *gap* à esquerda na Seq3. É normal de se encontrar muitos *gaps* nas extremidades de resultados de métodos de MSA porque podem ter sequências muito diferentes dentro de um mesmo alinhamento. Essas sequências diferentes vão colocar inúmeros *gaps* que serão desconsiderados no alinhamento final, porque são tratados como regiões pouco conservadas ou de baixa importância para o estudo por não se relacionar com mais nada. É normal que algoritmos populares como o Gblocks (CASTRESANA, 2000) façam um filtro no MSA, removendo essas regiões das extremidades que não foram contempladas no alinhamento. As regiões pouco conservadas das sequências explicitam as maiores diferenças de uma sequência para outras que estão sendo comparadas, sendo mais eficazes em diferenciar as sequências, e mostrando padrões que não necessariamente são vistos só nas similaridades. Mesmo com vários estudos entendendo a importância de regiões pouco conservadas das sequências em caráter evolutivo (NAG et al., 2006; LI et al., 2017; KLUSKA et al., 2022), um dos métodos mais utilizados até hoje para análise de sequências em biologia molecular é através dos métodos de MSA, com filtros como o Gblocks para comparar sequências. Por outro lado, existem também os métodos *alignment-free* (REN et al., 2018; ZIELEZINSKI et al., 2017) que tentam sanar problemas dos métodos de alinhamento para comparar moléculas biológicas, principalmente relativos a eficiência, mas que também acabam por utilizar essencialmente aspectos sequenciais das moléculas. Na família de métodos *alignment-free*, os algoritmos mais comuns basicamente utilizam frequência de repetição de caracteres da sequência (*k-mers*), ou estatísticas em cima dos caracteres, para inferir similaridades através de algoritmos de distância (VINGA; ALMEIDA, 2003). Um exemplo é utilizar a distância euclideana para comparar o percentual de frequência de nucleotídeos de várias sequências.

Abordagens baseadas em métodos *alignment-free*, como o *k-mers*, melhoram a eficiência na resolução de problemas da bioinformática através da

discretização de informação das sequências. Elas são uma excelente alternativa para resolver problemas que demoram muito tempo através de métodos de alinhamento, apesar de todos os aspectos supracitados.

Com isso, além da dificuldade do uso de ML em biologia molecular, nota-se que dificuldades semalhantes também são encontradas em metodologias mais tradicionais da bioinformática. O principal problema e motivador do trabalho resume-se como a dificuldade de ter uma visão holística das informações de moléculas biológicas pelos seguintes motivos:

- por conta da dificuldade de agregação de todas as informações de moléculas biológicas para obter uma visão holística, existe uma especialização nas abstrações sequenciais delas, fazendo com que sejam esquecidos outros aspectos evolutivos e interacionais dessas moléculas;
- dentro das comparações sequenciais, as sequências são cortadas em partes pouco conservadas, ou em *k-mers* fixos, que podem ser relevantes de serem comparadas juntas ou com outros agrupamentos dependendo de cada molécula e não necessariamente usando os valores fixos;
- as características das sequências, interação das moléculas, comportamento no ambiente podem ser comparados separadamente para uma molécula biológica, mas não há técnicas comparativas para a molécula biológica em sua completude.

Todos esses aspectos contribuem para um uso parcial do conhecimento existente sobre as moléculas biológicas, aumentando a chance de vieses e tornando mais difícil a comparação entre diferentes pesquisas em biologia molecular.

Dada a necessidade de avaliação do conhecimento das moléculas biológicas em diversos aspectos, além de somente um conjunto de letras em uma sequência, é proposta nesta tese uma nova forma de representação extensível, reutilizável e comparável de moléculas biológicas. Além disso, faz-se necessário entender a melhor forma de comparar essa nova representação levando em consideração todos os aspectos das moléculas. Este trabalho propõe um método livre de alinhamento (*alignment-free*) para identificação de sequências similares, simulando a realização das tarefas de classificação, agrupamento e busca de homólogos. Além dessas tarefas, nosso método se mostra promissor para outras aplicações de bioinformática que possam ser realizadas através de similaridade de sequências.

Este manuscrito está organizado inicialmente para explicar o que tem sido utilizado em biologia molecular para realizar a busca e o agrupamento de homólogos, olhando desde métodos mais tradicionais até o uso de ML e

seus desafios na aplicação em biologia no Capítulo 2. Também nesse capítulo, é feita uma revisão do que existe na literatura, visando a modelagem de dados de componentes biológicos. Em seguida serão apresentados, no Capítulo 3, os dados utilizados nos experimentos, uma modelagem física de dados de moléculas biológicas, que permite a criação de uma visão holística, e uma nova metodologia para comparação das representações de dados em casos de uso da bioinformática. No Capítulo 4, são sugeridos: uma representação de dados de sequências biológicas, algoritmos de similaridade propostos para a nova representação de dados e uma nova metodologia *alignment-free*. Também são apresentados alguns resultados obtidos com dados experimentais e bases curadas. Por último, no Capítulo 5, esta tese é concluída com um resumo das contribuições, trabalhos futuros e possíveis linhas de pesquisa que podem aproveitar da abordagem proposta.

2
Trabalhos relacionados

Neste capítulo encontram-se a explicação do que são sequências homólogas e os métodos mais clássicos da bioinformática aplicados a elas. Depois disso, ele começa a explorar técnicas de MSA, que é um método bastante usado como base para agrupamento de homólogos. Em seguida, será descrito como são feitas as buscas por sequências homólogas, junto com alguns métodos *alignment-free* com uso de ML, enaltecendo alguns pontos em aberto na literatura. Conclui-se esse capítulo com algumas formas de modelar os dados biológicos atualmente descritos na literatura.

2.1
Abordagens Tradicionais

Sequências homólogas podem ser tanto sequências de DNA, RNA ou de aminoácidos referentes a uma PTN que tenha ancestral comum de outra sequência de PTN com função similar (FITCH, 1970). Essas sequências são importantes de serem estudadas porque, a partir de um padrão encontrado, pode-se inferir a existência de proteínas similares ou com a mesma função em outras sequências desconhecidas (SÖDING, 2005). Não só através das sequências de letras é possível encontrar sequências homólogas mas, também, através da sua estrutura secundária (GINALSKI et al., 2003) ou terciária (LANGMEAD; DONALD, 2004). O alinhamento par a par e métodos de MSA são as formas mais eficazes de se buscar (LADUNGA, 2017) e agrupar homólogos (ALI et al., 2019), respectivamente, mesmo com aspectos estruturais já sendo bem descritos como importantes nesses casos de uso. Ultimamente, tem se investido em alinhamentos estruturais mais rápidos, como no caso do FastFoldSeek (KEMPEN et al., 2023), que podem ser usados também para detecção de homólogos.

Os métodos de MSA são um problema MaxSNP-difícil que tenta otimizar uma função multiobjetivo dado que o seu resultado ótimo depende do alinhamento correto entre várias sequências (WANG; JIANG, 1994). Esse tratamento do MSA como função multiobjetivo tem como objetivo evitar problemas anteriores de métodos de MSA que buscavam otimização única e acabavam por olhar somente sequências muito conservadas em todo o alinhamento. Com esse novo foco multiobjetivo, cada algoritmo de MSA começou a se tornar um encapsulador de metodologias para alinhar sequências. Um exemplo é o Clustal Omega (SIEVERS et al., 2011; SIEVERS; HIGGINS, 2021) que

usa o mbed (BLACKSHIELDS et al., 2010) para comparar a distância entre sequências através de *k-mers* e, em seguida, agrupa essas sequências com k-means ou UPGMA, gerando uma *guide tree* que vai ser usada pelo pacote HHalign (SÖDING, 2005) para alinhar par a par as sequências mais próximas no agrupamento através de *hidden markov models*, validando a probabilidade de uma sequência se tornar a outra através de inserções, deleções e mutações aleatórias. Esse alinhamento final par a par incremental é chamado de "alinhamento progressivo" e é usado para tentar evitar erros na inserção de *gaps*. No alinhamento progressivo, primeiro, é feito um dendrograma de base (*guide tree*) e, depois, vão sendo agregadas as sequências no alinhamento das folhas até a raiz, como mostrado na Fig 2.1.

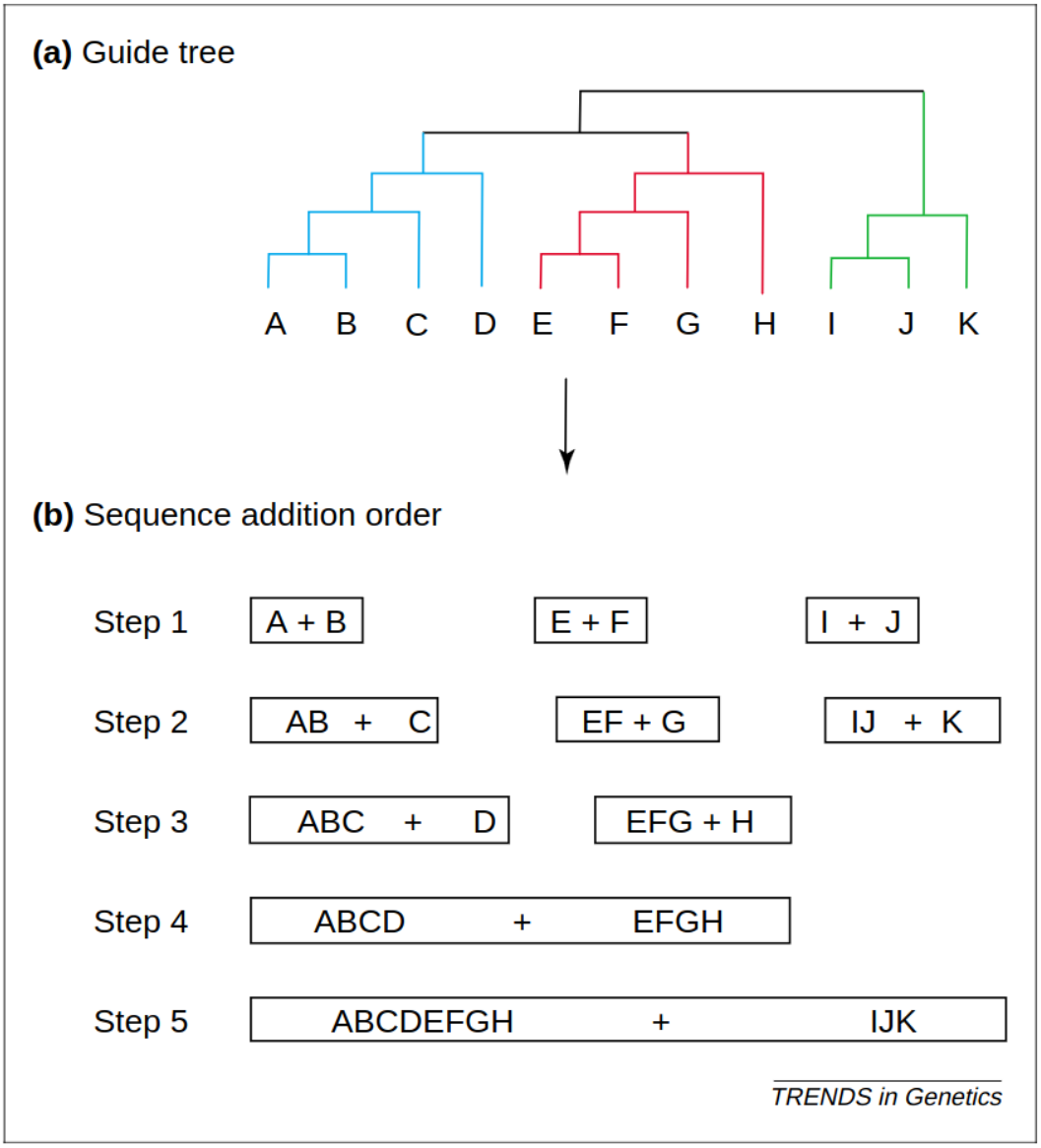


Figura 2.1: Alinhamento progressivo extraído de (BALDAUF, 2003).

Com essas novas técnicas, os algoritmos baseados em MSA passam a ser

bem mais aderentes ao contexto evolutivo da molécula biológica. Ainda assim, existe a necessidade de filtros similares ao Gblocks (CAPELLA-GUTIÉRREZ; SILLA-MARTÍNEZ; GABALDÓN, 2009; BOROWIEC, 2019; STEENWYK et al., 2020) para remoção de áreas de baixa complexidade ou partes de alinhamentos entre sequências muitos distintas. Esses filtros normalmente são chamados de *trimming* (aparadores) e ainda são bastante necessários, pelo menos em filogenia (PORTIK; WIENS, 2021). Além disso, existem também abordagens de MSA, que utilizam outras informações de aminoácidos (DAUGELAITE; DRISCOLL; SLEATOR, 2013), como o MAFFT (KATOH et al., 2002; KATOH; ROZEWICKI; YAMADA, 2019), que converte os aminoácidos em uma combinação de volume e polaridade para identificar similaridade entre as sequências; ou que fazem alinhamentos de estrutura 3D de sequências, como o 3D-COFFEE (O'SULLIVAN et al., 2004) ou o MAFFT-DASH (ROZEWICKI et al., 2019), que utilizam bases de dados com sequência e estrutura de proteínas. Os alinhamentos estruturais e das sequências de caracteres nesses casos são feitos separadamente e depois são agregados para compor o resultado final. Como o alinhamento estrutural, apesar dos recentes esforços, ainda ter que escolher entre ser computacionalmente custoso ou acurado (HOLM, 2022; KEMPEN et al., 2023), o MAFFT-DASH já disponibiliza alguns alinhamentos estruturais par a par de proteínas pré-calculados para acelerar as comparações. Essa alternativa de método de MSA é a que possui maior quantidade de informação agregada, mas ainda permanece com a necessidade de aparadores de MSA, é utilizado só para PTN e precisa processar os dados de diferentes formas para conseguir gerar um resultado final.

A forma de comparar os dados através dos métodos de MSA também pode ser utilizada para a busca de homólogos, ou a busca de qualquer sequência similar, no caso do BLAST (MCGINNIS; MADDEN, 2004; ALTSCHUL, 2014), que é o algoritmo mais famoso para a busca de sequências similares de PTN ou DNA. O BLAST divide as sequências em pequenas "palavras", aplicando a mesma técnica de *k-mers*, e faz buscas locais (chamadas de alinhamentos) entre as sequências. Para as sequências que possuem correspondências, o método filtra as mais parecidas de acordo com um limite e retorna somente as mais similares. Além dos métodos de MSA, para agrupamento, e o BLAST, para a busca de homólogos, existe uma variedade de métodos para a realização de vários casos de uso na bioinformática, incluindo métodos *alignment-free* (REN et al., 2018; DELIBAS, 2022) que não usam alinhamentos para solucionar os problemas da bioinformática.

2.2 Aprendizado de Máquina

Buscando, na literatura por alternativas a abordagens de *k-mers*, comumente utilizadas na bioinformática e, coincidentemente, também em pré-processamento de dados para NLP em geral, foi observado que existem diversas implementações de representações de dados biológicos, normalmente limitados ao uso de uma técnica específica. Com isso, o dado é modelado para funcionar com aquela técnica sem considerar uma consolidação do conhecimento adquirido da entidade biológica. Nessa revisão bibliográfica (YANG et al., 2020) são descritas diversas abordagens do uso de DNA em técnicas de ML. Junto com cada uma das técnicas são descritas formas de representar o dado biológico sempre com a mesma modelagem de *k-mers*, armazenados como: vetores de características, tuplas, grafos direcionados, tabelas *hash*, diagramas de frequência de nucleotídeos, etc. No mesmo trabalho são descritas algumas oportunidades em aberto, dentre elas três são abordadas nesta tese:

1. Como extrair conhecimento e armazenar informações do DNA para processamento com ferramentas de mineração de dados;

2. Desenhar algoritmos de distância entre moléculas biológicas para atuar efetivamente nos dados biológicos e no conhecimento gerado utilizando eles; e

3. Como conseguir a explicabilidade dos resultados para um dado algoritmo aplicado, entendendo quais informações foram mais relevantes para cada decisão feita durante a predição.

O interessante é que dois anos depois, outra revisão (AO et al., 2022) reuniu diversas formas de utilizar classificação em dados biológicos, tanto para DNA, como RNA e PTNs. Os autores comentaram sobre os mesmos pontos a cima como oportunidades, mostrando pouco avanço científico nessa questão. Além dos três pontos já enumerados aqui, apareceram oportunidades nas áreas de padronização de dados e dificuldade de colaboração na área científica, devido à alta complexidade da obtenção de conjuntos de dados através de várias fontes, assim como o processamento dos dados para geração de *features* para modelos de ML.

Mais uma revisão da literatura (GREENER et al., 2022) sobre ML para biólogos trouxe as mesmas preocupações e ainda adicionou algumas outras como: disponibilidade dos dados com uma variedade enorme de duplicidades; desconhecimento de um conjunto completo dos dados, podendo aparecer a qualquer momento algo que nunca tinha sido visto; e outros três que também são abordados nesta tese:

4. Privacidade no uso de dados biológicos, genéticos e biomédicos por serem dados altamente sensíveis de pacientes;

5. Dificuldade em encontrar algo reutilizável e de fácil utilização para outros modelos de ML; e

6. Colaboração e agregação de dados de diferentes pesquisas, assim como foi pincelado na última revisão.

Para resolver a primeira questão sobre extração e armazenamento de dados, já foram feitas iniciativas de modelagem de dados biológicos em *Data Warehouses* (DWs) (SCHÖNBACH; KOWALSKI-SAUNDERS; BRUSIC, 2000; BALLARD et al., 2002; HUANG et al., 2003; KARP; LEE; WAGNER, 2008), mas novamente as alternativas funcionam para a integração de dados obtidos a partir de diversos programas e não na extração de conhecimento a partir desses dados biológicos. Com a quantidade de informação e dados biológicos existentes, o maior desafio é não só de recuperação de informação (*information retrieval*), mas sim, de gerar conhecimento através dessa informação e conseguir medir a similaridade dos dados no contexto biológico.

Existem vários dados e contextos para se pensar em algoritmos de similaridade ou distância (inverso da similaridade) para dados biológicos, indo desde à distância entre estruturas de PTNs (JESCHKE, 2012) e medida de distância entre interação de proteínas PPIs (*protein protein interaction*) (IBRAGIMOV et al., 2013), até as distâncias entre os dados sequenciais de PTNs e DNA. Considerando dados sequenciais, existem diversas abordagens de distância envolvendo *k-mers* (RÖHLING et al., 2020) e outras opções envolvendo a distância entre DNAs com adaptações de algoritmos de *ranking* (IONESCU, 2013), metrificando a diferença das sequências através da ordenação das letras, ou ainda a distância através de cadeias de Markov (JUNYAN; CHENHUI, 2015) conforme explicado na Seção 2.1 no alinhamento progressivo.

Uma abordagem bastante interessante de representação de dados tem considerado as sequências como imagens através de *Chaos Game Representation* (CGR) (JEFFREY, 1990), onde a sequência de DNA é traduzida para uma imagem através de uma função matemática, que coloca cada nucleotídeo em um lugar da imagem, a partir da sua posição no DNA. Existem várias aplicações de CGR e da sua versão compacta baseada em frequências de letras (FCGR), não só para DNA, mas recentemente também para PTNs (DICK; GREEN, 2020), podendo ser utilizado em várias aplicações biológicas (LÖCHEL; HEIDER, 2021).

Para a forma de representação de CGR, foram testadas algumas medidas de distância e a que obteve melhores resultados para tarefa de agrupamento de

sequências, de acordo com sua classificação biológica, foi a baseada em *Structural Similarity Index Measure* (SSIM) (KARAMICHALIS et al., 2015). O SSIM utiliza análise estrutural das imagens para encontrar similaridade entre elas e, com isso, consegue inferir a distância através do inverso da similaridade DSSIM ($1-SSIM$), obtendo valores entre 0 e 2 de distância. Como os algoritmos baseados em SSIM não obedecem à desigualdade triangular (BRUNET; VRSCAY; WANG, 2012; NILSSON; NVIDIA, 2020), não podem ser chamados de distância. Por isso, será usado nesta tese o termo dissimilaridade ao invés de distância.

A CGR é uma abordagem que se conecta muito bem com o SSIM porque explicita uma assinatura genômica (DESCHAVANNE et al., 1999) com um formato geométrico na imagem, facilitando sua detecção por algoritmos que avaliam a estrutura e a topologia da imagem. Ainda que a CGR possa ser usado para a realização de várias tarefas como predição de homólogos (BURMA et al., 1992) ou predição de estrutura tridimensional de PTNs (SUN et al., 2020), não existe uma forma de agregar uma diversidade de informações na mesma estrutura, dado que ela sempre é adaptada em tamanho e formato para cada problema. Com isso, a CGR, assim como as outras formas de representação em *k-mers*, texto, grafos, etc., dificultam a agregação de informações e identificação de qual é a informação mais importante para a realização de uma tarefa através de ferramentas de explicabilidade.

2.3 Explicabilidade e Privacidade de Dados

A explicabilidade de algoritmos de ML é uma linha de pesquisa bastante recente, mais conhecida como *Explainable artificial intelligence* (XAI), em que a motivação principal é conseguir explicar o motivo dos algoritmos tomarem determinadas decisões e interpretar aquelas decisões tomadas por conta dos dados de entrada no algoritmo (MARCINKEVIČS; VOGT, 2020). Na bioinformática, a explicabilidade é uma área mais recente ainda e possui diversos desafios (HAN; LIU, 2022). O maior dos problemas é que os métodos que existem hoje não conseguem passar a transparência sobre o motivo dos resultados terem acontecido por conta da complexidade inerente aos vários formatos de dados biológicos que os modelos precisam lidar e transformar antes de serem realmente utilizados. Nessa revisão (KARIM et al., 2022) sobre como aplicar métodos de XAI em aplicações de bioinformática, os autores precisam lidar com cada dado específico e utilizar um determinado método de explicabilidade para cada problema encontrado, terminando por deixar claro essa dificuldade na conclusão.

A explicabilidade na área da saúde, como um todo, é bastante crítica porque a transparência e confiança nos resultados pode determinar a vida ou morte de um paciente (HULSEN, 2023). Ao mesmo tempo que a explicabilidade é de grande importância, ela também abre "portas" para desafios em privacidade (BOZORGPANAH; TORRA; ALIAHMADIPOUR, 2022; SAIFULLAH et al., 2022), dado que se é possível explicar exatamente o que aconteceu através dos dados de um paciente, precisa-se tomar cuidado para a transparência não acabar revelando dados sensíveis.

O tema de privacidade dos dados ganhou mais força recentemente com as regulamentações no uso de dados em 2018, sendo as principais a *General Data Protection and Regulation* (GDPR), na Europa, e a Lei Geral de Proteção de Dados (LGPD), no Brasil (LORENZON, 2021). Ambas as regulamentações atribuem dados genéticos e biomédicos como sensíveis e isso abre uma gama de implicações (ABOUELMEHDI; BENI-HESSANE; KHALOUFI, 2018; ARSHAD et al., 2021). Dentro dessas implicações, está a exposição dos dados genéticos que atualmente são salvos em bancos de dados encriptados ou mascarados. A técnica de *data masking* é diferente de uma encriptação porque o objetivo é impedir que alguém não autorizado entenda a informação que está escondida ali e não requer uma chave criptográfica para retorno do dado. As técnicas que existem hoje para ofuscar os dados acabam sendo feitas para impedir que eles sejam utilizados, a menos que se consiga remover a ofuscação (SIDDARTHA; RAVIKUMAR, 2019; SIDDARTHA; RAVIKUMAR, 2020; SATRA et al., 2023). Uma abordagem interessante seria remover a sensibilidade do dado, mantendo sua característica e semântica em relação aos outros para que permaneça utilizável sem o conteúdo sensível disponível.

2.4 Modelagem de Dados

A utilização dos dados biológicos é bastante complexa por conta desses vários tópicos que já foram citados, mas especialmente porque hoje não existe um padrão reutilizável de trabalhar com esses dados. Com isso, a escolha e aprimoramento de novas *features* tornam-se complexas. Muitas vezes, o que acontece com dados de moléculas biológicas é que são aplicadas técnicas de processamento de texto, mas é esquecido que ali não existem palavras, mas sim, um conjunto de caracteres com semântica apropriada. Quando se busca, na literatura por uma modelagem em dados biológicos, encontra-se muitas modelagens de redes e processos biológicos (SOMEKH; CHODER; DORI, 2012; NALDI et al., 2015). Esse tipo de modelagem costuma ser complexa por envolver processos dinâmicos (ALCALÁ-CORONA et al., 2021)

e pela preocupação em representar as interações do sistema como um todo. Existem modelagens que levam em consideração os aspectos moleculares dos compostos mais básicos como aminoácidos e nucleotídeos, mas muito focados em uma abordagem *top-down*, colocando as características sempre elencadas à macromolécula da que eles fazem parte (MACEDO et al., 2007; BORNBERG-BAUER; PATON, 2002; IDREES; KHAN, 2015; LIFSCHITZ et al., 2022).

Trabalhos recentes têm observado cada vez mais a necessidade de uma modelagem com abordagem *bottom-up* dos dados que considere diversas fontes para caracterizar parte daquele genoma. Depois dessa caracterização, é criado um modelo conceitual que abranja esses dados (CERI et al., 2018; BERNASCONI et al., 2023). O problema dessas abordagens *bottom-up* é que focam em conseguir organizar os dados obtidos ainda em uma estrutura *top-down*, indo das macromoléculas para as moléculas mais simples. A proposta desta tese pretende tangenciar cada um desses pontos, trazendo uma alternativa *bottom-up* de modelagem e representação de informação de moléculas biológicas, que preserve a semântica delas. A modelagem terá maior foco em sequências de nucleotídeos e aminoácidos, visando a simplificação da utilização completa do conhecimento das moléculas e facilitando a colaboração entre diferentes pesquisas.

3
Materiais e Métodos

A metodologia desenvolvida se baseia na informação presente nas moléculas biológicas, com maior foco nas sequências, para criar novas representações de dados *bottom-up*, extensíveis e reutilizáveis em diversos algoritmos. A nova representação leva em consideração e colabora com aspectos relevantes de privacidade e explicabilidade dos dados, agregando esses aspectos em algoritmos que a utilizem. Além da nova representação, fez-se necessário criar algoritmos de similaridade focados na nessa forma de representar os dados. As representações foram testadas com conjuntos de dados de sequências homólogas dentro de casos uso da bioinformática de busca, similaridade e agrupamento de homólogos. Depois de testadas, as representações e os algoritmos de similaridade foram validados com dados públicos curados de nucleotídeos e aminoácidos.

Os conjuntos de dados usados durante o trabalho e uma análise em cima deles são descritos na Seção 3.1, seguido por uma explicação fim a fim do novo método para representação e comparação de sequências biológicas, incluindo métricas de avaliação na Seção 3.2. Depois disso, é explicado como o novo método foi aplicado para casos de uso da bioinformática, assim como os recursos computacionais utilizados para sua aplicação e otimização na Seção 3.3.

3.1
Conjuntos de Dados

Durante o desenvolvimento desta pesquisa, foram considerados dados sintéticos e reais para criar as representações de dados, a fim de ter um grupo sintético mais controlado e outro grupo dos reais bem descritos e conhecidos.

3.1.1
Dados Experimentais

Para desenvolvimento dos experimentos, foram utilizadas as informações de DNA e de PTN das sequências. Inicialmente, foram obtidos dados através da ferramenta *INDELible* (FLETCHER; YANG, 2009), para simular homólogos distantes, e dados de sequências codificantes de homólogos da família das Globinas. As sequências utilizadas nos experimentos com DNA foram as mesmas coletadas nas fontes de informação, enquanto as de PTN foram obtidas através da tradução das sequências de DNA três a três, formando sequências de aminoácidos três vezes menores do que as de nucleotídeos. Esse agrupamento

três a três dos nucleotídeos para confeccionar aminoácidos é chamado de "códon".

3.1.1.1 Sequências Sintéticas

Para utilizar um conjunto de dados totalmente controlado, optou-se por obter dados bem simples de evolução do software *INDELible*. Para isso, utilizou-se como base o exemplo mais simples de nucleotídeos do próprio site [1], a fim de gerar as sequências. No exemplo, são utilizadas somente 2 populações A e B, com 100 variações cada uma e com 1000 nucleotídeos em cada indivíduo gerado. Esse exemplo foi trocado para 4 populações A, B, C e D, com 10 variações cada e com 3000 nucleotídeos em cada indivíduo. Com isso, uma maior variedade de indivíduos, com um número de nucleotídeos maior e pouca similaridade entre as sequências estão simulando populações de homólogos muito distantes.

A similaridade dessas sequências se comportam como se fossem 10 conjuntos de dados distintos e sem vínculo hereditário, cada um com 4 populações A, B, C e D. Com isso, durante este trabalho as populações do conjunto 0 seriam representadas por $A0$, $B0$, $C0$, $D0$; as do conjunto 1 por $A1$, $B1$, $C1$, e $D1$ e assim sucessivamente. Os conjuntos de dados possuem mais semelhança interna do que com os outros 9 conjuntos como mostra a Tabela 3.1. Nela estão os resultados de um alinhamento par a par das sequências através do algoritmo global Needleman-Wunsch (NW) (NEEDLEMAN; WUNSCH, 1970) com a média calculada para a as sequências dentro e fora do próprio conjunto de dados. Assim, consegue-se observar na tabela que $A0$, $B0$, $C0$ e $D0$ são mais parecidos entre si do que com $A2$, $B2$, $C2$ e $D2$, por exemplo.

Para geração das sequências foi considerado um modelo evolutivo bastante simples e estável denominado como JC69 (JUKES; CANTOR, 1969), onde as substituições entre nucleotídeos possuem probabilidade igual de acontecer entre todos eles. Além disso, a árvore a ser formada pelo modelo foi desenvolvida agrupando ramos A e B como mais próximos de um lado, assim como C e D do outro. Isso foi ilustrado com os conjuntos de dados 3 e 4 na Tabela 3.2, em que foi utilizado o algoritmo de NW par a par para encontrar as sequências mais similares. Os números em vermelho indicam maior similaridade das sequências e é possível observar com um azul mais claro, que dentro de um mesmo conjunto, 3 ou 4, os elementos possuem mais similaridade do que com os outros.

[1] <http://abacus.gene.ucl.ac.uk/software/indelible/tutorial/>

Dados	Média Dentro	Média Fora
0	0.544	0.502
1	0.547	0.503
2	0.542	0.502
3	0.542	0.500
4	0.545	0.500
5	0.548	0.502
6	0.544	0.503
7	0.541	0.501
8	0.548	0.502
9	0.544	0.502

Tabela 3.1: Média da identidade das sequências com outras dentro do próprio conjunto de dados e com outras sequências fora do conjunto. Maiores valores indicam maior similaridade com sequências de dentro do conjunto.

	A3	B3	C3	D3	A4	B4	C4	D4
A3	1.000	0.617	0.509	0.507	0.494	0.498	0.499	0.489
B3	0.617	1.000	0.520	0.506	0.494	0.499	0.493	0.493
C3	0.509	0.520	1.000	0.596	0.494	0.503	0.509	0.502
D3	0.507	0.506	0.596	1.000	0.488	0.503	0.499	0.494
A4	0.494	0.494	0.494	0.488	1.000	0.598	0.513	0.512
B4	0.498	0.499	0.503	0.503	0.598	1.000	0.514	0.517
C4	0.499	0.493	0.509	0.499	0.513	0.514	1.000	0.615
D4	0.489	0.493	0.502	0.494	0.512	0.517	0.615	1.000

Tabela 3.2: Identidade de alinhamento par a par feito para cada sequência de DNA gerada pelo *INDELible*. Maiores resultados indicam maior similaridade de sequências.

3.1.1.2 Globinas

O conjunto de dados reais de DNA com a região codificante de PTN da família das Globinas foi obtido no Ensembl [2], que contém tanto uma diversidade de espécies quanto de tipos de PTNs. Ao todo são 5 PTNs homólogas diferentes, com 15 espécies para cada uma dessas PTNs, totalizando 75 sequências de DNA. As espécies utilizadas para montar esse conjunto de dados foram de diferentes primatas: *Rhinopithecus Roxellana*, *Rhinopitecus Bieti*, *Mandrillus Leucophaeus*, *Chlorocebus Sabaeus*, *Pongo Abelii*, *Nomascus Leucogenys*, *Pan Troglodytes*, *Pan Paniscus*, *Homo Sapiens*, *Gorila Gorila*, *Carlito Syrichta*, *Cebus Capucinus*, *Aotus Nancymaae*, *Prolemus Simus* e *Otolemur Garnetti*. As sequências de Globinas de primatas foram escolhidas por terem uma filogenia bem conhecida e um comportamento evolutivo bem descrito. Além disso, diferentes globinas possuem uma variedade de características nas sequências que

[2] <https://www.ensembl.org/index.html>

simplifica a validação de diferentes aspectos no método proposto nesta tese, conforme será mostrado na próxima seção de análise de dados.

3.1.1.3 Análise dos Dados

Visando descrever bem os dados experimentais, foram feitas avaliações estatísticas sobre cada um dos conjuntos de dados. O tamanho das sequências de DNA estão descritas na Tabela 3.3a e de PTN, na Tabela 3.3b, ambas utilizando a mediana, média, desvio padrão, mínimo e máximo. Outra informação extraída é a análise par a par do *Longest Common Subsequence* (LCS) que traz o dado do máximo de caracteres sequenciais que as sequências no conjunto de dados têm em comum, indicando subsequencias similares. Depois de calcular todos os LCS par a par, foi feita a mediana por conjunto de dado. Serão referidos aos tamanhos para sequências de DNA em pares de base (pb) e de PTN em aminoácidos (aa).

Dados	**# Seqs**	**Mediana**	**Média**	**Desv**	**Mín**	**Máx**	**LCS**
Hemoglobin$_\beta$	15	441	441	0	441	441	88
Myoglobin	15	465	465	0	465	465	65
Neuroglobin	15	456	451,6	11,43	417	456	77
Cytoglobin	15	618	596,2	66,25	378	678	113
Androglobin	15	4929	4726,4	694,56	2148	5004	130
INDELible	40	3000	3000	0	3000	3000	11

(a) Estatísticas de DNA em pb.

Dados	**# Seqs**	**Mediana**	**Média**	**Desv**	**Mín**	**Máx**	**LCS**
Hemoglobin$_\beta$	15	147	147	0	147	147	40
Myoglobin	15	154	154	0	154	154	33
Neuroglobin	15	151	149,6	4,61	139	151	113
Cytoglobin	15	205	197,9	22,1	125	225	67
Androglobin	15	1642	1574,9	231,31	716	1667	100
INDELible	40	1000	1000	0	1000	1000	5

(b) Estatísticas traduzidas para PTNs em aa.

Tabela 3.3: Estatística dos conjuntos de dados para os nucleotídeos e aminoácidos traduzidos das 5 PTNs com os 15 homólogos utilizados durante os experimentos de validação, além dos dados gerados artificialmente pelo *INDELible*. Valores relativos ao tamanho das sequências das amostras. Número de sequências (# Seqs) em cada conjunto de dados, mediana, média, desvio padrão (Desv), valor mínimo (Mín) e máximo (Máx) dos tamanhos dessas sequências. Por último valor mediano de LCS de cada conjunto de dados.

Em relação às sequências, tanto de nucleotídeos quanto de aminoácidos, consegue-se observar que as PTNs *Hemoglobin*$_\beta$ e *Myoglobin* possuem nenhuma variação de tamanho, tendo valores fixos de 441 e 465 pb em DNA, e 147 e 154

aa como PTN, respectivamente, sem desvio padrão (Desv). Enquanto isso, a diferença entre o tamanho das sequências aumenta gradativamente para *Neuroglobin* (451,6 pb e 149,5 aa), *Cytoglobin* (596,2 pb e 197,9 aa) e *Androglobin* (4726,4 pb e 1574,9 aa), assim como seu desvio padrão sendo, respectivamente, 11,43, 66,25 e 694,56 pb; e 4,61, 22,10, 231,31 aa. Além do tamanho aumentar gradativamente, também aumenta a variação de tamanho mínimo e máximo das sequências, sendo mais um complicador para comparação entre elas. Os dados artificiais do *INDELible* não possuem variação de tamanho e pouquíssima similaridade de LCS dentro deles (11 pb e 5 aa).

Dentro dos conjuntos de dados há também sequências que são realmente iguais entre as diferentes espécies. Nas de *Hemoglobin*$_\beta$ e *Myoglobin*, existem duas sequências que são iguais entre as espécies; no conjunto de *Neuroglobin* existem 2 pares de sequências iguais entre as espécies; enquanto na *Cytoglobin*, *Androglobin* e *INDELible* não existem sequências iguais.

Utilizado o alinhamento com o programa Clustal Omega (SIEVERS; HIGGINS, 2018) foram obtidas algumas estatísticas para descrever melhor alguns dados sobre as sequências dos conjuntos de dados. O Clustal é uma das técnicas mais utilizadas de MSA que foi aplicado para cada família de PTNs.

Com o detalhamento das estatísticas de alinhamento descritas na Tabela 3.4a, são observadas pequenas diferenças entre a identidade da *Myoglobin* (0,75) e da *Hemoglobin*$_\beta$ (0,76), mas ainda são as que possuem homólogos mais similares entre si sem nenhum *gap*. O mesmo se repete na Tabela 3.4b mas com uma inversão na identidade, ficando a *Myoglobin* com maior identidade (0,79) que a *Hemoglobin*$_\beta$ (0,77). Esse tipo de inversão pode acontecer porque mais de um códon de nucleotídeos se traduz em um mesmo aminoácido.

Apesar da *Androglobin* ter maior variação entre o tamanho das sequências, conforme citado, possui identidade tão boa quanto a *Neuroglobin* em DNA (0,61 para ambos), mas quando transformada em PTN, a identidade da *Neuroglobin* aumenta para 0,64 e da *Androglobin* diminui para 0,52. Apesar da identidade similar a *Neuroglobin*, a *Androglobin* possui um número muito maior de *gaps* (327,6 DNA e 101,1 PTN) dado a variação de tamanho dos dados. A *Neuroglobin*, por outro lado, quase não possui *gaps* (4,4 DNA e 5,4 PTN), devido ao seu tamanho e a sua identidade. Quando são traduzidas em aminoácidos existe um aumento na quantidade de sequências com *gaps* de 2 para 15, indicando que há mais dificuldade de alinhá-las. A *Cytoglobin*, apesar de possuir uma identidade baixa dentro dos seus homólogos (0,4 DNA e 0,36 PTN), não possui mais *gaps* (87,8 DNA e 38,1 PTN) do que a *Androglobin*. Todas as sequências possuem desvio padrão altos e as médias sendo maiores que as medianas em relação a quantidade de *gaps*, sinalizando variação de *gaps*

Dados	**Id**	**# *Gaps***	**Mediana**	**Média**	**Mín**	**Máx**
Hemoglobin$_\beta$	0,75	0	0	0	0	0
Myoglobin	0,76	0	0	0	0	0
Neuroglobin	0,61	2	0	4,4	0	39
Cytoglobin	0,40	15	66	87,8	6	306
Androglobin	0,61	15	125	327,6	50	2906
INDELible	0	40	3302	3302	3302	3302

(a) Sequências de DNA.

Dados	**Id**	**# *Gaps***	**Mediana**	**Média**	**Mín**	**Máx**
Hemoglobin$_\beta$	0,77	0	0	0	0	0
Myoglobin	0,79	0	0	0	0	0
Neuroglobin	0,64	15	4	5,4	4	16
Cytoglobin	0,36	15	31	38,1	11	111
Androglobin	0,52	15	34	101,1	9	960
INDELible	0	40	1169	1169	1169	1169

(b) Sequências de PTNs.

Tabela 3.4: Resultados do alinhamento com Clustal Omega para os nucleotídeos e aminoácidos das 5 PTNs com os 15 homólogos, utilizados durante os experimentos de validação, além dos dados gerados artificialmente pelo *INDELible*. Identidade (Id) do alinhamento, número de sequências com *gaps* (# *Gaps*), mediana, média, valor mínimo (Mín) e máximo (Máx) da quantidade de *gaps* das amostras.

para valores mais baixos, com poucos *outliers* mais altos, tanto para DNA quanto para PTN.

Como conclusão dessa análise, os conjuntos de dados em DNA e PTN mais simples de avaliar a similaridade são *Hemoglobin*$_\beta$ e *Myoglobin* por terem alta identidade (0,75 DNA e 0,77 PTN; e 0,76 DNA e 0,79 PTN, respectivamente), não terem variação de tamanho, nem *gaps*, e alto LCS (88 pb e 40 aa; 65 pb e 33 aa). Os dados de *Neuroglobin* são intermediários em complexidade, para realização dos casos de uso, por conta da identidade média (0,61 DNA e 0,64 PTN), bom LCS (77 pb e 113 aa), baixa quantidade de *gaps* (4,4 DNA e 5,4 PTN) e variação de tamanho (11,43 DNA e 4,61 PTN). Os dados de *Cytoglobin*, *Androglobin* e *INDELible* são considerados mais complexos, mas por motivos diferentes. A *Cytoglobin* tem menor identidade (0,4 DNA e 0,46 PTN) e muitos *gaps* (87,8 DNA e 38,1 PTN); a *Androglobin* tem uma identidade um pouco melhor (0,61 DNA e 0,52 PTN), mas uma variação de tamanho de sequências muito alta (694,56 DNA e 231,3 PTN); enquanto o *INDELible* apesar de não ter variação de tamanho, não teve também identidade alguma no método de MSA e possui um LCS muito pequeno (11 pb e 5 aa), dificultando a detecção de similaridades.

3.1.2 Dados de Validação

Como conjunto de dados mais extenso para uma validação em larga escala, somente a tarefa de similaridade de sequências foi validada tanto para a modelagem de aminoácidos como nucleotídeos em conjuntos de dados do AFProject (ZIELEZINSKI et al., 2019). Além da restrição de casos de uso, também foram restritos somente alguns algoritmos de similaridade testados por conta do poder computacional necessário.

Para PTNs, foi utilizado o SwissTree [3] com onze conjuntos de dados representando cada família de PTNs e de dezenas a centenas de sequências em cada um dos conjuntos. Cada uma das famílias e quantidade de PTNs nela está ilustrada na Tabela 3.5 extraída do website do projeto AFProject. Analisando as sequências sabemos que possuem uma média de quatrocentos aminoácidos com uma variação de duzentos. Para nucleotídeos foi validado com o FishMito [4], que representa um conjunto de vinte e cinco genomas mitocondriais completos (mtDNA) de peixes com uma média de dezesseis mil e novecentos nucleotídeos por sequência e uma variação de cerca de mil nucleotídeos entre elas.

Id Família	**Nome Família**	**# Seqs**
ST001	*Popeye domain-containing protein family*	49
ST002	*NOX 'ancestral-type' subfamily NADPH oxidases*	54
ST003	*V-type ATPase beta subunit*	49
ST004	*Serine incorporator family*	115
ST005	*SUMF family*	29
ST007	*Ribosomal protein S10/S20*	60
ST008	*Bambi family*	42
ST009	*Asterix family*	39
ST010	*Cited family*	34
ST011	*Glycosyl hydrolase 14 family*	159
ST012	*Ant transformer protein*	21

Tabela 3.5: Dados de PTNs de diversas famílias do AFProject, obtidos pelo SwissProt e o número de sequências de PTNs de cada família (# Seqs).

Obtem-se para validação alguns conjuntos de dados de PTNs com grande quantidade de sequências, mas de tamanhos pequenos e com grande variação entre elas. Por outro lado, para DNAs existe um conjunto menor de dados

[3] <https://afproject.org/app/benchmark/genetree/swisstree/dataset/>
[4] <https://afproject.org/app/benchmark/genome/std/assembled/fish_mito/dataset/>

com sequências bastante grandes e uma variação relativa de tamanho menor. Mesmo com essa variação relativa sendo menor, é maior em termos absolutos do que todos os outros conjuntos de dados testados.

3.2 Método Proposto

O método proposto é composto por uma premissa para modelagem física de dados biológicos moleculares, uma modelagem física extensível dos dados em camadas, utilizando as premissas estabelecidas, e de extratores de características, planejados e exemplificados em PTNs e DNAs. Esse método permite a criação de várias camadas reutilizáveis, preservando a semântica das características das moléculas biológicas, inclusive durante a comparação das suas características. Essas características podem ser usadas de forma padronizada por técnicas de bioinformática ou outros modelos de ML.

3.2.1 Premissa de Modelagem

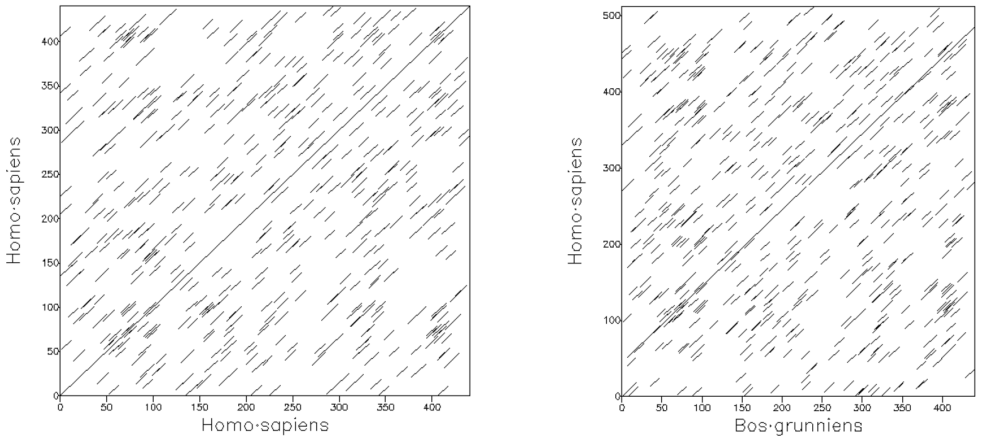
A premissa considera como base os componentes mais básicos das moléculas biológicas, sendo uma sequência de nucleotídeos e de aminoácidos modelados nos contextos de genomas (DNA), RNA e PTN. Diferente do que costuma ser encontrado na literatura, essa premissa permite adicionar e realçar as características desses componentes mais básicos em uma abordagem *bottom-up*, explicitando a importância de um nucleotídeo ou aminoácido na sequência como um todo e identificando interações e similaridades dentro das sequências.

Cada sequência biológica molecular pode ser de DNA, RNA ou PTN e essa sequência tem vários compostos moleculares representados respectivamente por desoxirribonucleotídeo, ribonucleotídeo e aminoácidos, que serão referidos como "blocos". Cada um desses blocos pode estar em uma posição dentro de uma sequência e ter características interacionais, morfológicas ou funcionais. Como características interacionais, pode-se incluir: o quão custoso é substituir um bloco pelo outro, ou alguma medida de correlação entre as posições dos blocos. As morfológicas são relacionadas a alguma estrutura terciária (alfa-hélice, folha beta, etc.) secundária, ou até propriedades físico-químicas dos blocos. As funcionais representam se aquele bloco está presente em algum processo, redes ou função biológica. Acredita-se que essa premissa pode ser extensível a outras características, mas essas de cunho sequencial foram as mapeadas e consideradas mais importantes neste trabalho.

A importância dessa premissa é a designação de características para as estruturas mais granulares das moléculas biológicas, tornando possível a confecção de modelagens físicas, tanto nos blocos fundamentais das sequências, como na molécula como um todo.

3.2.2 Modelagem Física em Camadas

A inspiração da modelagem física foram as imagens, que são normalmente utilizadas com técnicas aplicadas em biologia e medicina (LATIF et al., 2019; KAN, 2017; WÄLDCHEN; MÄDER, 2018; BERG et al., 2019). Em biologia, existe uma estrutura muito utilizada que se utiliza de imagens para mostrar similaridades internas das sequências através de imagens com pontos (*dot plots*). A representação de sequências em *dot plots* é uma técnica bastante antiga de comparação visual de moléculas, mas que ainda continua sendo aprimorada (SEIBT; SCHMIDT; HEITKAM, 2018). Recentemente tem se mostrado que é possível distinguir moléculas através de um padrão topológico nelas (COLLETT; PEARCE, 2021) ainda exploradas de forma visual. Na Fig 3.1 foram feitos pelo software Dotmatcher [5] *dot plots* de sequências de DNA de cadeias *Beta* de Hemoglobina, sendo em A uma autocomparação da sequência de DNA da PTN humana e em B uma comparação entre a PTN humana e a mesma de um iaque (intercomparação).



(a) Autocomparação da sequência humana.

(b) Intercomparação entre sequência humana e iaque.

Figura 3.1: *Dot plot* entre cadeias *Beta* de Hemoglobina.

Em um *dot plot*, quando uma base de uma sequência é igual a da outra, um ponto é marcado. Com isso, é possível obter linhas em sequências que

[5] <https://www.bioinformatics.nl/cgi-bin/emboss/dotmatcher>

possuem mesmos dados, sendo inclusive possível observar uma linha diagonal dividindo a figura nas autocomparações. As intercomparações são comparações em que uma sequência fica na base horizontal e outra na vertical, sendo assim, não necessariamente formando uma figura quadrada. Para evitar muita poluição visual, é possível escolher um número de elementos da sequência para serem comparados como uma janela deslizante de comparação (Ex: de 3 em 3 elementos) e o quão similares as janelas precisam ser para ser considerado um ponto, conhecido como *threshold* estabelecido.

O convencional para representação dos *dot plots* são imagens em escala de cinza, mas as imagens são elementos que podem ser representados de algumas formas. Uma forma convencional é através do padrão RGB (*Red*, *Green*, *Blue*) para imagens coloridas, ou escala de cinza, para imagens em preto e branco. Dessa forma, a imagem possui até 3 matrizes de pixels, que são chamadas de canais (RGB), e que recebem valores de 0 até 255, dependendo de quão intensa seja a cor. Assim, quando os pixels que estão na mesma posição em todos os canais são 0 a imagem é preta e quando todos os pixels possuem valor 255, é branca. Imagens coloridas são formadas através da combinação de intensidade de cor dessas três camadas e são consideradas tensores por representarem os dados como múltiplas matrizes.

Ao visualizar um *dot plot*, conforme ilustrado na Fig 3.1, é possível de observar alguns padrões, como: regiões mais densas, áreas com uma linha reta maior que outras, regiões com poucas linhas. Esses padrões perceptíveis visualmente são a semântica interna daquela sequência que podem ter a sua detecção automatizada através de algoritmos de ML.

No modelo físico criado, foram feitas matrizes de autocomparação de uma sequência, adicionando uma numeração na matriz, sempre que tivesse uma correspondência de blocos para evidenciar uma determinada característica, tal qual em uma matriz normal de autocomparação faz com a correspondência entre blocos iguais na Fig 3.1. Para criar cada canal da imagem foram criadas matrizes com pixels de tamanho (N, N) nucleotídeos ou aminoácidos, onde o valor do pixel é dado de acordo com a heurística de correspondência conforme ilustrado no Alg 1. Com isso, a representação obtida respeita a premissa estabelecida e consegue representar cada bloco molecular.

Através dessa representação, é possível criar diversas matrizes de características ou juntar cada uma das matrizes em um grande tensor que utilize todas as matrizes de uma vez, não precisando ficar limitado a três, que é a quantidade de canais das imagens, ou a 255 que é a quantidade máxima do valor de pixel de uma imagem. As camadas dessa modelagem física simplificam o reuso, porque cada camada funciona como uma característica que se

Algoritmo 1: Pseudocódigo utilizado para criar canais através de comparação de sequências com complexidade $O(tamanho_1 \times tamanho_2)$.

Entrada: seq_1, seq_2
Saída: Matriz (Canal) com pixels em valor fixo ou zero

$tamanho_1 \leftarrow Size(seq_1)$

$tamanho_2 \leftarrow Size(seq_2)$

$$canal \leftarrow \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \\ 0 & & 0 \end{bmatrix}_{(tamanho_1 \times tamanho_2)}$$

para $t_1 = 0$, t_1++, $t_1 \leq tamanho_1$ **faça**
    para $t_2 = 0$, t_2++, $t_2 \leq tamanho_2$ **faça**
        $pixel_{val} \leftarrow Corresp(seq_1[t_1], seq_2[t_2])$ *Preencha com* $pixel_{val}$ *de acordo com a correspondência.*
        $canal[t_1, t_2] = pixel_{val}$

retorna *canal*

deseja representar, estando cada uma diretamente relacionada com um canal modelado.

Durante o trabalho, optou-se por estipular e normalizar os valores colocados nas camadas entre 0 e 255 para facilitar a visualização e comparação das matrizes, mas não é obrigatório para que a representação funcione. A única necessidade para utilizar essas imagens é a construção de métodos que consigam compará-las ou utilizar métodos existentes para a extração das características delas.

3.2.3 Extração e Comparação de Características

Existem várias técnicas (TIAN, 2013) utilizadas em diferentes aplicações que usufruem da comparação de imagens, como: reconhecimento de faces, detecção de objetos, análise de plágio, dentre outras. Dentro dessas técnicas, existe a de geração de *embedding*, que representa algum elemento como: texto, imagem, vídeo ou qualquer outra coisa, representando suas características dentro de um espaço vetorial.

Através desse *embedding* é possível agrupar os elementos de acordo com a proximidade deles no espaço vetorial. Um exemplo bastante comum de representação de *embedding* é o gerado por algoritmos de processamento de texto como o *Word2Vec* (CHURCH, 2017). Com ele, se torna possível realizar operações matemáticas com os vetores obtidos a partir de palavras dentro

de um contexto conforme Fig 3.2. Nessa figura, ao representar a palavra *man*, *woman*, *king* e *queen* em *embeddings*, é possível realizar essa operação matemática e concluir que a divisão de *man* por *woman* é igual à divisão de *king* por *queen*.

$$\frac{man}{woman} = \frac{king}{queen}$$

Figura 3.2: Matemática com palavras, utilizando *Word2Vec*, extraído de (CHURCH, 2017).

A modelagem de dados proposta pode ser comparável à geração de um *embedding* da sequência, mas com a diferença que os *embeddings* são padronizados em um tamanho fixo e não possuem qualquer semântica, visto que são representações numéricas de elementos que previamente tinham semântica. Neste trabalho, foi aplicada uma técnica de *embedding* em cima da representação produzida para extração das suas características e, depois, mensurada a similaridade entre os *embeddings* gerados.

Para criar esses *embeddings*, normalmente utilizam-se modelos de *deep learning* (DL) pré-treinados em tarefas de classificação para passar a imagem nessa rede neural e deixar que a própria rede faça as operações matemáticas adequadas para gerar um vetor que represente aquela imagem ou vídeo (KIELA; BOTTOU, 2014). A única diferença do modelo que faz o *embedding* para o que faz a classificação em si, é que para obter o vetor de *embeddings*, deve-se ignorar a última camada relativa à classificação e pegar o vetor resultante na penúltima camada.

O algoritmo chamado de *Deep Search* neste trabalho usa a distância de *embeddings* através do buscador Annoy [6] para buscar sequências similares usando a representação de dados proposta na tese. Para gerar esses *embeddings*, precisa-se transformar a imagem em um vetor de características (*embedding*) de alguma forma, e a escolhida aqui foi através de algum modelo de classificação de imagens. Depois disso, é necessário indexar no Annoy os *embeddings* gerados e aplicar o algoritmo de *Approximate Nearest Neighbors* (ANN) (BEIS; LOWE, 1997) conseguindo *scores* de similaridades pelos itens buscados. O modelo de classificação usado foi o VGG16 (SIMONYAN; ZISSERMAN, 2014) com pesos pré-treinados do *ImageNet* [7]. O VGG16 é um modelo convolucional que consegue extrair as características de uma imagem utilizando várias camadas de convoluções na imagem de entrada e aglomerando essas características até condensar tudo em um vetor final.

[6] <https://github.com/spotify/annoy>
[7] <https://www.image-net.org/>

A técnica de ANN consiste em calcular a similaridade aproximada entre vetores. No caso do Annoy, tudo que vai ser indexado é colocado em um hiperplano. Depois divide o hiperplano, utilizando k cortes ao escolher aleatoriamente dois itens e traçar uma reta no meio entre os dois pontos, dividindo o espaço vetorial entre eles. Quando o índice é criado, escolhe-se o número t de árvores que terá, fazendo com que armazene, para cada nó de árvore, os pontos que estão de um lado ou do outro da reta. No fim, compõe uma árvore binária com profundidade k. Cada uma das árvores vai utilizar k cortes aleatórios, aumentando a chance de encontrar uma distância mais acurada. O valor de k é especificado ao fazer uma busca por k itens mais próximos. Cada corte nesse hiperplano, gera uma área e uma vez que você queira buscar por um novo elemento, basta caminhar pelas árvores e encontrar, no fim, os nós com mais similaridades (que estão na mesma área). Uma vez encontrando os nós mais similares, basta calcular a distância até eles e encontrar o mais similar conforme a Fig 3.3.

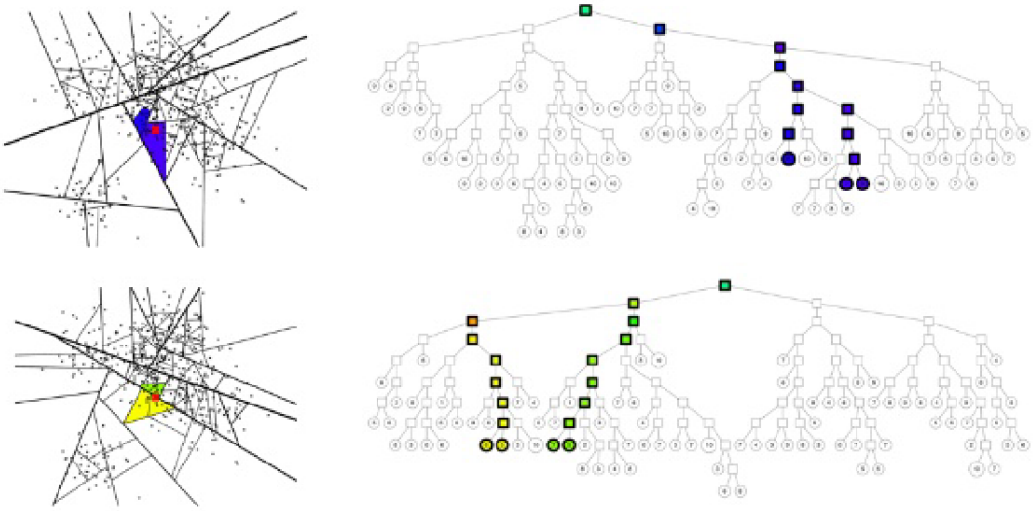


Figura 3.3: Imagem extraída do trabalho (WOLFF, 2016), mostrando a divisão em hiperplanos e como a árvore encontra a região mais próxima.

Outras técnicas de extração e comparação de características, também exploradas neste trabalho, foram feitas através dos algoritmos baseados em comparação estrutural, que tentam validar se imagens são similares utilizando propriedades perceptíveis pela visão humana, tal qual é usado manualmente no *dot plot*. Elas foram melhor exploradas aqui, porque mostraram ser melhores do que outras técnicas para medir similaridade utilizando imagens de CGR (KARAMICHALIS et al., 2015). Além do SSIM (WANG et al., 2004; BAKUROV et al., 2022), foi utilizado também seu precursor *Universal Quality Index* (UQI) (WANG; BOVIK, 2002) e o *MultiScale Structural Similarity Index Measure* (MS-SSIM) (WANG; SIMONCELLI; BOVIK, 2003), que é uma

variação do SSIM, comparando as imagens com diferentes focos ao invés de só um. Esses três algoritmos serviram como base para a implementação de algoritmos de similaridade focados em comparação de sequências biológicas, utilizando as representações de dados.

Todos os algoritmos de comparação estrutural utilizam um quadrado de lado igual a $filter\_size$ similar a uma convolução para comparar uma imagem com a outra e se aproveitar do contexto desse quadrado, evitando olhar a imagem inteira de uma vez só. Com isso, cada imagem é quebrada em N quadrados de tamanho $filter\_size$ e cada quadrado é comparado com o correspondente da outra imagem. Assim, o quadrado N da primeira imagem será comparado com o quadrado N da segunda utilizando o algoritmo de comparação estrutural (o mesmo vale para $N-1, N-2, N-3, \ldots$). Depois disso, os algoritmos fazem uma média de todos os valores obtidos nos algoritmos aplicados para os N quadrados, compondo o resultado final. O valor de $filter\_size = 11$ foi definido para o tamanho do quadrado em DNA e $filter\_size = 4$ para PTN em todos os algoritmos, uma vez que 11 era padrão da implementação do SSIM e do MS-SSIM utilizados. O valor menor em PTNs foi definido dividindo o $filter\_size$ por três e arredondado por conta de três nucleotídeos codificarem um aminoácido.

O interessante desses métodos de comparação de imagem é que eles já possuem embutidos um filtro gaussiano ($filter\_sigma$), que é utilizado para remover ruídos das imagens, evitando que os métodos comparem falhas na imagem. Com isso sendo aplicado nas representações feitas através da modelagem física, já são removidos possíveis ruídos que existam nas sequências biológicas e nas características modeladas.

(a) Comparação com redimensionamento. (b) Comparação com diagonal.

Figura 3.4: Formas de comparar similaridade entre imagens utilizando comparação estrutural.

Diferente da técnica de *embedding* que força a representação de todos os vetores em um tamanho fixo, as técnicas utilizando comparação estrutural, exploradas para medir similaridade, precisam comparar dois tensores (ou imagens) do mesmo tamanho, mas não necessariamente forçando que todos sejam do mesmo tamanho, e sim, somente o par de tensores comparados. Para

isso, neste trabalho, foram utilizadas duas abordagens diferentes para comparar imagens de tamanhos distintos: ou aumentando o tamanho da imagem menor para ficar igual à imagem maior através de redimensionamento bi-cúbico; ou comparando partes da imagem menor com partes da imagem maior através da diagonal como ilustrado na Fig 3.4.

Outro aspecto interessante dos métodos de comparação estrutural é que eles calculam as similaridades por camada da imagem, sendo simples de compreender tanto a colaboração de uma camada, quanto da combinação de várias delas para o resultado obtido. A característica desses métodos vai colaborar para a explicabilidade mostrada nos resultados, permitindo a extração de métricas por camada e preservação de sua semântica durante as comparações.

3.2.4 Métricas de Avaliação

As representações de dados e algoritmos de similaridade propostos foram testados em diversas tarefas comuns da bioinformática para avaliar se conseguiam boa performance, resolvendo-as da melhor maneira. Os resultados obtidos nos casos de uso que serão mostrados a seguir, foram medidos com métricas de avaliação aplicadas de acordo com cada um deles. No caso da tarefa de busca de homólogos, a métrica de avaliação foi *Mean Average Precision* (MAP) (ALHIJAWI; AWAJAN; FRAIHAT, 2023), bastante comum para medir algoritmos de recuperação de informação, validando a ordenação dos itens retornados. Ao dar uma sequência de entrada para o algoritmo, as sequências mais similares recebidas devem ser relacionadas àquelas utilizadas na entrada. Os resultados dessa métrica são normalizados entre zero e o número total de sequências retornadas, sendo assim, os resultados mais próximos de um indicam que um maior número de sequências parecidas foram retornadas. O MAP foi calculado manualmente, replicando sua fórmula nos códigos dos experimentos.

Para validar o caso de uso de agrupamento de homólogos e a similaridade de homólogos, foram comparados os dendrogramas gerados para os grupos de sequências. Os dendrogramas vão mostrar a similaridade entre as sequências, quão mais próximas elas estão pelos seus ramos. Para avaliar esses dendrogramas, foi utilizada a métrica de Robinson-Foulds (RF) (ROBINSON; FOULDS, 1981), normalizada entre zero e um. Essa métrica compara a quantidade de passos necessários para um dendrograma virar outro e é normalizada pelo número total de passos possíveis. Outra métrica importante para avaliar as arestas e estruturas mais próximas das folhas do dendrograma é o *Branch Congruence*

Measure (BCM) (HUERTA-CEPAS; SERRA; BORK, 2016), que compara a quantidade de arestas iguais existentes entre os dendrogramas, por mais que estejam distantes uma das outras. Isso indica que mesmo que o dendrograma global não seja parecido, as suas folhas foram agrupadas corretamente. Para que a métrica BCM ficasse na mesma escala do RF em que o zero é o melhor resultado, ela foi invertida. Com isso, $BIM = 1 - BCM$ refletindo o nível de incongruência dos dendrogramas, que agora será referenciada no trabalho como *Branch Incongruence Measure* (BIM). Ambas as medidas de distância foram utilizadas pelo pacote filogenético ETE 3 (HUERTA-CEPAS; SERRA; BORK, 2016).

As comparações dos dendrogramas gerados através de cada um dos algoritmos apresentados neste trabalho foram feitas contra os dendrogramas gerados por controles. Os controles foram feitos através de alinhamentos do Clustal Omega, alinhamentos locais pelo algoritmo de Smith–Waterman (SW) (SMITH; WATERMAN, 1981) e pelos alinhamentos globais com NW. Cada controle será descrito na próxima seção e sinalizado devidamente nos resultados mostrados no Capítulo 4.

3.3 Casos de Uso em Bioinformática

Uma vez que foi construída uma nova forma de representar as moléculas biológicas e foram desenvolvidos algoritmos que conseguem extrair a informação de quais sequências são similares, abre-se a possibilidade de utilizar esses componentes em conjunto para resolver problemas da bioinformática. Uma forma bastante comum de resolver esses problemas é através de diversos programas distintos, específicos para cada fim. Neste trabalho são utilizados dois programas bastante comuns em bioinformática para validar os casos de uso: o Clustal Omega e o BLAST.

3.3.1 Similaridade de Homólogos

O método aplicado de medir a similaridade de homólogos basicamente se resume em utilizar algoritmos de similaridade produzidos na tese nas representações de dados propostas neste trabalho. Essa abordagem foi aplicada separadamente em cada conjunto de dados experimentais descritos na Seção 3.1.1, assim como nos de validação da Seção 3.1.2. A partir do momento que isso é feito dentro de um conjunto de dados, resulta em uma matriz de similaridades de uma sequência com a outra. Se essa matriz for invertida para utilizar a dissimilaridade, basta aplicar $1 - similaridade$ em cada um dos resultados. No

caso, a matriz de dissimilaridade é utilizada pelo algoritmo de agrupamentos hierárquicos *Neighbor Joining* (NJ) e o dendrograma gerado é comparado com o do Clustal Omega através das métricas de RF e BIM. O algoritmo NJ é bastante utilizado no desenvolvimento de dendrogramas para filogenia por alinhamentos (SAITOU; NEI, 1987) e foi utilizado neste trabalho para facilitar a comparação de similaridade dentro de um conjunto de dados. Quando os dendrogramas estão iguais, é um sinal de que os resultados estão similares ao Clustal Omega, utilizando todos os parâmetros padrões.

3.3.2 Busca de Homólogos

Tendo as dissimilaridades e similaridades entre todas as sequências indexadas, dentro de cada conjunto de dados, só se faz necessário calcular as dissimilaridades e similaridades de sequências entre conjuntos de dados distintos. Em seguida, são buscadas pelas k sequências mais similares a uma determinada sequência-alvo e mensurado, através do MAP, quantas dessas sequências correspondem ao mesmo conjunto de dados da sequência-alvo. O k escolhido sempre é igual ao número de sequências no conjunto de dados da sequência-alvo, ou seja, se tiverem 15 sequências no conjunto de dados, o k será 15. Para calcular o MAP final, é feita a média de todos os valores de MAP calculados para cada sequência do conjunto de dados.

Do lado do controle, foi utilizado o programa BLASTn para os nucleotídeos e o BLASTp para os aminoácidos. Visando uma comparação justa do $filter\_size$ dos algoritmos propostos com o $word\_size$ do BLAST, foi utilizado $word\_size = 11$ no BLASTn e $word\_size = 4$ no BLASTp para indexar todos os conjuntos de dados em uma base local. O $word\_size$ é a janela de comparação que o BLAST vai utilizar entre as sequências para retornar às mais similares. Para conseguir todos os resultados no BLAST de forma única para cada sequência-alvo, foram utilizados os parâmetros $evalue = 10000$ e $max\_hsps = 1$. Uma vez que a busca é ordenada por $bitscore$, basta calcular o MAP dos k primeiros resultados para validar se estão dentro do grupo de dados correto.

Tanto a busca de homólogos, quanto a próxima tarefa de agrupamento foram feitas utilizando somente a combinação de todas as camadas, chamada de *full* no trabalho, visando validar as representações de dados completas. A representação *full* será melhor detalhada na Seção 4.1.

3.3.3 Agrupamento de Homólogos

Dado que já foram calculadas todas as dissimilaridades e similaridades intra e interconjuntos de dados, foi utilizada a mesma metodologia da Seção 3.3.1 para o agrupamento dos homólogos. Agora, todos os conjuntos de dados estão juntos para serem agrupados por similaridade usando os métodos propostos em relação ao agrupamento feito com o Clustal. O método é validado através de RF e BIM, mas outra forma de comparação utilizada é a validação visual dos agrupamentos de cada conjunto de dados em relação aos outros através das cores.

3.3.4 Otimização de Parâmetros

Foi aplicada a otimização bayesiana utilizando a biblioteca Optuna (AKIBA et al., 2019) nos algoritmos com as representações compostas por todos os canais (*full*), visando encontrar os valores de RF mais próximos dos feitos pelo alinhamento com Clustal, através do caso de uso de similaridade de homólogos. Basicamente, a técnica de otimização bayesiana busca, dentro de um conjunto de possibilidades, qual a melhor combinação de parâmetros para otimizar uma função-objetivo, e nesse caso, o objetivo era alcançar, com os algoritmos, o mesmo RF do obtido com o Clustal.

3.3.5 Recursos Computacionais Utilizados

Para executar a otimização e os experimentos, foi utilizado o ambiente do grupo de pesquisa com 8 núcleos de processamento e 32GB de memória RAM, sendo que nenhum experimento excedeu a quantidade de memória de 8GB. A única exceção foi na validação do conjunto de dados FishMito da Seção 3.1.2, visto que as sequências de DNA eram de vários genomas completos e foi necessária uma instância EC2 (*Elastic Compute Cloud*) na AWS (*Amazon Web Services*) [8] com 32 núcleos de processamento e 64GB de RAM para deixar o processo mais rápido.

Todos os experimentos foram produzidos utilizando Python 3.7 ou superior e automatizados para serem executados em imagem Docker [9] disponível para download. O código está modularizado e disponível no Github [10] com todas as suas dependências bem especificadas. Os dados resultantes dos expe-

[8] <https://aws.amazon.com>
[9] <https://github.com/BioBD/biomodelml/pkgs/container/biomodelml>
[10] <https://github.com/BioBD/biomodelml/>

rimentos estão salvos no Google Drive, versionados pela ferramenta DVC [11], para garantir completa reprodutibilidade de tudo.

[11] <https://dvc.org/>

4
Resultados e Discussão

Neste capítulo são apresentados os resultados obtidos a partir de representações feitas através do modelo físico, com as premissas propostas nesta tese para nucleotídeos e aminoácidos. Como essas representações são inovadoras e necessitam de formas de compará-las para extrair suas características, são apresentados métodos para esse fim.

Em resumo, na Seção 4.1 são apresentadas as representações de dados criadas para nucleotídeos em 4.1.1 e aminoácidos em 4.1.2; em seguida, na Seção 4.2, são introduzidos métodos para possibilitar a comparação entre as representações. Esses métodos são divididos entre métodos tradicionais de comparação de imagens, somente com redimensionamento na Seção 4.2.1, e os novos algoritmos propostos nesta tese na Seção 4.2.2. Para finalizar a Seção 4.2, foi explorada a possibilidade de otimização dos novos algoritmos visando melhorar sua performance em 4.2.3. Por fim, na Seção 4.3 é proposto um novo método *alignment-free* utilizando as representações e os algoritmos de similaridade, testado em casos de uso da bioinformática em 4.3.1 e em dados abertos do AFProject em 4.3.2.

4.1
Nova Representação de Dados

Para validar o modelo físico proposto, foram desenvolvidas com ele uma representação de dados para nucleotídeos e outra para aminoácidos. Ambas as representações utilizam três matrizes, que vão ser citadas como camadas R, G e B das imagens apresentadas nas próximas seções e as combinações das 3 camadas vai ser chamado de *full* durante o trabalho. Logo, uma imagem com camadas R, G e B é um sinônimo para representação *full*. Este trabalho, foi limitado a três camadas, para simplificar a obtenção de resultados com adaptações mais fáceis dos algoritmos que existem para imagens, mas que podem ser expandidos para a quantidade necessária de camadas.

4.1.1
Representação para Nucleotídeos

A representação de nucleotídeos considerou a existência de sinais evolutivos dentro da sequência de DNA ou RNA e como é possível explicitá-los com mais facilidade. Com isso, foram utilizadas características morfológicas e interacionais que representam esses sinais evolutivos. Por padrão biológico,

quando não se tem certeza do nucleotídeo em uma sequência, é adicionado um caractere de ambiguidade diferente de A (Adenina), T (Timina), C (Citosina), G (Guanina) e U (Uracila). Nesta tese, foi colocado o menor valor possível na matriz (zero) sempre que apareciam ambiguidades. Somente nos resultados de validação da Seção 4.3.2 aparecem ambiguidades e não nos de teste dos casos de uso em 4.3.1. A representação está exemplificada na Fig 4.1 e cada camada será melhor descrita a seguir.

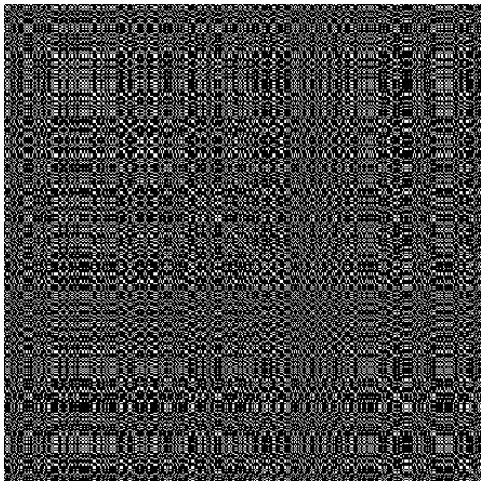
	Sequencial (R)				**Complementar (G)**				**Diferenças (B)**			
	G	**T**	**T**	**A**	**C**	**A**	**A**	**T**	**G**	**T**	**T**	**A**
G	255	0	0	0	0	0	0	0	0	255	255	255
T	0	255	255	0	0	0	0	255	255	0	0	0
T	0	255	255	0	0	0	0	255	255	0	0	0
A	0	0	0	255	0	255	255	0	255	0	0	0

Figura 4.1: Demonstração de uma representação física preenchida para nucleotídeos. Na camada R, as comparações sequenciais, na G as comparações da sequência com a complementar dela e na B as diferenças que não estão nem na R e nem na G.

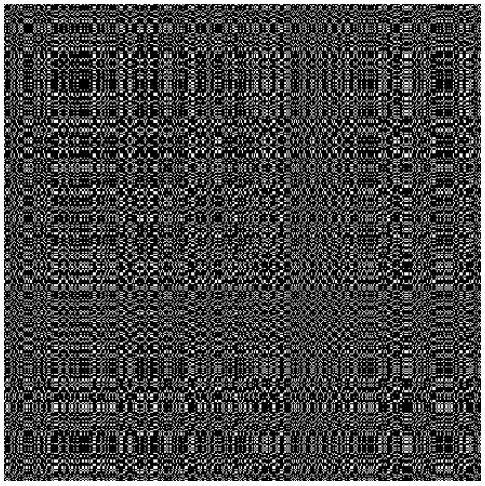
Na camada R, entraram as correspondências diretas entre os nucleotídeos, então, sempre que um nucleotídeo da sequência comparada contra ela mesma for igual, o valor máximo de 255 das imagens é estipulado. A importância biológica da camada R é que repetições diretas de subsequências dentro de uma sequência são um potencial sinal evolutivo conhecido. O valor de 255 foi escolhido simplesmente por ser o mais aparente nas imagens finais, mas não impacta nos resultados com algoritmos de similaridade escolhidos. Em modelagens preliminares, foram testadas escalas de importância de acordo com o quanto de nucleotídeos similares existem em sequência (na diagonal), mas não são práticos de gerenciar. O problema com escalas desse tipo é que se uma sequência menor for comparada com uma sequência maior, a escala tende a ficar maior na de maior tamanho e a normalização não necessariamente é justa. Como colocar um valor fixo não influencia no resultado final, a adição de escalas pode ser planejada futuramente.

Na camada G, entraram as correspondências entre uma sequência e sua sequência complementar. No exemplo da Fig 4.1, a sequência $GTTA$ se transformou em $CAAT$, dado que o nucleotídeo C é o complementar de G, assim como A é complementar de T e vice-versa. A importância biológica da camada G é a detecção de repetições invertidas de subsequências dentro da sequência. O uso da complementar reversa normalmente é feito para detectar essas repetições, porque o DNA tem dupla hélice que se ligam de forma

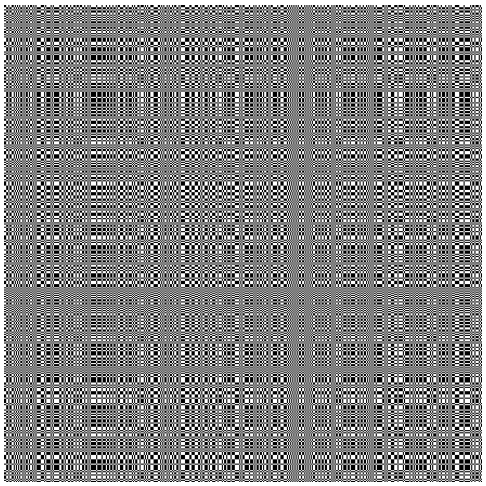
invertida. Com isso, o início de uma fita de DNA se liga com o fim da outra e o último nucleotídeo de uma se liga ao primeiro, sendo complementar a ele. O mesmo sinal evolutivo é capturado tanto usando somente a complementar, como usando a complementar reversa, sendo simplesmente importante manter um padrão entre todas as representações de camada G que são comparadas. Por fins visuais, dado que onde o sequencial é preenchido com 255 (R), não é preenchido na complementar (G) conforme ilustrado na Fig 4.2; e por esse formato colaborar com as outras camadas, que será explicado a seguir, optou-se por usar a camada G como complementar.



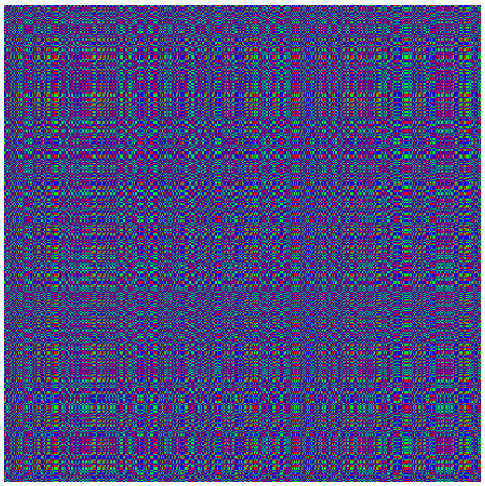
(a) Camada R, repetições ou duplicações.



(b) Camada G, repetições invertidas.



(c) Camada B, substituições.



(d) Representação agregada com as três.

Figura 4.2: Resultado de cada camada desenvolvida com sinais evolutivos para nucleotídeos na sequência de DNA humano da Hemoglobina$_\beta$.

Por último, na camada B, as diferenças das outras duas camadas. Logo, se não houve pontuação na camada R e G, então, entra na B. Com isso, é adicionado um peso para as diferenças e para os nucleotídeos que

não são nem complementares e nem o próprio nucleotídeo na sequência. Esse aspecto valida se realmente as diferenças são mais importantes que as semelhanças ao representar e comparar sequências. Essa camada evidencia possíveis substituições, como no caso de ter na sequência a Adenina em que seu complementar é Timina ou Uracila, que está representado na camada G. Sendo assim, o que aparece na camada B são correspondências com Guanina ou Citosina. Utilizando o mesmo exemplo anterior, poderia ter sido feita uma nova camada para diferenciar entre Guanina e Citosina como sendo as diferentes purinas e pirimidinas, respectivamente. As substituições de bases também são sinais evolutivos das moléculas biológicas, e optou-se por manter nessa camada uma informação mais abrangente de todas as outras possíveis substituições de nucleotídeos que já não estivessem sendo capturadas pelas camadas R e G. Inclusive, que na Fig 4.2 a camada B, não possui uma reta na diagonal clara por conta dessas várias substituições.

Na Fig 4.2 estão ilustradas cada camada desenvolvida na sequência de nucleotídeos da Hemoglobina$_\beta$ humana. As imagens ficaram como um tartã, com várias listras e regiões de coloração mais densa. Ainda nessa figura, existe a camada R com as duplicações e repetições; na G, as repetições invertidas e; na B, as outras substituições. Na descrição e elaboração das camadas, elas são criadas para terem um significado evolutivo sozinhas e em conjunto representarem outras informações. Alguns exemplos dessas informações são as substituições por outros nucleotídeos, assim como regiões de baixa complexidade, em que existe pouca variação de bases nitrogenadas. Essas regiões vão estar presentes nas três camadas e normalmente são quadrados ou retângulos nas imagens porque possuem repetição de nucleotídeos.

4.1.2 Representação para Aminoácidos

Na representação de aminoácidos, estão sendo utilizadas algumas medidas um pouco diferente do convencional em relação ao que foi feito em nucleotídeos. Ainda existem camadas com características morfológicas e interacionais, mas se aproximam das características funcionais através das camadas representadas. A representação está exemplificada na Fig 4.3 e cada camada será melhor descrita a seguir.

As matrizes de aminoácidos vão levar em consideração todos os 20 aminoácidos: Alanina (A), Arginina (R), Aspartato (D), Asparagina (N), Cisteína (C), Fenilalanina (F), Glicina (G), Glutamato (E), Glutamina (Q), Histidina (H), Isoleucina (I), Leucina (L), Lisina (K), Metionina (M), Prolina (P), Serina (S), Tirosina (Y), Treonina (T), Triptofano (W) e Valina (V).

Além disso, existem também os caracteres de ambiguidade, que são tratados nas camadas G e B, deixando valor mínimo (zero) na matriz, caso apareçam. Os caracteres de ambiguidade são letras que podem significar mais de um aminoácido. Lembrando, novamente, que somente em dados de validação, existem caracteres de ambiguidade.

	ProtSub (R)				**Sequencial (G)**				***Sneath* (B)**			
	A	**F**	**F**	**W**	**A**	**F**	**F**	**W**	**A**	**F**	**F**	**W**
A	120	45	45	15	255	0	0	0	255	101	101	42
F	45	120	120	75	0	255	255	0	101	255	255	178
F	45	120	120	75	0	255	255	0	101	255	255	178
W	15	75	75	255	0	0	0	255	42	178	178	255

Figura 4.3: Demonstração de uma representação física preenchida para aminoácidos. Na camada R, os valores de substituição da ProtSub; na G, as comparações da sequência com ela mesma e; na B, os valores de similaridade entre aminoácidos, baseados no *Sneath's Index*.

A camada R foi utilizada para mostrar a frequência de substituição entre aminoácidos de uma molécula. Ela foi a única camada que utilizou valores dos caracteres de ambiguidade diferentes de zero, dado que a matriz de substituição ProtSub (JIA et al., 2021) comporta esses valores. A ProtSub foi utilizada em preferência à BLOSUM62, mesmo que a última seja uma matriz de substituição mais popular, porque a ProtSub foi obtida a partir da similaridade 3D das proteínas. Ela utiliza uma metodologia diferente da BLOSUM62, que obtém a matriz através de blocos de alinhamento para identificar os valores de substituição dos aminoácidos. Além disso, várias matrizes de substituição (TRIVEDI; NAGARAJARAM, 2020) têm surgido como alternativa, principalmente por causa do problema na BLOSUM62 que depois de corrigido, piorou resultados anteriores de pesquisas (STYCZYNSKI et al., 2008). A matriz de substituição ProtSub possui um valor de substituição para cada aminoácido e esse valor foi normalizado entre 0 e 255 por conta da representação. Para cada correspondência que ocorre, um valor de frequência de substituição é dado de acordo com a pontuação da matriz normalizada.

Na camada G, a mesma abordagem da camada R dos nucleotídeos foi utilizada, buscando as duplicações ou repetições nas sequências. Basicamente, foi desenvolvida essa camada para identificar mais claramente quem são os aminoácidos iguais dentro da sequência, assim como padrões de repetições dentro das PTNs. Esses padrões podem ser entendidos tanto como sinais evolutivos, como estruturas similares de aminoácidos dentro de uma mesma PTN, visando desempenhar alguma função ou enovelamento específico.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	4	-2	-2	-2	1	-1	0	0	-1	-1	-1	-1	-1	-1	-1	1	0	-3	-3	0	-2	-1	0	-4
R	-2	6	0	-2	-4	1	0	-2	1	-3	-2	2	-1	-2	-2	0	-1	-2	-2	-2	-1	0	-1	-4
N	-2	0	7	1	-3	2	-1	0	1	-3	-4	0	-2	-3	-2	1	0	-4	-2	-3	3	0	-1	-4
D	-2	-2	1	9	-3	0	3	-1	1	-3	-4	-1	-3	-3	-1	1	-2	-4	-3	-3	4	1	-1	-4
C	1	-4	-3	-3	14	-2	-3	-3	-3	-2	-1	-3	-1	-2	-3	-1	-1	-1	-2	-1	-3	-3	-2	-4
Q	-1	1	2	0	-2	4	1	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	0	-2	0	3	-1	-4
E	0	0	-1	3	-3	1	4	-3	0	-4	-3	0	-2	-4	-1	-1	-4	-3	-2	-2	1	4	-1	-4
G	0	-2	0	-1	-3	-2	-3	9	-3	-4	-4	-2	-3	-4	-2	0	-2	-2	-3	-3	-1	-2	-1	-4
H	-1	1	1	1	-3	0	0	-3	9	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0	-1	-4
I	-1	-3	-3	-3	-2	-3	-4	-4	-3	4	3	-4	1	0	-3	-2	-1	-3	-1	3	-3	-3	-1	-4
L	-1	-2	-4	-4	-1	-2	-3	-4	-3	3	6	-3	2	0	-3	-3	-1	-2	-1	3	-4	-3	-1	-4
K	-1	2	0	-1	-3	1	0	-2	-1	-4	-3	6	-2	-3	-1	0	-1	-3	-2	-2	0	1	-1	-4
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-2	8	1	-3	-1	-1	-3	0	1	-3	-1	-1	-4
F	-1	-2	-3	-3	-2	-3	-4	-4	-1	0	0	-3	1	6	-4	-2	-2	1	3	0	-3	-3	-1	-4
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-3	-4	8	-1	-1	-4	-3	-3	-2	-1	-2	-4
S	1	0	1	1	-1	0	-1	0	-1	-2	-3	0	-1	-2	-1	4	2	-3	-1	-2	0	0	0	-4
T	0	-1	0	-2	-1	-1	-4	-2	-2	-1	-1	-1	-1	-2	-1	2	7	-2	-2	0	-1	-1	0	-4
W	-3	-2	-4	-4	-1	-2	-3	-2	-2	-3	-2	-3	-3	1	-4	-3	-2	13	3	-3	-4	-3	-2	-4
Y	-3	-2	-2	-3	-2	0	-2	-3	2	-1	-1	-2	0	3	-3	-1	-2	3	8	-1	-3	-2	-1	-4
V	0	-2	-3	-3	-1	-2	-2	-3	-3	3	3	-2	1	0	-3	-2	0	-3	-1	4	-3	-2	-1	-4
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	1	-1	-4
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
X	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	0	0	-2	-1	-1	-1	-1	-1	-4
*****	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	1

(a) ProtSub

	L	I	V	G	A	P	Q	N	M	T	S	C	E	D	K	R	Y	F	W	H
L	100	95	91	76	85	77	78	80	80	77	77	76	70	75	77	67	70	81	70	75
I	95	100	93	75	83	76	76	77	78	79	75	74	69	72	76	66	66	78	66	72
V	91	93	100	81	88	80	75	77	77	83	80	79	69	72	74	64	64	74	63	69
G	76	75	81	100	91	83	68	74	66	80	81	79	63	67	69	57	64	71	61	66
A	85	83	88	91	100	84	74	75	75	80	84	87	66	70	74	63	66	74	64	71
P	77	76	80	83	84	100	67	69	69	75	76	75	57	60	69	57	63	73	63	64
Q	78	76	75	68	74	67	100	90	87	76	79	78	86	78	79	77	71	76	69	73
N	80	77	77	74	75	69	90	100	79	81	85	81	81	86	73	69	72	76	68	76
M	80	78	77	66	75	69	87	79	100	75	78	83	74	69	76	72	68	76	69	70
T	77	79	83	80	80	75	76	81	75	100	88	81	66	71	66	62	68	72	62	66
S	77	75	80	81	84	76	79	85	78	88	100	87	71	75	69	63	71	75	65	72
C	76	74	79	79	87	75	78	81	83	81	87	100	67	72	68	64	66	71	63	69
E	70	69	69	63	66	57	86	81	74	66	71	67	100	93	74	69	66	65	57	73
D	75	72	72	67	70	60	78	86	69	71	75	72	93	100	66	61	66	65	55	65
K	77	76	74	69	74	69	79	73	76	66	69	68	74	66	100	86	66	72	66	73
R	67	66	64	57	63	57	77	69	72	62	63	64	69	61	86	100	64	66	64	69
Y	70	66	64	64	66	63	71	72	68	68	71	66	66	66	66	64	100	87	79	77
F	81	78	74	71	74	73	76	76	76	72	75	71	65	65	72	66	87	100	87	82
W	70	66	63	61	64	63	69	68	69	62	65	63	57	55	66	64	79	87	100	75
H	75	72	69	66	71	64	73	76	70	66	72	69	73	65	73	69	77	82	75	100

(b) *Sneath's Index*

Figura 4.4: Matrizes de Substituição utilizadas para a representação de dados de aminoácidos.

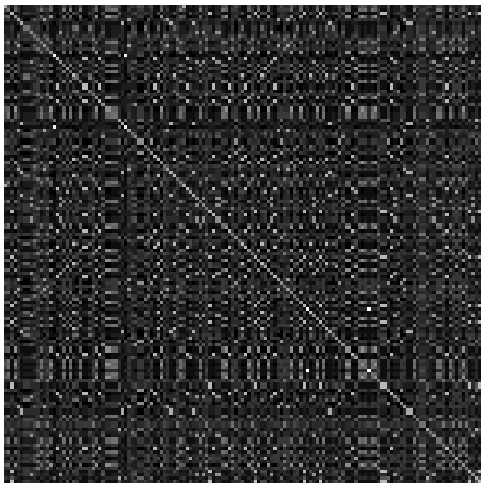
Sneath's Index (SNEATH, 1966) é uma técnica bastante antiga, mas ela continua sendo importante, inclusive, recentemente, quando foi utilizada para ajustar pesos de algoritmos de DL para predizer mutações (BERMAN et al., 2023). Os valores de *Sneath's Index* desse último trabalho foram utilizados como medida de similaridade de aminoácidos, que foram aplicados na camada B. A diferença é que, neste trabalho, os valores foram normalizados entre 0 e 255 para manter o padrão entre todas as camadas. O índice de *Sneath* descreve o grau de correlação entre aminoácidos, de acordo com 134 características relativas à estrutura e à atividade, e isso refletiu diretamente na criação da camada B. Ela explicita a similaridade molecular e de atividade dentro do conjunto de aminoácidos. Dentro dos aspectos estruturais, o índice de *Sneath* considera: peso molecular, pontes de hidrogênio, presença de grupo isopropil, anéis aromáticos, etc. Do ponto de vista de atividade *Sneath* considera como atividade do aminoácido no meio, sendo assim utiliza: polaridade, ponto isoelétrico em diferentes pHs, solubilidade na água, momento de inércia de grupamentos estruturais, etc. As matrizes de substituição usadas na modelagem das camadas estão representadas na Fig 4.4, antes de serem normalizadas entre 0 e 255 para seguir o padrão do resto da tese.

Na Fig 4.5, estão ilustradas cada camada desenvolvida na sequência de aminoácidos de Hemoglobina$_\beta$ humana e o mesmo padrão de nucleotídeos das imagens como um tartã, com várias listras e regiões de coloração mais densa, aparece novamente. Nela, estão representadas as frequências de substituição na camada R, as duplicações ou repetições na camada G e, na camada B, a similaridade molecular e de atividade dos aminoácidos. Vale reparar que a camada B ficou com bastante informação distinta por conta da granularidade da medida de *Sneath*.

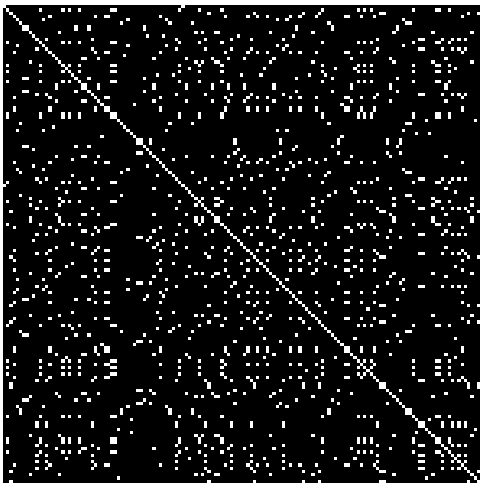
Na modelagem de aminoácidos, assim como na de nucleotídeos, também existem nas representações das regiões de baixa complexidade. A complexidade nessa modelagem não é somente em relação às letras comparadas da sequências (camada G), mas também, de acordo com áreas que potencialmente podem ter grande frequência de substituição (camada R), ou grande similaridade de estrutura e atividade molecular (camada B).

Possibilitar a visualização de todos esses fatores em conjunto habilita um ganho importante para pesquisadores, que despendem de trabalho para avaliar tudo separadamente. A proposta que está sendo apresentada nesta tese automatiza a extração de características e comparações entre as representações sem necessidade manual. Além disso, a criação das camadas R e B, nesta representação, mostram o poder das representações na colaboração entre pesquisas científicas. Utilizaram-se dois trabalhos, com menos de dois anos,

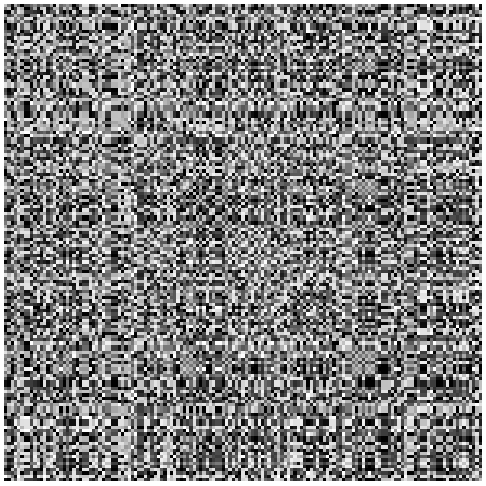
como base para representar características nas camadas.



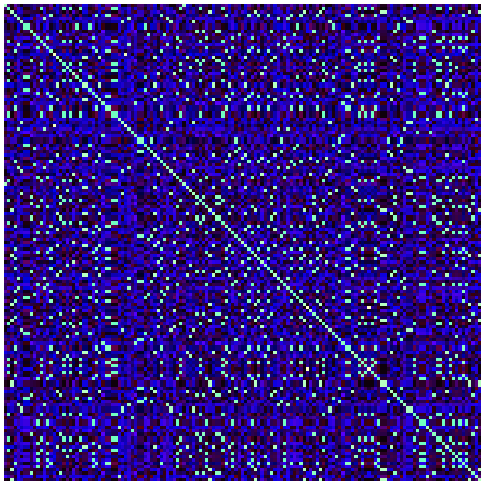
(a) Camada R, frequência de substituição.



(b) Camada G, duplicações ou repetições.



(c) Camada B, similaridade molecular e de atividade.



(d) Representação agregada com as três.

Figura 4.5: Resultado de cada camada desenvolvida para aminoácidos na sequência de PTN humana da Hemoglobina$_\beta$.

4.2 Comparação de Similaridade

As técnicas de extração de características e comparação utilizadas na tese se baseiam em técnicas de imagens, mas não se limitam a isso. O objetivo da criação e adaptação das técnicas descritas possuem um foco em dados de moléculas biológicas, mas são extensíveis a qualquer comparação de imagens ou estruturas com mesmo padrão comparativo de divisão na diagonal.

Essa seção está dividida apresentando primeiramente métodos de comparação de similaridade com algoritmos tradicionais de comparação de imagem

que utilizaram redimensionamento para fazer as comparações na Seção 4.2.1. Dentro dos algoritmos com redimensionamento, foram testados na tese algoritmos de *embedding* com técnicas de DL em 4.2.1.1 e, na Seção 4.2.1.2 algoritmos tradicionais baseados em similaridade estrutural de imagem. Além dos algoritmos de redimensionamento, são propostos novos algoritmos na Seção 4.2.2 baseados em similaridade estrutural, sendo eles: WMS-SSIM 4.2.2.1, GS-SSIM 4.2.2.2 e US-SSIM 4.2.2.3. Por último, na Seção 4.2.3 os algoritmos passam por otimização de parâmetros e conseguem resultados mais próximos ao Clustal.

4.2.1 Com Redimensionamento

O objetivo da validação de resultados de algoritmos mais tradicionais com redimensionamento são principalmente duas:

1. ter uma base comparativa de performance com os algoritmos propostos na Seção 4.2.2;

2. investigar se existe a possibilidade de uso das representações propostas na Seção 4.1 com algoritmos já existentes.

4.2.1.1 *Deep Search* e *Embedding*

A primeira abordagem descrita, utiliza a metodologia da Seção 3.2.3, muito tradicional em comparações de imagem usando DL. Nela, os dados são transformados em vetores de *embeddings* e comparados através de algoritmos de distância. O buscador de vetores similares Annoy cria um índice para todos os vetores e busca através dele usando a técnica de ANN.

Com a indexação no Annoy das representações mostrou ser possível acoplar a extração de características nas técnicas de ML e DL. Grande parte dessas técnicas transformam as estruturas como texto ou imagem nesses vetores de *embedding* para uso em algoritmos de classificação, agrupamento, etc. Essa indexação gerou bases de dados separadas para cada conjunto de dados utilizados na tese, assim como dois índices maiores: um com todas as representações para nucleotídeos e, outra para aminoácidos. Com esses índices, os casos de uso aplicados são executados instantaneamente, requisitando quase nenhum poder computacional.

Para o uso dessa técnica de ANN, é necessário ter todas as imagens com mesmo tamanho. Para isso, foi utilizada a padronização da imagem através da técnica de redimensionamento para 1000x1000 nos casos de uso

(tarefas) de bioinformática, dado que o o Annoy funciona melhor em baixa dimensionalidade (WOLFF, 2016). Como o tamanho de 1000 é superior ao tamanho de algumas sequências utilizadas, o redimensionamento foi feito adicionando uma borda preta ao redor da imagem para não deformar as imagens menores comparadas e preservar as características delas antes de gerar o *embedding*, conforme Fig 4.6.

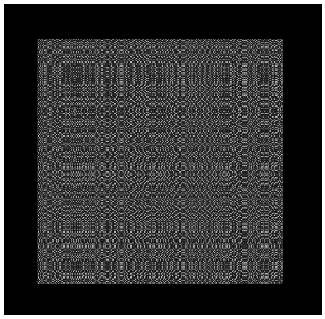


Figura 4.6: Imagem redimensionada utilizando borda preta para preservar características.

Existem outras ferramentas gratuitas e de código aberto que fazem indexação para comparação de ANN, em mais larga escala, conseguindo lidar com a limitação de dimensionalidade como o Vespa [1]. Para validação o Annoy foi um primeiro caso satisfatório de uso, pela facilidade de desenvolvimento. Na implementação, foi utilizada a distância euclideana para comparar os vetores e o número de t de árvores foi igual ao número de amostras para validação descartando resultados aproximados. Tudo foi planejado para que a técnica esteja nas condições ideais, buscando pelas distâncias par a par.

Sobre a técnica de redimensionamento utilizada, existem outras abordagens como métodos de MSA para entender onde colocar as quebras em preto ou tentar produzir algum outro tipo de heurística de redimensionamento. Apesar disso, neste manuscrito, foi utilizado somente o redimensionamento com a borda preta, deixando em aberto outras alternativas. A qualidade do vetor de características produzido precisa ser calibrada para que as representações estejam em um mesmo tamanho indexável, mas, o melhor resultado foi obtido sem esticar as representações de dados no momento.

O que será demonstrado nos resultados do algoritmo *Deep Search*, na Seção 4.3.1, será resultante da combinação da qualidade do vetor de *embedding* com o algoritmo de DL aplicado para extrair as características. Além disso, no resultado, também há o impacto da escolha do algoritmo e a distância euclideana utilizada para comparação das sequências. Com isso, deve-

[1] <https://vespa.ai/>

se entender que essa técnica é muito mais complexa e não depende somente da representação de dados proposta.

O *Deep Search* consegue funcionar, inclusive, usando separadamente cada camada, como ilustrado previamente nas Figs 4.2 e 4.5. Ele extrai a distância de cada uma das camadas da representação, assim como para a camada agregada conhecida como *full* nos resultados a seguir, na Seção 4.3.1. Isso contribui para a explicabilidade dos modelos de ML durante a validação de experimentos, permitindo identificar qual das camadas possui maior influência nos resultados positivos ou negativos. Além disso, também possibilita a junção do resultado das camadas, aplicando pesos distintos, caso necessário, ao invés de manter equalizado, como é no *full*.

Ao extrair as características, não são necessárias letras estampadas e os algoritmos nem dependem dessa informação. Isso permite a utilização delas sem a necessidade de conhecer a sequência originária e ainda tendo a informação associada aos padrões internos dela. O dado passou por um processo de *data masking* automaticamente, que contribui para a segurança e a privacidade dos dados que são analisados, dificultando a chance de rastreabilidade e identificação da fonte originária dos dados.

Outro resultado importante foi conseguir aplicar a técnica de ANN nos *embeddings* criados, porque isso possibilita buscas e comparações extremamente mais rápidas, utilizando as novas representações de dados. A dissimilaridade utilizada foi a distância euclideana, mas é possível implementar ou utilizar novas medidas de dissimilaridade. Uma vez que essas medidas representem bem as comparações com as representações propostas, permite trocar a distância euclideana. Em paralelo, o que será mostrado é que a representação proposta pode servir como entrada direta para algoritmos de similaridade solucionarem problemas conhecidos da bioinformática sem a necessidade da criação de *embeddings*.

4.2.1.2 *Resized* UQI, SSIM e MS-SSIM

A forma mais simples de fazer comparações utilizando os algoritmos de comparação estrutural é através da padronização das imagens para o mesmo tamanho. Para evitar que sejam perdidas informações com a diminuição de imagens, como acontece no FCGR, optou-se por aumentar o tamanho das imagens geradas, quando forem comparadas imagens de tamanhos diferentes. Essa abordagem pode ser perigosa por inventar informação que não existe nas moléculas de menor tamanho, mas já é esperado que esses algoritmos não funcionem bem para comparações de tamanhos muito distintos.

Existem diferenças entre os algoritmos UQI, SSIM e MS-SSIM. Nos resultados obtidos o UQI possui problemas de implementação que o SSIM veio corrigir, mas ambos possuem uma implementação bem parecida. O problema do UQI será mais bem descrito na Seção 4.3.1. O MS-SSIM utiliza vários focos de comparação das imagens e consegue captar alguns padrões que os anteriores não conseguem, mas ao mesmo tempo, é mais lento que seus precursores.

As técnicas de redimensionamento foram utilizadas para os algoritmos UQI, SSIM e MS-SSIM, sendo chamados respectivamente de UQI, R-SSIM e RMS-SSIM nos experimentos. A complexidade assintótica desses algoritmos, no pior caso, é equivalente à complexidade de redimensionamento mais a complexidade dos algoritmos utilizados na base, e no melhor caso, é igual à complexidade do algoritmo da base porque nenhum redimensionamento é requerido.

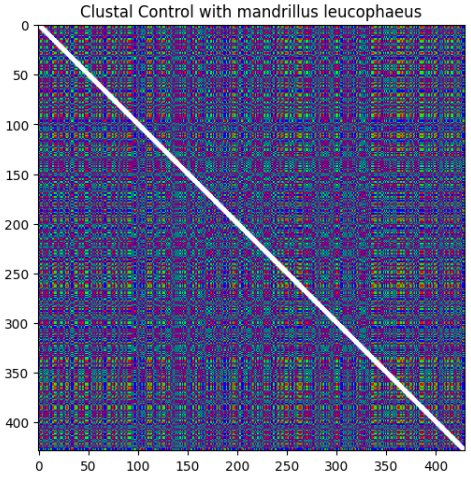
4.2.2 Novos Algoritmos Propostos

Como essa representação de dados não é comparável diretamente devido a moléculas de diferentes tamanhos, a outra alternativa, além da criação de vetores de *embedding* e redimensionamento de representações, é criar novos algoritmos que consigam medir as dissimilaridades entre essas moléculas. Para propor os algoritmos, foi utilizado como base que as representações de dados explicitam padrões internos das sequências, bem como CGR, e que algoritmos de comparação estrutural conseguiram bons resultados com CGR (KARAMICHALIS et al., 2015), conforme comentado previamente.

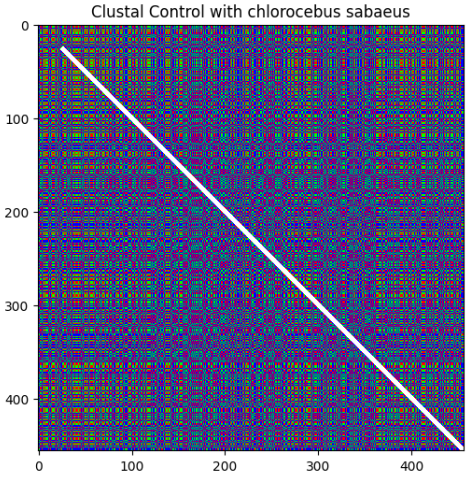
Em cada um dos algoritmos criados, são apresentadas como foi a sua execução nas sequências de nucleotídeos de *Chlorocebus Sabaeus* e *Mandrillus Leucophaeus* do conjunto de dados de *Neuroglobin*. Nessas duas sequências, a segunda é quase uma subsequência da primeira e isso facilita a comparação visual dos resultados. Na Fig 4.7, são mostrados o alinhamento feito com Clustal e, a seguir, a mesma representação do alinhamento, mas nas imagens geradas da representação de dados. A linha branca representa a porção das sequências que foram usadas no alinhamento. Os asteriscos, embaixo das letras no alinhamento, representam uma correspondência exata dos nucleotídeos. O alinhamento começa a ocorrer em torno da posição 40, mas os asteriscos ficam realmente consecutivos em torno da 90, com muitas correspondências erradas antes. Essas correspondências erradas são bastante comuns, porque criar um *gap* em um alinhamento é, muitas vezes, mais custoso para o alinhamento do que alinhar com correspondências diferentes.

```
CLUSTAL O(1.2.4) multiple sequence alignment

chlorocebus_sabaeus_ENSCSAG00000011845        ATGGAGCGCCCAGAGCCCGAGCTGATCCGGCAGAGCTGGCGGGCGGTGAGCCGCAGCCCG     60
mandrillus_leucophaeus_ENSMLEG00000035424     -------------------------ATGGCTGTG-GTCTTGGCCCTGTCCCTACGGCTG     33
                                                                    *** * *      ***  **  **   * * *

chlorocebus_sabaeus_ENSCSAG00000011845        CTGGAGCACGGCACCGTCCTGTTCGCCAGGCTGTTTGCCCTGGAGCCTGACCTGCTGCCC     120
mandrillus_leucophaeus_ENSMLEG00000035424     AAAGCCCTGGTGGTCCCCAACGTGTGCTGGCTGTTTGCCCTGGAGCCTGACCTGCTGCCC     93
                                                 *  *  *    *  *    *   * ********************************

chlorocebus_sabaeus_ENSCSAG00000011845        CTCTTCCAGTACAACTGCCGCCAGTTCTCCAGCCCAGAGGACTGTCTCTCCTCACCTGAG     180
mandrillus_leucophaeus_ENSMLEG00000035424     CTCTTTCAGTACAACTGCCGCCAGTTCTCCAGCCCAGAGGACTGTCTCTCCTCACCTGAG     153
                                              ***** ************************************************************

chlorocebus_sabaeus_ENSCSAG00000011845        TTCCTGGACCACATCAGGAAGGTGATGCTCGTGATTGATGCTGCGGTGACCAATGTGGAA     240
mandrillus_leucophaeus_ENSMLEG00000035424     TTCCTGGACCACATCAGGAAGGTGATGCTCGTGATTGATGCTGCGGTGACCAATGTGGAA     213
                                              ************************************************************

chlorocebus_sabaeus_ENSCSAG00000011845        GACCTGTCCTCACTGGAGGAGTACCTTGCCAGCCTGGGCAGGAAGCACCGGGCAGTGGGT     300
mandrillus_leucophaeus_ENSMLEG00000035424     GACCTGTCCTCACTGGAGGAGTACCTTGCCAGCCTGGGCAGGAAGCACCGGGCAGTGGGT     273
                                              ************************************************************

chlorocebus_sabaeus_ENSCSAG00000011845        GTGAAGCTCAGCTCCTTCTCGACAGTGGGTGAATCTCTGCTCTACATGCTGGAGAAGTGT     360
mandrillus_leucophaeus_ENSMLEG00000035424     GTGAAGCTCAGCTCCTTCTCGACAGTGGGTGAATCTCTGCTCTACATGCTGGAGAAGTGT     333
                                              ************************************************************

chlorocebus_sabaeus_ENSCSAG00000011845        CTGGGCCCTGCCTTCACACCAGCCACACGGGCTGCCTGGAGCCAGCTCTACGGGGCTGTG     420
mandrillus_leucophaeus_ENSMLEG00000035424     CTGGGCCCTGCCTTCACACCAGCCACACGGGCCGCCTGGAGCCAGCTCTACGGGGCTGTG     393
                                              ********************************** ************************

chlorocebus_sabaeus_ENSCSAG00000011845        GTGCAGGCCATGAGTCGAGGCTGGGACAGCGAGTAA     456
mandrillus_leucophaeus_ENSMLEG00000035424     GTGCAGGCCATGAGTCGAGGCTGGGACAGCGAGTAA     429
                                              ***********************************
```

(a) Alinhamento tradicional com Clustal.



(b) Sequência menor, *Mandrillus Leucophaeus*.



(c) Sequência maior, *Chlorocebus Sabaeus*.

Figura 4.7: Representação de alinhamento como imagens dos nucleotídeos de *Chlorocebus Sabaeus* e *Mandrillus Leucophaeus* de *Neuroglobin*.

Nos algoritmos desenvolvidos neste trabalho, todos possuem esse monitoramento de correspondências para serem comparados aos alinhamentos. O monitoramento de correspondências indica o passo a passo que o algoritmo percorreu e quais regiões da representação de dados está sendo utilizada durante a comparação. Com isso, conseguimos saber as regiões de maior similaridade de uma representação em relação a outra que está sendo comparada. O mesmo não é possível de ser medido nos algoritmos que usam redimensionamento, porque não se sabe a porção da sequência que está sendo usada, uma vez que toda ela é usada ao mesmo tempo. Além do comparativo com alinhamentos, todos

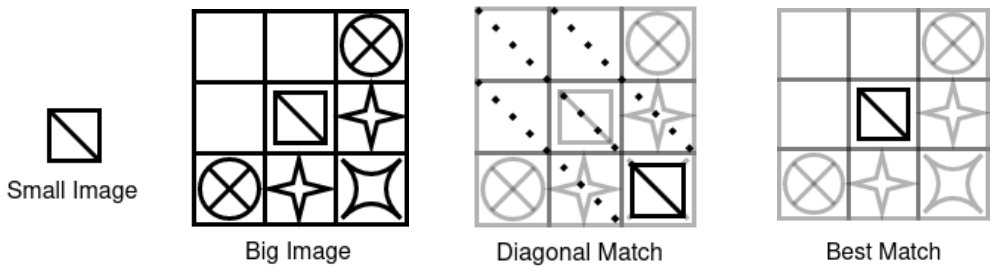
os algoritmos podem ter um controle fino sobre a utilização de cada camada, assim como dar pesos diferentes para cada uma das camadas, mas optou-se por deixar todas com o mesmo peso.

Este controle fino sobre o que está sendo usado e comparado aos alinhamentos é uma vantagem sobre outros métodos porque permite que o usuário entenda o que está sendo avaliado. Isso contribui como possibilidade de controle da explicabilidade dos resultados por parte dos algoritmos e não só das camadas.

4.2.2.1 *Windowed* MS-SSIM

Considerando sequências de tamanhos diferentes, o novo método proposto utilizando como base o algoritmo MS-SSIM é usado para comparar uma representação menor com a representação maior. Como o algoritmo MS-SSIM não suporta comparações de imagens de tamanhos distintos, foi preciso elaborar uma metodologia para suportar as comparações que não fosse voltada para o redimensionamento das imagens.

O método *Windowed* MS-SSIM (WMS-SSIM) consiste na comparação da representação menor em diagonal com a representação maior. A comparação é feita da esquerda para a direita e de cima para baixo, seguindo a diagonal principal, conforme ilustrado na Fig 4.8. O algoritmo caminha na diagonal por uma janela igual ao tamanho da representação menor, utilizando passos de 1 em 1. Ele valida sempre todas as possibilidades de comparação e busca a melhor correspondência da representação menor na maior. Após percorrer toda a representação maior ou até encontrar o valor máximo de correspondência possível (totalmente idênticas), o algoritmo termina e retorna ao valor máximo de correspondência encontrado. No Alg 2 foi feito um pseudocódigo ilustrando o algoritmo desenvolvido.



(a) Representação menor e representação maior.

(b) Percorrendo a representação maior pela diagonal.

(c) Melhor correspondência encontrada.

Figura 4.8: Exemplo de atuação do algoritmo WMS-SSIM.

Algoritmo 2: Pseudocódigo do algoritmo WMS-SSIM de comparação de similaridades.

Entrada: $rep_{small}, rep_{big}, P$
Saída: Ponto flutuante entre 0 e 1

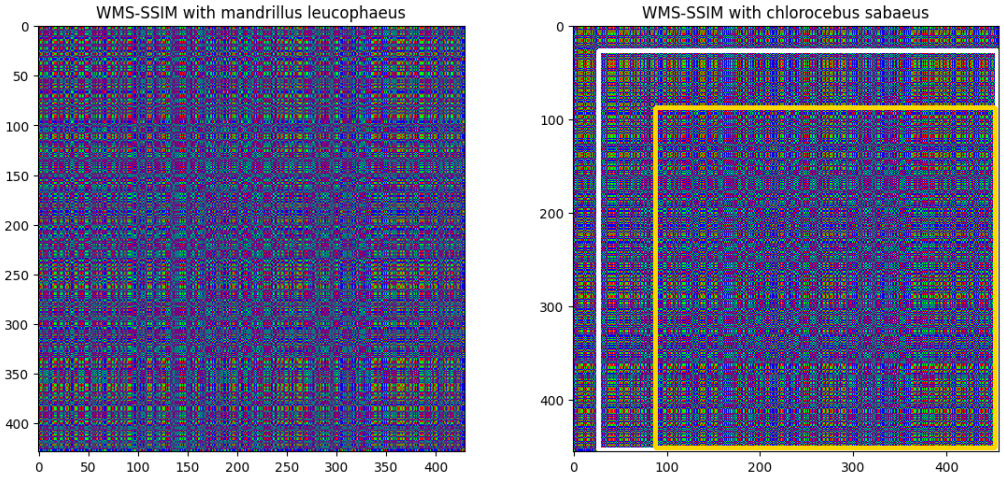
$S_{small} \leftarrow Size(rep_{small})$
$S_{big} \leftarrow Size(rep_{big})$
$score_{max} \leftarrow 0$
para $i = 0$, i+=P, $i \leq (S_{big} - S_{small})$ **faça**
    Pega uma fatia do mesmo tamanho da pequena na grande
    $slice_{big} \leftarrow rep_{big}[i : (S_{small} + i), i : (S_{small} + i)]$
    $score \leftarrow MSSSIM(rep_{small}, slice_{big})$
    $score_{max} \leftarrow Max(score, score_{max})$
retorna $score_{max}$

Nesta tese, o foco foi no desenvolvimento e validação do algoritmo, mas existem melhorias que podem ser feitas, como o caso de uma escolha mais performática para o passo do algoritmo, evitando que todas as comparações sejam feitas. Sendo S_{big} o tamanho da representação maior, S_{small} o tamanho da menor, B a complexidade do algoritmo MS-SSIM e P o passo do algoritmo, que hoje é 1, a complexidade assintótica dele, no pior caso é $\frac{(S_{big} - S_{small}) \cdot B}{P}$, dado que a distância percorrida é $(S_{big} - S_{small})$, por estar indo na diagonal, comparando só o tamanho que falta para o menor chegar ao maior.

Esse algoritmo será mais efetivo encontrando subsequências dentro de outra sequência, podendo detectar também processos evolutivos, tais quais mutações, inserções ou deleções tenham ocorrido nelas. Em contrapartida aos algoritmos anteriores, que comparavam as imagens no âmbito global, esse método foca em comparações locais, mas ainda possui um pouco do contexto global a sua volta.

A Fig 4.9 mostra uma aplicação do algoritmo similar ao que foi mostrado na Fig 4.7 com o Clustal. A representação menor não possui nenhum tipo de marcação, porque sua totalidade é utilizada para comparação. Por outro lado, a representação maior possui o quadrado branco, representando o melhor posicionamento da representação menor dentro da maior encontrado pelo algoritmo. O quadrado amarelo reflete o que o alinhamento mostra como o lugar em que existem várias correspondências seguidas, sem quase nenhum *gap*, a partir da posição 88 mais especificamente. O quadrado amarelo é o que o algoritmo deveria encontrar. Como o WMS-SSIM utiliza a imagem menor inteira na busca, que é maior do que o quadrado amarelo, ele encontrou como

melhor correspondência de acordo com o esperado do algoritmo.



(a) Sequência menor, *Mandrillus Leucophaeus*.

(b) Sequência maior, *Chlorocebus Sabaeus*.

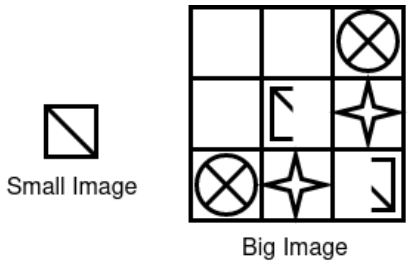
Figura 4.9: Representação do algoritmo WMS-SSIM nas imagens dos nucleotídeos de *Chlorocebus Sabaeus* e *Mandrillus Leucophaeus* de *Neuroglobin*.

4.2.2.2 *Greedy Sliced* SSIM

O algoritmo WMS-SSIM foca em um comportamento local e os de redimensionamento focam no global. A aposta na criação deste novo algoritmo é em alternativas glocais, que vão combinar tanto as vantagens de similaridades locais como globais (BRUDNO et al., 2003). Na primeira proposta glocal, foi utilizada a lógica de fatiar uma representação de molécula em colunas conforme a Fig 4.10b. A lógica é que em cada coluna, existem todas as correspondências de um nucleotídeo ou aminoácido específico com todos os outros. Isso faz com que todo nucleotídeo ou aminoácido tenha seus contextos de correspondência de forma global na sequência.

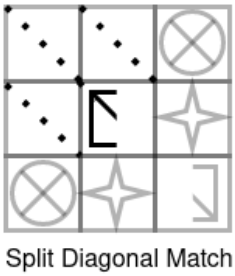
Uma vez que o aspecto global é conseguido dentro da coluna de uma representação menor, é possível buscar dentro da representação maior a melhor correspondência local para aquela coluna. A Fig 4.10 mostra o funcionamento do algoritmo *Greedy Sliced* SSIM (GS-SSIM) em que uma representação menor possui suas colunas fatiadas no valor $filter\_size$ definido no algoritmo. Cada uma dessas colunas vai buscando sua melhor correspondência após a anterior ter encontrado a dela. Com essa abordagem cada coluna pode se manter colada na anterior ou pode encontrar uma correspondência melhor a seguir. O tamanho das colunas criadas pode ser chamada de "janela de comparação" e é definida pelo tamanho do $filter\_size$ do algoritmo de

comparação estrutural utilizado internamente. O algoritmo segue fazendo o deslocamento em $\lfloor \frac{filter\_size}{2} \rfloor$ na diagonal. Com isso, a nova comparação sempre mantém metade da janela anterior nela, incluindo tanto o contexto colunar como das posições anteriores na medida de similaridade.

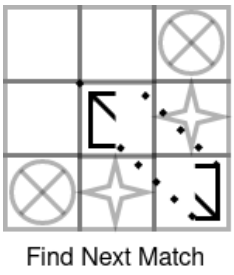


(a) Representação menor e representação maior.

(b) Representação menor é quebrada em colunas.



(c) Encontrando melhor correspondência para primeira coluna na diagonal.



(d) A partir da última, segue buscando para outras colunas.

Figura 4.10: Exemplo de atuação do algoritmo GS-SSIM.

Por conta de limitações com o algoritmo MS-SSIM de tamanho de representação, foi utilizado o algoritmo SSIM nessa e na próxima abordagem glocal. Como contribuição para a comunidade científica do Tensorflow, este trabalho deixou uma colaboração [2] que controle e informe ao utilizador do algoritmo MS-SSIM que caso ele não possa ser utilizado nas condições que o usuário deseja.

Apesar de conseguir abordar tanto o aspecto local quanto global, o GS-SSIM possui a limitação de não levar em consideração todas as opções possíveis de comparação e otimizar tal qual a um algoritmo guloso. Ele funciona bem como uma alternativa para encontrar subsequências que contenham *gaps* dentro de outra sequência maior. Para considerar que a sequência menor pode ter *gaps* em relação a maior, é possível definir um limiar de similaridade para que uma coluna da representação menor seja considerada para correspondência

[2] <https://github.com/tensorflow/tensorflow/pull/57732>

Algoritmo 3: Pseudocódigo do algoritmo GS-SSIM de comparação de similaridades.

Entrada: $rep_{small}, rep_{big}, P, filter\_size, limiar$
Saída: Ponto flutuante entre -1 e 1

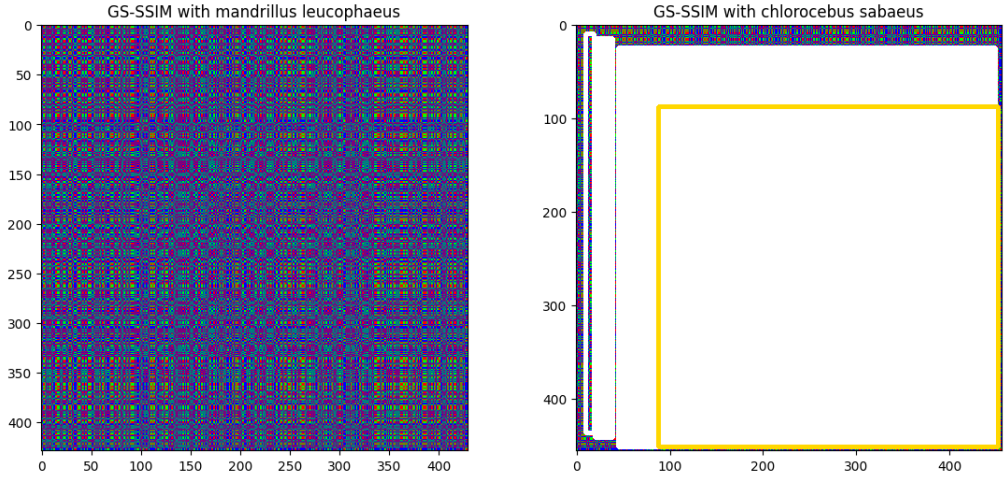
$S_{small} \leftarrow Size(rep_{small})$
$S_{big} \leftarrow Size(rep_{big})$
$scores \leftarrow 0$
$denominador \leftarrow 0$
$last\_filled_{line} \leftarrow 0$
$last\_filled_{line}$ guarda a última melhor posição para continuar dela
para $i = 0$, i+=P, $i \leq (S_{small} - filter\_size)$ **faça**
    $score_{max} \leftarrow 0$
    $last_{line} \leftarrow last\_filled_{line}$
    $slice_{small} \leftarrow rep_{small}[:, i : (filter\_size + i)]$
    $filter\_size$ determina a quantidade de colunas da menor
    para $j = 0$, j+=P, $(j + last_{line}) \leq (S_{big} - S_{small})$ **faça**
        $slice_{big} \leftarrow rep_{big}[(last_{line}+j) : (last_{line}+S_{small}+j), last_{line}+i+j : (last_{line} + filter\_size + i + j)]$
        $score \leftarrow SSIM(rep_{small}, slice_{big})$
        se $score > score_{max}$ **então**
            $last_{line} \leftarrow (last_{line} + j)$
            $score_{max} \leftarrow score$
    se $score > limiar$ **então**
        $last\_filled_{line} \leftarrow last_{line}$
        $scores \leftarrow scores + score$
        $denominador \leftarrow denominador + 1$
retorna $scores/denominador$

na maior. Neste manuscrito, foi usado o limiar como zero, considerando todas as colunas que tenham algum tipo de similaridade com a maior. Como o algoritmo SSIM vai de -1 até 1, sendo -1 completamente oposto e 1 completamente idêntico o zero pareceu um primeiro bom corte.

Esse algoritmo possui um comportamento um pouco mais sequencial, dado que depende da interação anterior de uma coluna para seguir para a próxima, sendo mais difícil de paralelizar. Por outro lado, esse aspecto mais sequencial faz com que esse algoritmo seja mais conservador, evitando comparações equivocadas com inversões e translocações da sequência analisada. Dadas as colunas $c1, c2, c3 \ldots cn$, não tem como a coluna $c2$ encontrar uma

correspondência menor do que a coluna $c1$. Um pseudocódigo de exemplo foi disponibilizado em Alg 3 mostrando o funcionamento dele.

Sendo S_{big} o tamanho da representação maior, S_{small} o tamanho da menor, B a complexidade do algoritmo SSIM e P o passo do algoritmo, a complexidade assintótica desse algoritmo é superior ao WMS-SSIM no pior caso $\frac{(S_{big}-S_{small})\cdot(S_{small}-filter\_size)\cdot B}{P}$. Mas o tempo de execução do SSIM é em torno de cinco vezes menor que o MS-SSIM. Isso acontece porque o MS-SSIM na configuração padrão repete a mesma operação do SSIM para cinco diferentes escalas na imagem, dando zoom para realizar comparações. A complexidade acaba sendo menor para comparação de moléculas parecidas porque sempre que $last\_filled_{line}$ é preenchida, a quantidade de comparações é amortizada. O algoritmo tendeu a ser sempre o mais rápido nos experimentos por amortizar algumas opções de correspondência para as próximas colunas.



(a) Sequência menor, *Mandrillus Leucophaeus*.

(b) Sequência maior, *Chlorocebus Sabaeus*.

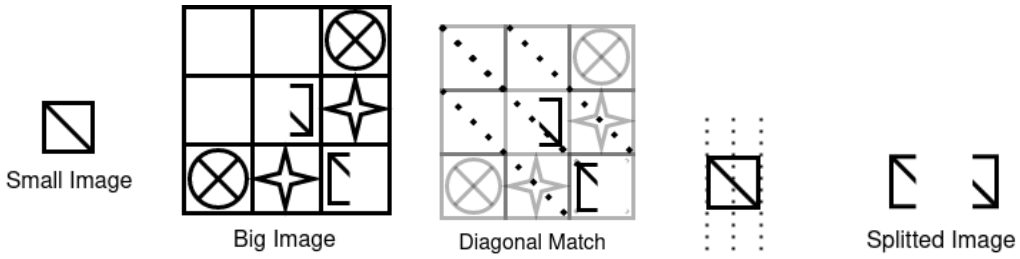
Figura 4.11: Representação do algoritmo GS-SSIM nas imagens dos nucleotídeos de *Chlorocebus Sabaeus* e *Mandrillus Leucophaeus* de *Neuroglobin*.

Na Fig 4.11 a representação menor não possui nenhum tipo de marcação, já que sua totalidade foi utilizada para comparação, porque as imagens são muito parecidas e não foram afetadas pelo limiar de corte. Igual ao algoritmo anterior, todas as marcações estão na representação maior, mostrando o caminho na diagonal da representação menor pela maior. Assim como o Clustal, o algoritmo conseguiu incluir alguns *gaps* e se concentrou próximo ao fim da representação maior, que é realmente onde o quadrado amarelo mostra maior quantidade de posições correspondentes.

4.2.2.3 *Unrestricted Sliced* SSIM

Como o último algoritmo possui uma abordagem glocal bastante ampla, mas tenta seguir passos similares ao de alinhamentos, foi proposta uma outra abordagem que se aproprie melhor do contexto global. Esse novo algoritmo visa algo que consiga lidar melhor com trocas na posição de nucleotídeos e aminoácidos. Ao mesmo tempo que possui melhor controle das comparações das posições, esse algoritmo possui mais riscos de encontrar correspondências ao acaso.

No algoritmo *Unrestricted Sliced* SSIM (US-SSIM), são consideradas as colunas da representação menor como um espaço que deve ser preenchido ou não por uma coluna correspondente da representação maior. Como mostra na Fig 4.12, primeiro a representação menor é comparada com a representação maior, tal qual ao WMS-SSIM. Dessa vez, o valor das comparações pixel a pixel são retornados ao invés da similaridade agregada completa do algoritmo. Isso se chama *Index Map* no algoritmo SSIM e optou-se por agregar todos esses valores de similaridade por coluna através da média.



(a) Representação menor e representação maior.

(b) Representação menor percorrendo a representação maior pela diagonal e ficando só com maiores correspondências.

(c) Representação menor é quebrada em colunas e só as colunas com melhores correspondências são consideradas.

Figura 4.12: Exemplo de atuação do algoritmo US-SSIM.

Se o valor da média naquela coluna for superior ao que tinha antes e superior a um limiar definido de similaridade, ele passa a ser o valor final de similaridade da coluna. Somente os valores finais de similaridade das colunas participam do cálculo final de similaridade. Caso não entre nenhum valor de similaridade naquela coluna, ela é dispensada do resultado final. Quando a comparação chega ao fim da representação, a média dos valores

de similaridade guardados é feita, considerando, assim, só os valores mais relevantes. Um pseudocódigo foi implementado em Alg 4 com as informações de implementação comentadas.

Os valores de limiar e de tamanho de passos não foram explorados nesse algoritmo. O limiar é zero igual ao último algoritmo e o tamanho do passo foi unitário igual no WMS-SSIM. As principais diferenças dele para o WMS-SSIM são o algoritmo usado mais leve (SSIM), tal qual o GS-SSIM, e a heurística de agregação de similaridades. A responsabilidade por essa agregação foi removida do algoritmo interno para um produzido nesta tese. Mesmo assim, a complexidade assintótica do pior caso se manteve a mesma que o WMS-SSIM de $\frac{(S_{big}-S_{small})\cdot B}{P}$.

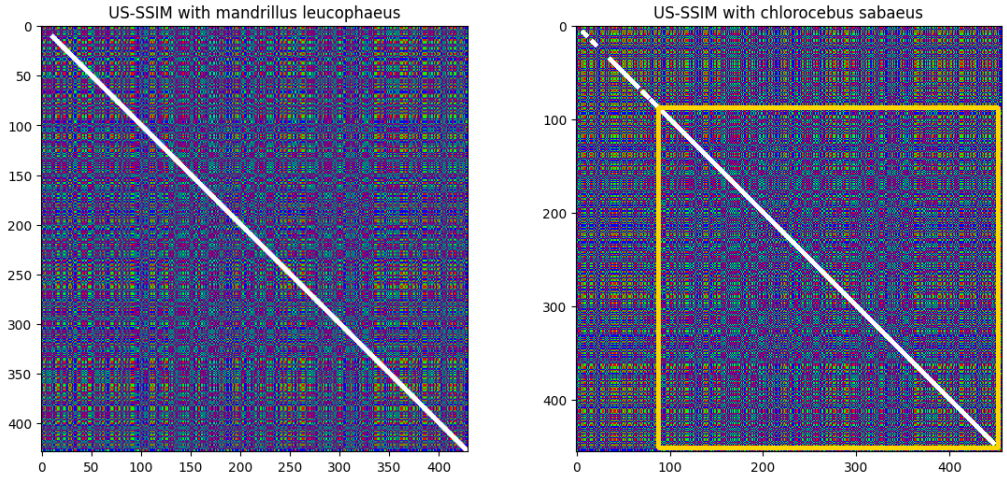
Algoritmo 4: Pseudocódigo do algoritmo US-SSIM de comparação de similaridades.

Entrada: $rep_{small}, rep_{big}, P, limiar$
Saída: Ponto flutuante entre -1 e 1

$S_{small} \leftarrow Size(rep_{small})$
$S_{big} \leftarrow Size(rep_{big})$
$scores \leftarrow 0$
$denominador \leftarrow 0$
$scores_{maxcols} \leftarrow [0] * S_{small}$
para $i = 0$, i+=P, $i \leq (S_{big} - S_{small})$ **faça**
    Pega uma fatia do mesmo tamanho da pequena na grande
    $slice_{big} \leftarrow rep_{big}[i : (S_{small} + i), i : (S_{small} + i)]$
    Utiliza SSIM pegando valor por pixel
    $scores_{pixel} \leftarrow SSIM(rep_{small}, slice_{big})$
    Agrega média do valor dos pixels por coluna
    $scores_{cols} \leftarrow MeanByColumn(scores_{pixel})$
    $scores_{maxcols} \leftarrow Max(scores_{maxcols}, scores_{cols})$
Usa somente a cima do limiar
para $j = 0$, j++, $j \leq S_{small}$ **faça**
    se $scores_{maxcols}[j] > limiar$ **então**
        $scores \leftarrow scores + scores_{maxcols}[j]$
        $denominador \leftarrow denominador + 1$
retorna $scores/denominador$

Os principais benefícios desse algoritmo, comparado ao anterior, é que ele pode ser facilmente paralelizável em GPUs por não depender de passos anteriores. Além disso, ele utiliza todas as correspondências possíveis para

depois fazer uma média do valor de SSIM de cada uma delas. Analisando todas as correspondências, torna este algoritmo aberto a encontrar inversões e translocações genômicas que não podem ser contempladas nos outros métodos. Ao mesmo tempo que habilita as inversões e translocações, também aumenta-se o risco de correspondências equivocadas como comentado anteriormente.



(a) Sequência menor, *Mandrillus Leucophaeus*.

(b) Sequência maior, *Chlorocebus Sabaeus*.

Figura 4.13: Representação do algoritmo US-SSIM nas representações dos nucleotídeos de *Chlorocebus Sabaeus* e *Mandrillus Leucophaeus* de *Neuroglobin*.

Na Fig 4.13, o mesmo exemplo comparativo relativo a *Neuroglobin*. A marcação do que foi usado na sequência maior com a linha branca, sinaliza o uso de praticamente toda a representação menor, exceto as que antecederam às primeiras linhas, que realmente possuem muitos *gaps*. Diferente do Clustal na Fig 4.7, não foram consideradas no US-SSIM mesmo com um limiar de zero, indicando que realmente são muito distintas. Um bom resultado é que, assim como o Clustal, o algoritmo US-SSIM desconsiderou várias colunas do início da representação grande. A quantidade de colunas desconsideradas poderia até ser maior, caso outros valores mais rígidos de limiar fossem testados.

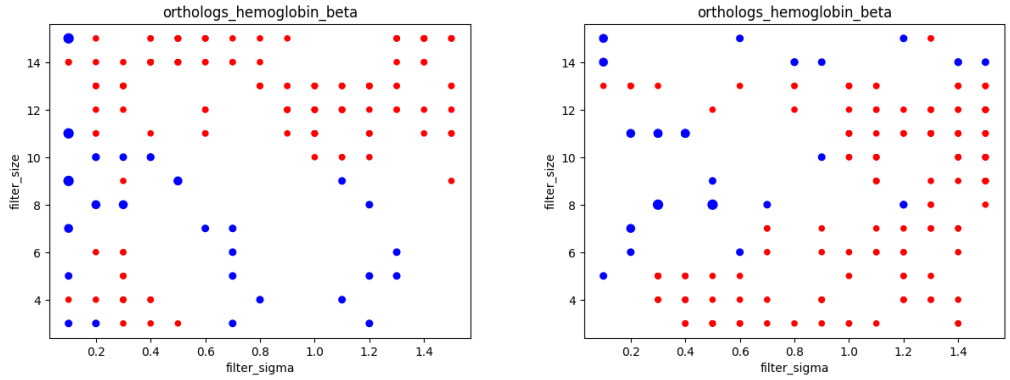
4.2.3 Otimização de Parâmetros

Para investigar a possibilidade de otimização dos algoritmos, tanto de redimensionamento, quanto os propostos, foi aplicada a otimização de parâmetros neles e alguns resultados serão mostrados aqui. A otimização foi contra o resultado do Clustal em cada conjunto de dados para validar se resultados similares ao Clustal seriam obtidos mudando os parâmetros.

Não são só os parâmetros específicos de cada algoritmo criado que podem ser otimizados, mas também os próprios parâmetros dos algorítimos utilizados como base. Nesta seção, são apresentados os resultados obtidos comparativamente para os algoritmos RMS-SSIM, WMS-SSIM, GS-SSIM e US-SSIM quando é aplicada a otimização bayesiana nos hiperparâmetros $filter\_size$ e $filter\_sigma$ visando aproximar os resultados obtidos do Clustal somente para as representações de nucleotídeos. Neste trabalho, a técnica de otimização bayesiana foi feita somente com nucleotídeos, mas é extensível a um trabalho futuro com aminoácidos.

Através dessa técnica, foram obtidos valores iguais ao Clustal, demonstrando que existe a possibilidade de melhorar os resultados da Seção 4.3.1 através de otimização de parâmetros dos algoritmos. Na Fig 4.14, as bolas vermelhas ocorrem quando uma combinação de $filter\_size$ e $filter\_sigma$ alcança o mesmo resultado do Clustal. As azuis possuem tamanhos distintos que, quanto maior, pior é o resultado comparativo com RF do Clustal.

Existe uma tendência de otimização, indo na diagonal para cima, da esquerda para a direita. Isso faz muito sentido, porque quanto maior a janela olhada ($filter\_size$) e quão mais "embaçar" as imagens com valores mais altos de $filter\_sigma$, mais o algoritmo ignora as pequenas diferenças, tal qual acontece com a visão humana quando existem objetos distantes e embaçados. O algoritmo foi validado com todos os conjuntos de dados e a tendência permanece ocorrendo.



(a) Obtidos para RMS-SSIM e WMS-SSIM.

(b) Obtidos para GS-SSIM e US-SSIM.

Figura 4.14: Valores de RF variando $filter\_size$ e $filter\_sigma$ no conjunto de nucleotídeos de Hemoglobina com algoritmos comparados ao Clustal, bolas vermelhas são os melhores.

Além disso, pela Fig 4.14 houve uma preferência da busca bayesiana por valores mais altos de $filter\_size$ nos algoritmos originários de MS-

SSIM, enquanto a preferência oposta de valores mais baixos de $filter\_size$ para os que possuem sua base usando o SSIM. O interessante disso é que, nos algoritmos GS-SSIM e US-SSIM, existe um valor ótimo de RF muito perto do três, que por acaso é o número de nucleotídeos necessários para codificar um aminoácido. Alguns múltiplos de três também tiveram melhor performance, desde que o $filter\_sigma$ seja maior e remova alguns ruídos da representação. Ainda sobre múltiplos de três, dentro dos parâmetros do MS-SSIM, ele utiliza cinco escalas de zoom, e os valores mais otimizados ficaram entre um $filter\_size$ de doze a quinze, que, por acaso, são quatro ou cinco vezes maiores do que o número necessário de nucleotídeos para codificar um aminoácido. Existem outros pontos com melhores valores também, mas que podem acontecer por acaso ou devido à dinâmica do algoritmo através das janelas e dos passos dados.

Esse resultado demonstra um bom primeiro passo, mas uma validação mais completa de todas as combinações de pontos, em todas as sequências, ainda é necessária para calibrar algum valor fixo, se é que será possível encontrar um padrão para todas as espécies, tanto em nucleotídeos quanto em aminoácidos.

4.2.4 Performance de Algoritmos

Como já citado nas seções dentro da Seção 4.2.2, os novos algoritmos criados possuem a performance bastante impactada pela diferença de tamanho entre as sequências comparadas ($S_{big} - S_{small}$). Enquanto isso, na Seção 4.2.1, os algoritmos que utilizam redimensionamento já dependem basicamente do custo de redimensionamento da representação para a representação de dados proposta. Um resumo sobre como é o comportamento da performance dos algoritmos propostos aqui em relação a $S_{big} - S_{small}$ e seu tempo estimado de execução está na Tabela 4.1.

Diferença (pb ou aa)	**Redim.**	***Deep Search***	**Novos**	**Controles**
nenhuma	minutos	segundos	segundos	segundos
menos de 10	minutos	segundos	minutos	segundos
dezenas	minutos	segundos	horas	segundos
centenas	minutos	segundos	dias	segundos
milhares	horas	minutos	meses	minutos

Tabela 4.1: Tempo de execução estimado entre algoritmos relativo a diferença entre tamanho das sequências comparadas ($S_{big} - S_{small}$).

Na Tabela 4.1 foram introduzidos os controles Blast e Clustal em uma única coluna por terem tempos de execução similares, assim como todos os

algoritmos de redimensionamento (Redim.) e os novos propostos (Novos). Também na tabela, é ilustrado que o algoritmo *Deep Search* é o único com tempo de execução similar aos controles. Isso porque o algoritmo *Deep Search* possui um método de indexação citado na Seção 4.2.1.1, mas esse método também possui um custo associado. O tempo de execução relativo ao tamanho das sequências para realização da indexação está apresentado na Tabela 4.2, uma vez que esse tempo é majoritariamente afetado por isso. Apesar de existir esse tempo, ele é bastante otimizado em segundos ou no máximo minutos para nossas sequências.

Tamanho (pb ou aa)	**Representação**	**Indexação *Deep Search***
dezenas	segundos	segundos
centenas	minutos	segundos
milhares	horas	minutos
dezenas de milhares	dias	minutos

Tabela 4.2: Tempo de execução estimado para criação da representação de dados (Representação) e para indexação do algoritmo Deep Search.

O custo de performance dos algoritmos propostos nesse trabalho também envolve o custo da criação das representações de dados, mas esse custo é proporcional ao tamanho das sequências e não a $S_{big} - S_{small}$. Quanto maior as sequências, mais tempo demora para gerar sua representação de dados conforme apresentado também na Tabela 4.2. Apesar de não impactar tanto grande parte dos algoritmos, esse tempo da criação da representação deve ser levando em consideração ao incorporar uma sequência nova no conjunto.

Assim, todos os algoritmos apresentados neste trabalho buscam melhor eficácia comparados aos controles dado que a eficiência não foi um foco de trabalho, mas pode ser considerado posteriormente. O único algoritmo que pode ter uma eficiência comparável aos controles é o *Deep Search*. Veremos a seguir como serão aplicados esses algoritmos no contexto biológico.

4.3 Novo Método *Alignment-free*

Na maioria das vezes, os problemas biológicos utilizam os alinhamentos para determinar similaridade das sequências como ilustrado na Fig 1.1 e os métodos que utilizam outros fins para isso são denominados *alignment-free*. Como utilizar as representações de dados e algoritmos desenvolvidos para chegar ao objetivo de encontrar a similaridade entre as sequências sem o uso de algoritmos de alinhamentos para esse fim, logo, foi desenvolvido um método *alignment-free*.

Sabe-se que grande parte dos problemas em biologia molecular são resolvidos através de similaridade, agrupamentos (*clustering*) ou busca (*ranking*) das sequências mais parecidas a uma outra. Para esses fins foi proposto o método *alignment-free*, conforme ilustrado na Fig 4.15. Nesse método, as sequências passam pela confecção das matrizes de características, depois têm a similaridade calculada par a par, gerando uma matriz de dissimilaridades entre elas. Por último, obtem-se todas as similaridades para aplicar uma busca em um *ranking* de similares ou agrupar com algoritmos de *clustering*. Tanto para o método de busca quanto para o de agrupamento existem controles bastante utilizados na literatura, em trabalhos de bioinformática, com o BLAST e com o Clustal Omega, para busca e agrupamento respectivamente.

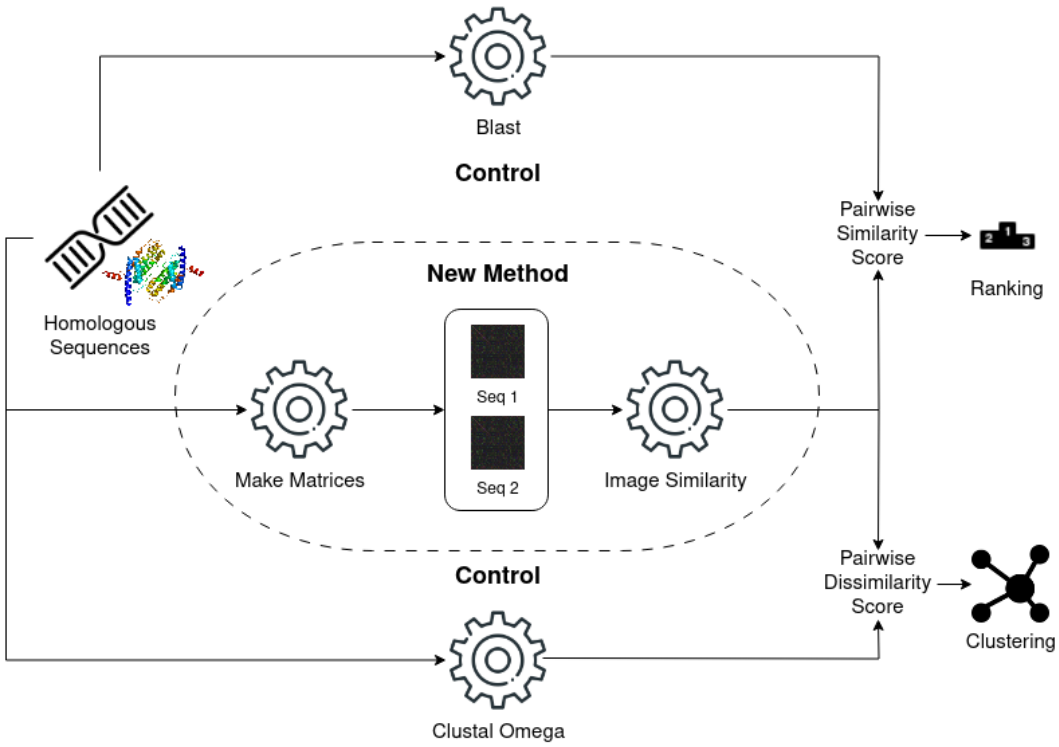


Figura 4.15: Metodologia *alignment-free* para similaridade, agrupamento (*clustering*) e busca (*ranking*) de sequências.

É possível notar na Fig 4.15 que para os controles e para o novo método as entradas dos algoritmos são diferentes. Para o novo método, inicialmente é necessário converter as sequências textuais na nova representação de dados para depois extrair a similaridade dessa representação, enquanto nos controles, já são utilizadas as similaridades através das sequências textuais diretamente. O mais demorado, nessa metodologia, é aplicar os algoritmos para encontrar as similaridades entre todas as sequências, uma vez fazendo isso, tanto a tarefa de similaridade de sequências, como o agrupamento e a busca se tornam bastante eficientes.

4.3.1 Aplicação nos Casos de Uso

Na aplicação da medida de similaridade, nos casos de uso em bioinformática, não foram utilizados os parâmetros otimizados na seção anterior, mas optou-se por manter o padrão, tendo em vista que é possível melhorar esses resultados, aplicando os parâmetros da otimização posteriormente. Ao aplicar os algoritmos de similaridade propostos para resolver problemas da bioinformática, está sendo construída uma metodologia *alignment-free*, como descrito na Seção 4.3. Nessa seção são ilustrados os resultados obtidos, resolvendo os problemas de:

- encontrar sequências mais similares dentro de um conjunto de dados;
- buscar sequências similares dentro de todos os conjuntos de dados e;
- agrupar conjuntos de dados de homólogos dados todas as sequências.

Durante a representação dos resultados mostrados, fez-se necessário abreviar os nomes dos conjuntos de dados de *Hemoglobin*$_\beta$, *Myoglobin*, *Neuroglobin*, *Cytoglobin*, *Androglobin* e *INDELible* para Hemo, Myo, Neuro, Cyto, Andro e Indeli respectivamente. Durante as análises, são referenciadas descobertas da Seção 3.1.1.3 sobre os conjuntos de dados.

4.3.1.1 Similaridade de Homólogos

Inicialmente se faz necessário entender se dado um grupo (conjunto de dados), é possível encontrar sequências similares que estão no mesmo grupo de homólogos. Este é um teste mais controlado em um conjunto menor de dados para tirar primeiros resultados.

Foi aplicada a metodologia em cada um dos conjuntos de dados dos experimentos e os resultados de RF para DNA estão na Tabela 4.3, assim como para PTNs na Tabela 4.4. Os resultados estão quebrados por camadas, visando o entendimento do peso de cada camada no resultado final, assim como seu agregado em *full*.

O resultado da agregação *full* não representa uma média das outras, dado que a medição é das similaridades dos dendrogramas gerados e não da matriz de dissimilaridades que dá origem aos dendrogramas. Cada camada vai contribuir para a dissimilaridade final, gerada na matriz de dissimilaridade entre todas as sequências, mas não é diretamente proporcional ao resultado final dos dendrogramas, uma vez que esses são feitos a partir do agrupamento hierárquico de todas as dissimilaridades da matriz, com uma sequência tendo influência das outras.

RF em DNA		**Hemo**	**Myo**	**Neuro**	**Cyto**	**Andro**	**Indeli**
SW Local		**0,00**	**0,00**	**0,00**	0,50	**0,50**	0,16
NW Global		**0,00**	**0,00**	**0,00**	0,67	0,83	0,16
UQI	**R**	**0,00**	0,08	0,08	0,92	0,92	0,19
	G	0,83	0,08	1	0,92	1	0,19
	B	0,17	0,17	0,08	0,92	0,92	0,19
	Full	0,83	**0,00**	1	0,92	1	0,19
R-SSIM	**R**	**0,00**	0,08	0,08	0,92	0,92	0,19
	G	**0,00**	0,08	0,08	0,92	0,92	0,19
	B	0,17	0,17	0,08	0,83	0,92	0,19
	Full	**0,00**	**0,00**	0,08	0,92	0,92	0,19
GS-SSIM	**R**	**0,00**	0,08	0,08	0,58	0,92	0,19
	G	**0,00**	0,08	0,08	0,58	0,92	0,19
	B	0,17	0,17	0,08	0,58	0,92	0,19
	Full	**0,00**	**0,00**	0,08	0,58	0,92	0,19
US-SSIM	**R**	**0,00**	0,08	0,33	0,33	1	0,19
	G	**0,00**	0,08	0,33	0,33	1	0,19
	B	0,17	0,17	0,33	**0,25**	1	0,19
	Full	**0,00**	**0,00**	0,33	**0,25**	1	0,19
RMS-SSIM	**R**	0,08	**0,00**	0,17	0,92	0,92	0,19
	G	**0,00**	0,08	0,17	0,92	0,92	0,19
	B	0,08	0,17	0,17	0,92	0,92	**0,14**
	Full	**0,00**	**0,00**	0,17	0,92	0,92	0,19
WMS-SSIM	**R**	0,08	**0,00**	0,33	0,42	0,83	0,19
	G	**0,00**	0,08	0,33	0,33	0,83	0,19
	B	0,08	0,17	0,42	0,42	0,83	**0,14**
	Full	**0,00**	**0,00**	0,33	0,33	0,83	0,19
Deep Search	**R**	0,33	0,17	0,17	0,83	0,92	0,95
	G	0,33	0,17	0,17	0,92	0,92	1
	B	0,33	0,25	0,08	0,92	0,92	0,97
	Full	0,08	0,08	**0,00**	0,92	0,92	0,92

Tabela 4.3: Resultado de RF para dendrogramas gerados com diferentes algoritmos de similaridade em nucleotídeos, incluindo SW de alinhamento local e NW de alinhamento Global, em comparação ao controle Clustal Omega. Valores em negrito indicam camadas com melhores resultados das metodologias em cada conjunto de dados.

A primeira interpretação a partir da Tabela 4.3 é que quase todos os algoritmos tiveram resultados perfeitos (0) no agregado *full* em *Hemoglobin*$_\beta$ e *Myoglobin*, exceto dois deles: *Deep Search* e UQI. O *Deep Search* não teve resultados perfeitos, mas conseguiu 0,08, que é um excelente resultado, agora o UQI teve um problema técnico em uma de suas camadas (G), que seria a característica relativa ao complemento da sequência contra ela mesma. Esse problema aconteceu somente no conjunto de dados de *Hemoglobin*$_\beta$ e *Neuroglobin* que impactou o agregado das camadas também. Essas camadas

RF em PTN		**Hemo**	**Myo**	**Neuro**	**Cyto**	**Andro**	**Indeli**
SW Local		0,25	0,25	**0,42**	0,58	**0,67**	0,19
NW Global		0,25	0,25	0,67	0,67	0,92	0,19
UQI	**R**	0,25	**0,17**	0,75	0,92	0,92	0,19
	G	0,17	0,33	0,67	0,92	0,92	0,19
	B	0,25	**0,17**	0,67	0,92	0,92	0,19
	Full	**0,00**	0,25	0,67	0,92	0,92	**0,16**
R-SSIM	**R**	0,17	**0,17**	0,75	0,92	0,92	0,19
	G	0,17	0,33	0,67	0,92	0,92	0,19
	B	0,25	0,25	0,67	0,92	0,92	**0,16**
	Full	0,08	**0,17**	0,67	0,92	0,92	0,19
GS-SSIM	**R**	0,17	**0,17**	0,75	0,83	0,83	0,19
	G	0,17	0,33	0,67	0,83	0,75	**0,16**
	B	0,25	0,25	0,67	0,83	0,83	**0,16**
	Full	0,08	**0,17**	0,67	0,83	0,83	0,19
US-SSIM	**R**	0,17	**0,17**	0,83	**0,08**	1	0,19
	G	0,17	0,33	0,75	0,42	1	0,19
	B	0,25	0,25	0,67	0,33	1	**0,16**
	Full	0,08	**0,17**	0,75	0,33	1	0,19
RMS-SSIM	**R**	0,08	**0,17**	0,58	0,92	0,92	**0,16**
	G	**0,00**	**0,17**	0,67	0,83	0,92	**0,16**
	B	0,25	**0,17**	0,67	0,92	0,92	0,19
	Full	**0,00**	**0,17**	0,58	0,92	0,92	0,19
WMS-SSIM	**R**	0,08	**0,17**	0,83	0,42	1	**0,16**
	G	**0,00**	**0,17**	0,92	0,42	1	**0,16**
	B	0,25	**0,17**	0,92	0,25	1	0,19
	Full	**0,00**	**0,17**	0,92	0,33	1	0,19
Deep Search	**R**	0,25	0,33	0,58	0,92	0,92	0,95
	G	0,08	0,50	0,58	0,83	0,92	0,97
	B	0,33	0,25	0,50	0,83	0,92	0,95
	Full	0,25	**0,17**	0,67	0,83	0,92	1

Tabela 4.4: Resultado de RF para dendrogramas gerados com diferentes algoritmos de similaridade em aminoácidos, incluindo SW de alinhamento local e NW de alinhamento Global, em comparação ao controle Clustal Omega. Valores em negrito indicam camadas com melhores resultados das metodologias em cada conjunto de dados.

permitem abrir o problema entre todas as características representadas. Com isso, isolar a camada G para um estudo mais aprofundado e a correção da característica, visto que sem ela, os resultados do UQI seriam ótimos.

O problema de instabilidade do algoritmo de UQI é conhecido e o SSIM veio corrigir isso com uma adaptação no código (WANG et al., 2004) em que basicamente duas constantes são adicionadas para dar mais estabilidade ao algoritmo. Em algumas circunstâncias, o UQI acaba gerando valores de similaridade muito grandes, virando *outliers* e, como as dissimilaridades acabam

tendo codependência, atrapalha em todos os resultados. Tanto os resultados de busca e agrupamento com UQI no DNA foram prejudicados por conta disso.

Mesmo com esse problema, o algoritmo demonstrou ótimos resultados, sendo, muitas vezes, melhor do que seu sucessor R-SSIM ou RMS-SSIM, como mostrado na Tabela 4.5. Essa diferença de performance é uma descoberta bastante interessante porque os algoritmos SSIM e MS-SSIM usados para implementar o R-SSIM e o RMS-SSIM são implementações paralelizadas no Tensorflow [3], enquanto o UQI utilizado não tinha uma versão em Tensorflow. Com isso, foi utilizado um sequencial produzido em Python [4]. Essa diferença sugere que existem divergências entre as implementações e oportunidades para melhoria de uma implementação paralela com mesmos resultados.

Voltando aos resultados da Tabela 4.3, é possível comentar sobre o algoritmo *Deep Search* em *Hemoglobin*$_\beta$, *Myoglobin*, *Neuroglobin* e *INDELible*, que alcançou sempre resultados piores nas camadas isoladas, mas quando as camadas eram agregadas, o resultado melhorava. Nestes quatro conjuntos de dados, aparentemente as camadas estão colaborando para um melhor *embedding* e como são conjuntos de dados de tamanhos similares, é possível manter uma base similar de comparação.

Nos outros conjuntos de dados, como tinham diferentes redimensionamentos das sequências que os compunham antes de virar um *embedding*, as camadas até pioraram os resultados, agregando ruído nos *embeddings*. Com os bons resultados, o *Deep Search* até conseguiu um valor perfeito em *Neuroglobin* (0), igual aos outros alinhamentos, sugerindo que uma nova técnica de redimensionamento, antes de fazer o *embedding*, pode ser realmente promissor.

Como o *Deep Search* só teve bons resultados nesse caso de uso, concluise que as dissimilaridades entre conjuntos de dados não estão corretamente definidas, mas que, dentro desses conjuntos, apresentam bons resultados. Isso sugere que o uso de uma nova medida de dissimilaridade, diferente da euclideana, pode tornar essa metodologia bastante competitiva com as outras propostas aqui.

Por outro lado, em *Cytoglobin* o algoritmo US-SSIM teve melhor compatibilidade com o Clustal (RF: 0,25) do que os alinhamento local (RF: 0,5) e global (RF: 0,67). Não só ele foi melhor que os alinhamentos na *Cytoglobin*, o WMS-SSIM também foi melhor que os alinhamentos (RF: 0,33) e o GS-SSIM foi melhor que o global, com uma RF de 0,58. Em *Androglobin*, todos os algoritmos performaram mal, mas comparando com os alinhamentos, todos foram bem distintos. Em especial, em comparação ao alinhamento global, o

[3] <tensorflow.org>
[4] <https://pypi.org/project/sewar>

WMS-SSIM conseguiu performance igual (0,83).

Ao investigar os conjuntos de dados, a *Androglobin* possui quase cinco mil (4726,4) nucleotídeos na média e um desvio padrão maior que muita sequência dos outros conjuntos de dados (694,56). Com isso, o baixo número de LCS (130) e a presença de *gaps*, *Androglobin* indica ser um conjunto de sequências que realmente apresenta muita divergência entre as técnicas de similaridades usadas.

Os algoritmos de redimensionamento UQI, R-SSIM e MS-SSIM conseguiram ótimos resultados com valores de RF de até 0,17, exceto o UQI na camada G de *Hemoglobin*$_\beta$ e *Neuroglobin* por conta da instabilidade. Inclusive, o RMS-SSIM teve o melhor resultado em *INDELible*, comparado ao Clustal, somente na camada B. No geral, não foi só o RMS-SSIM que teve esse resultado superior, mas também o WMS-SSIM, que no caso de sequências do mesmo tamanho, faz as mesmas comparações que o RMS-SSIM, dado que o primeiro não tem como andar com a janela e o segundo não precisa fazer redimensionamento. O mesmo se repete tanto para *Hemoglobin*$_\beta$ quanto para *Myoglobin* tanto em DNA quanto em PTN na Tabela 4.4.

Sobre as camadas, a camada B foi importante no *INDELible* para chegar mais perto dos resultados do Clustal, tanto no algoritmo US-SSIM, como WMS-SSIM (ambos com 0,14 na B comparados a 0,19 das outras). O mesmo acontece para *Cytoglobin* com o algoritmo US-SSIM que a camada B foi mais relevante que as outras (0,25 em relação a 0,33 das outras). Com isso, pode-se compreender que para essas sequências, com esses algoritmos, as diferenças foram mais relevantes do que as similaridades para determinar um melhor resultado. Isso faz bastante sentido, porque, como apresentado nas estatísticas dos conjuntos de dados, o *INDELible* tem o menor LCS de todos (DNA: 11, PTN: 5) e identidade zero no alinhamento com o Clustal, indicando muita diferença entre as sequências. Nos conjuntos de dados de mesmo tamanho e menos complexidade (*Hemoglobin*$_\beta$ e *Myoglobin*), a ordem sequencial (R) e a complementar (G) foram mais relevantes para determinar um dendrograma mais similar ao Clustal. Enquanto que a *Neuroglobin* parece ficar no meio do caminho de complexidade e possui resultados muito similares nas camadas, todos em cada algoritmo.

Analisando os resultados da Tabela 4.4 com PTNs, é possível visualizar grande discrepância entre os resultados do Clustal e os alinhamentos globais e locais. Não existe convergência em nenhum resultados deles. Os únicos resultados exatamente iguais ao Clustal são os dos algoritmos UQI, RMS-SSIM e WMS-SSIM para *Hemoglobin*$_\beta$. Para *Myoglobin*, os algoritmos implementados foram os que chegaram mais próximos do Clustal também com 0,17 de RF,

usando UQI (R e B), R-SSIM, GS-SSIM, US-SSIM, RMS-SSIM, WMS-SSIM e *Deep Search*. No geral, a camada das frequências de substituição (R) foi bem relevante para os resultados da *Myoglobin*, enquanto que para *Hemoglobin*$_\beta$ as duplicações foi mais relevante (G). A camada B de similaridade molecular também aparece relevante em alguns resultados de *INDELible* (0,16), mas, algumas vezes, em conjunto com a G. Teve outro local que a camada B foi relevante no conjunto de dados de *Cytoglobin* pelo algoritmo WMS-SSIM (0,25), conseguindo resultados melhores que os alinhamentos local (0,58) e global (0,67). Aliás, em *Cytoglobin*, o algoritmo US-SSIM (*full* e B: 0,33, R: 0,08) também foi superior.

Analisando o conjunto de dados de *Neuroglobin*, é ilustrado que o alinhamento local teve melhor performance (0,42), mas dado a divergência de RF e pelo resultado do alinhamento global (0,67), é correto afirmar que os algoritmos UQI (0,67), R-SSIM (0,67), GS-SSIM (0,67), RMS-SSIM (0,58) e *Deep Search* (B: 0,5, *full*: 0,67) possuem performance competitiva nesse conjunto. Em *Androglobin* que realmente os resultados são péssimos e o mais próximo do Clustal é o alinhamento local (0,67), sendo que o global é 0,92. Nesse caso, somente o algoritmo GS-SSIM (*full*: 0,83, G: 0,75) performou um pouco melhor, mas quase todos exceto o US-SSIM e o WMS-SSIM tiveram performance igual ao alinhamento global.

Como resultado importante, aqui são as camadas R e B relativos à frequência de substituição de aminoácidos e similaridade molecular, respectivamente. Ambos tiveram resultados melhores do que a análise de sequência da camada G, em todos os algoritmos no conjunto de *Myoglobin*, sendo melhores que os alinhamentos locais e globais (0,25 cada). Além disso, nos outros conjuntos de dados, as camadas R e B tiveram sempre resultados bastante competitivos com a camada G ou com os alinhamentos, sugerindo a importância das camadas com características mais diversificadas sobre os aminoácidos.

Por último, no caso do *INDELible*, quase todos ficaram muito parecidos, tanto os alinhamentos quanto os algoritmos, exceto o *Deep Search*. Com isso, o *Deep Search* consegue diferenciar homólogos dentro do seu conjunto quando eles são parecidos (*Hemoglobin*$_\beta$, *Myoglobin*, *Neuroglobin*), mas não quando são diferentes, ou possuem tamanhos diferentes, ou ainda, existem vários outros conjuntos de dados juntos. A limitação da diferença de tamanho e de conseguir diferenciar os grupos, tem relação com as distorções feitas nas representações para realizar o *embedding*, porque se não for redimensionado corretamente, resulta em uma extração precária de características. Com relação ao *INDELible*, os resultados mostram que o *embedding* não está conseguindo representar as poucas similaridades entre as sequências.

4.3.1.2 Busca de Homólogos

Ao realizar a tarefa de busca de homólogos dentro de todos os conjuntos de dados, foram ilustrados os valores de MAP tanto para DNA quanto PTN na Tabela 4.5. O entendimento é que se um algoritmo tiver boa performance nessa tarefa, ele consegue distinguir bem um conjunto de dados dos outros através das sequências que estão nele. Isso mostra a qualidade da distribuição de dispersão das distâncias entre os pares de sequência dadas pelos algoritmos.

MAP nas buscas		**Hemo**	**Myo**	**Neuro**	**Cyto**	**Andro**	**Indeli**
BLAST	**DNA**	1	1	1	1	1	0,66
	PTN	1	1	1	1	1	0,6
UQI	**DNA**	0,95	1	0,75	0,29	0,88	0,08
	PTN	1	1	1	0,73	0,31	0,98
R-SSIM	**DNA**	1	1	0,72	0,43	0,35	0,38
	PTN	1	1	0,69	0,18	0,05	0,06
RMS-SSIM	**DNA**	1	1	0,79	0,45	0,9	0,98
	PTN	1	1	0,8	0,59	0,9	0,98
WMS-SSIM	**DNA**	1	1	0,99	0,93	0,75	0,91
	PTN	1	1	0,99	0,93	0,75	0,01
GS-SSIM	**DNA**	1	1	0,69	0,93	0,51	0,08
	PTN	1	1	0,69	0,75	0,32	0,08
US-SSIM	**DNA**	1	1	0,99	0,93	0,75	0,89
	PTN	1	1	0,99	0,93	0,75	0
Deep Search	**DNA**	0	0,2	0,18	0,3	0,24	0,2
	PTN	0	0,2	0,18	0,3	0,24	0,2

Tabela 4.5: Resultado de MAP para busca de homólogos geradas com diferentes algoritmos de similaridade em DNA e PTN comparados com busca feita com BLAST. Valores mais altos são melhores.

Nos resultados, foi feito um comparativo com o resultado do BLAST tanto para DNA como para PTN, sendo que, em todos os conjuntos de dados, ele conseguiu concluir a tarefa, exceto nas sequências de *INDELible* (0,66 DNA e 0,6 PTN). Como essas sequências possuem baixo LCS (11 DNA e 5 PTN), torna-se realmente um desafio difícil correlacioná-las como sendo de um único conjunto de dados. Mesmo assim, os algoritmos UQI e RMS-SSIM conseguiram ótimos resultados em PTN (0,98) para encontrar todo o conjunto de dados, uma vez que as sequências de *INDELible* não variam seu tamanho. No caso do UQI, o mesmo não ocorreu com DNA (0,08), devido à instabilidade do algoritmo, que acabou por prejudicar todo seu resultado em DNA para busca de sequências. Aliás, esses resultados demonstram diferença na implementação serial do UQI em relação ao R-SSIM paralelo do Tensorflow, que conseguiu resultados bem piores no *INDELible* (0,38 DNA e 0,06 PTN).

Além do UQI com problema no DNA, o R-SSIM foi o único algoritmo, com redimensionamento, que conseguiu resultados piores que o controle no conjunto de dados de *INDELible*.

Inclusive, os resultados com *INDELible* revelam um padrão curioso em que os algoritmos utilizando DNA conseguiram resultados muito melhores do que os que utilizam PTN. Isso porque ao converter o DNA para PTN, os dados são simplificados, porque vários códons de DNA codificam o mesmo aminoácido, sugerindo que mesmo mantendo aminoácidos mais conservados, o padrão de nucleotídeos que gerou eles é importante. Com essa simplificação e dado que o *INDELible* naturalmente não possui quase nenhuma similaridade entre as suas sequências, os valores baixos de similaridade se confundem com comparações contra sequências de outros conjuntos de dados. Até no caso do RMS-SSIM (0,98 DNA e 0,98 PTN) e do WMS-SSIM (0,91 DNA e 0,01 PTN), que possuem mesmo comportamento em sequências de mesmo tamanho, tiveram resultados distorcidos nessa análise, porque eles possuem comportamento distinto, com sequências de diferentes tamanhos dos outros conjuntos. O WMS-SSIM ficou muito prejudicado nos resultados com PTNs por conta dos resultados de similaridade parecidos, assim como o US-SSIM (0,89 DNA e 0 PTN). Com isso, existe a indicação que, para a busca de sequências homólogas muito distantes, é interessante trabalhar com dados de DNA através dos algoritmos US-SSIM, WMS-SSIM ou RMS-SSIM; ou com PTN e algoritmos de redimensionamento multiescala como o RMS-SSIM. Tanto o *Deep Search* (0,2 DNA e PTN) quanto o GS-SSIM (0,08 DNA e PTN) tiveram resultados abaixo do esperado nessa detecção mais fina de diferenças.

Todos os algoritmos, exceto o *Deep Search*, conseguiram diferenciar muito bem os conjuntos de dados de *Hemoglobin*$_\beta$ e *Myoglobin* que são os conjuntos mais estáveis, sem variação de tamanho e *gaps*. Em particular, o *Deep Search* não conseguiu nenhum resultado a cima de 0,3 de MAP em todos os conjuntos de dados.

Incrementando o grau de complexidade, será comentado sobre os conjuntos de dados de *Neuroglobin*, *Cytoglobin* e *Androglobin*. Começando por *Neuroglobin*, em PTNs, o UQI conseguiu excelentes resultados (1,0) na busca de sequências, mas tendo alguns problemas com DNA (0,75) por conta da instabilidade do algoritmo. Como o conjunto de dados de *Neuroglobin* quase não possui diferença de tamanho, todos os algoritmos, exceto *Deep Search*, conseguiram bons resultados de MAP (a cima de 0,69). Em particular, os algoritmos US-SSIM e WMS-SSIM conseguiram tanto em PTN quanto em DNA, ótimos resultados (0,99). Os bons resultados do US-SSIM e WMS-SSIM permaneceram também em *Cytoglobin* (ambos com 0,93) e para DNA o GS-SSIM

conseguiu resultados similares com 0,93 de MAP, mas em PTN foi pior com 0,75. Isso indica que os algoritmos implementados (US-SSIM, WMS-SSIM, GS-SSIM) conseguem realizar busca de sequências homólogas com pouca ou nenhuma variação de tamanho. Principalmente os melhores foram o US-SSIM e o WMS-SSIM com resultados bem similares ao controle quando a variação de tamanho aumenta um pouco mais, como com a *Cytoglobin*.

Para *Androglobin*, o algoritmo RMS-SSIM conseguiu um excelente resultado de 0,9 tanto para DNA como PTN, mostrando que algoritmos multiescala com redimensionamento, como o RMS-SSIM, funcionam muito bem para busca de sequências homólogas com grande variação de tamanho, tendo resultados muito próximos ao controle (ambos com 1). Os algoritmos US-SSIM (0,75 em ambos) e WMS-SSIM (0,75 em ambos) também tiveram resultados bastante promissores tanto em DNA quanto em PTN, podendo ser otimizados no futuro.

4.3.1.3 Agrupamento de Homólogos

O resultados de agrupamento de conjuntos de dados de homólogos estão ilustrados na Tabela 4.6 com os valores comparativos para todos os algoritmos das métricas RF e BIM, em DNA e PTN, contra os dendrogramas obtidos com o Clustal Omega. Diferente do MAP, os valores mais baixos são os melhores. Para a tarefa de agrupamento, a métrica de BIM é mais relevante porque é importante saber se conseguiram formar grupos e não se esses grupos estão nas localizações corretas no dendrograma. Na tabela, os valores de BIM estão bons para quase todos os algoritmos tanto em PTN quanto em DNA, exceto no UQI em DNA (0,38) e *Deep Search* em ambos (0,49).

Todos os três algoritmos implementados: US-SSIM (BIM: 0,15 DNA e 0,21 PTN), WMS-SSIM (BIM: 0,17 DNA e 0,18 PTN) e GS-SSIM (BIM: 0,16 DNA e 0,22 PTN) tiveram os melhores resultados comparados ao controle, mas, além disso, os algoritmos de redimensionamento R-SSIM (BIM: 0,18 DNA e 0,23 PTN) e RMS-SSIM (BIM: 0,2 DNA e 0,22 PTN) também obtiveram resultados tão competitivos quanto. Isso é um bom indicativo de que todos os algoritmos estruturais podem agrupar devidamente os grupos de homólogos.

Na Fig 4.16, tem um comparativo ilustrativo dos agrupamentos feitos por Clustal e US-SSIM, utilizando os conjuntos de dados de DNA. Cada conjunto de dados foi colorido de uma cor própria para facilitar a distinção da posição que ficaram no dendrograma e são elas: vermelho para *Androglobin*, laranja para *Cytoglobin*, azul para *Neuroglobin*, marrom para *Myoglobin*, roxo para $Hemoglobin_{\beta}$ e tonalidades de verde para cada geração dentro do *INDELible*. No US-SSIM, somente um elemento (*Otolemur Garnetti*) do

Contra Clustal		**RF**	**BIM**
UQI	**DNA**	0,77	0,38
	PTN	0,46	0,23
R-SSIM	**DNA**	0,37	0,18
	PTN	0,47	0,23
RMS-SSIM	**DNA**	0,4	0,2
	PTN	0,45	0,22
WMS-SSIM	**DNA**	0,34	0,17
	PTN	0,37	0,18
GS-SSIM	**DNA**	0,34	0,16
	PTN	0,45	0,22
US-SSIM	**DNA**	0,3	0,15
	PTN	0,43	0,21
Deep Search	**DNA**	0,99	0,49
	PTN	0,99	0,49

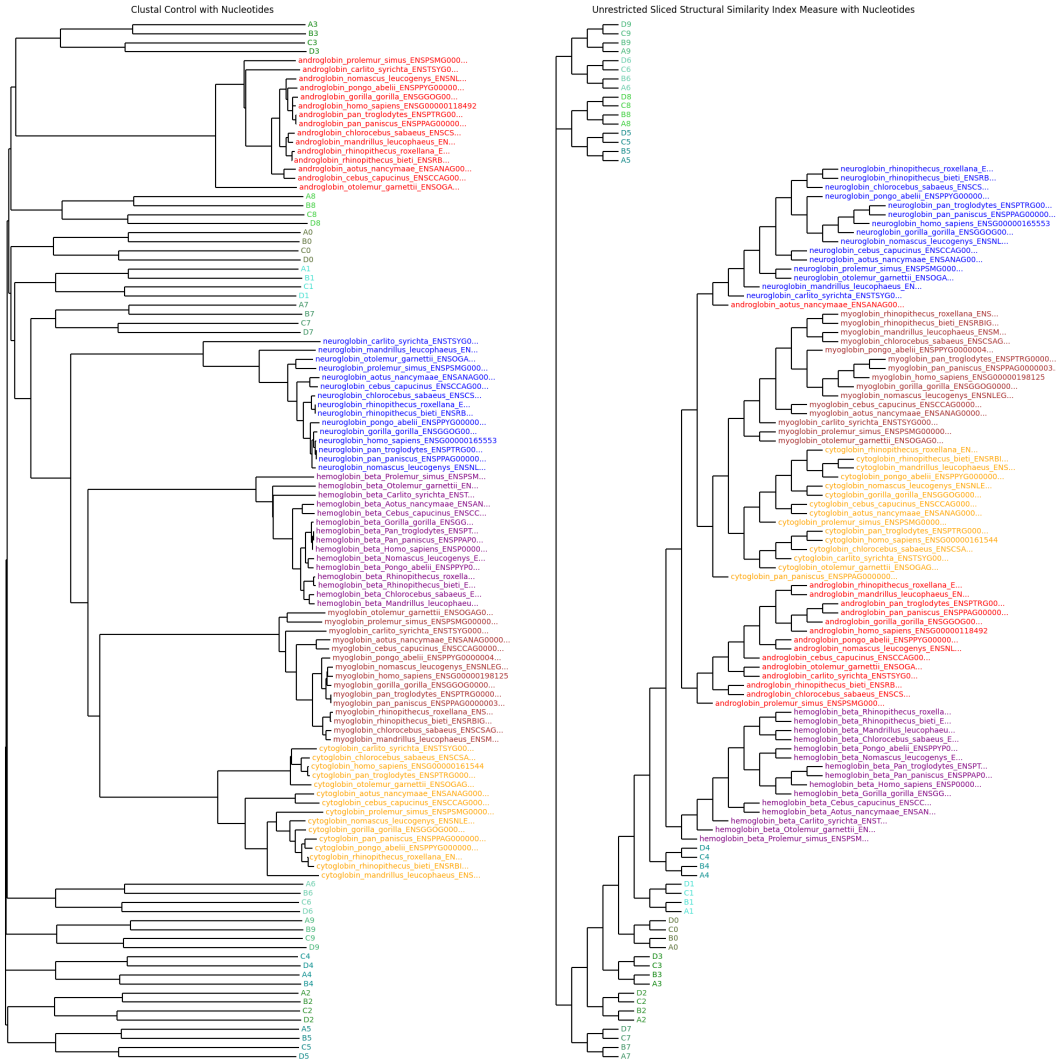
Tabela 4.6: Resultado de RF e BIM para agrupamentos gerados com diferentes algoritmos de similaridade em DNA e PTN, em comparação ao controle Clustal Omega. Valores mais baixos são melhores.

conjunto de dados *Androglobin* ficou fora do agrupamento correto. Na verdade, ele ficou junto com um conjunto de *Neuroglobin*, mas como último elemento mais desagrupado. Com uma rotação dos elementos azuis com os amarelos, *Androglobin* e *Neuroglobin* ficariam bem próximos.

Outro ponto importante da Fig 4.16, é que mostra as sequências de *INDELible* sempre bem agrupadas 4 a 4 em suas gerações. Tanto em DNA como em PTN, todos os algoritmos, exceto o *Deep Search* e o DNA do UQI, conseguiram fazer esse agrupamento das gerações, mas sem necessariamente juntar devidamente todos os *INDELible*. Nem mesmo o Clustal conseguiu definir o agrupamento dos *INDELible* tão bem, deixando claro a dificuldade da tarefa.

Novamente percebe-se a instabilidade do UQI no DNA (RF: 0,77 e BIM 0,38) e a péssima performance do *Deep Search* (RF: 0,99 e BIM: 0,49 em ambos DNA e PTN), mostrando que necessita de muitas melhorias para trabalhar com a representação de dados, nesse caso de uso.

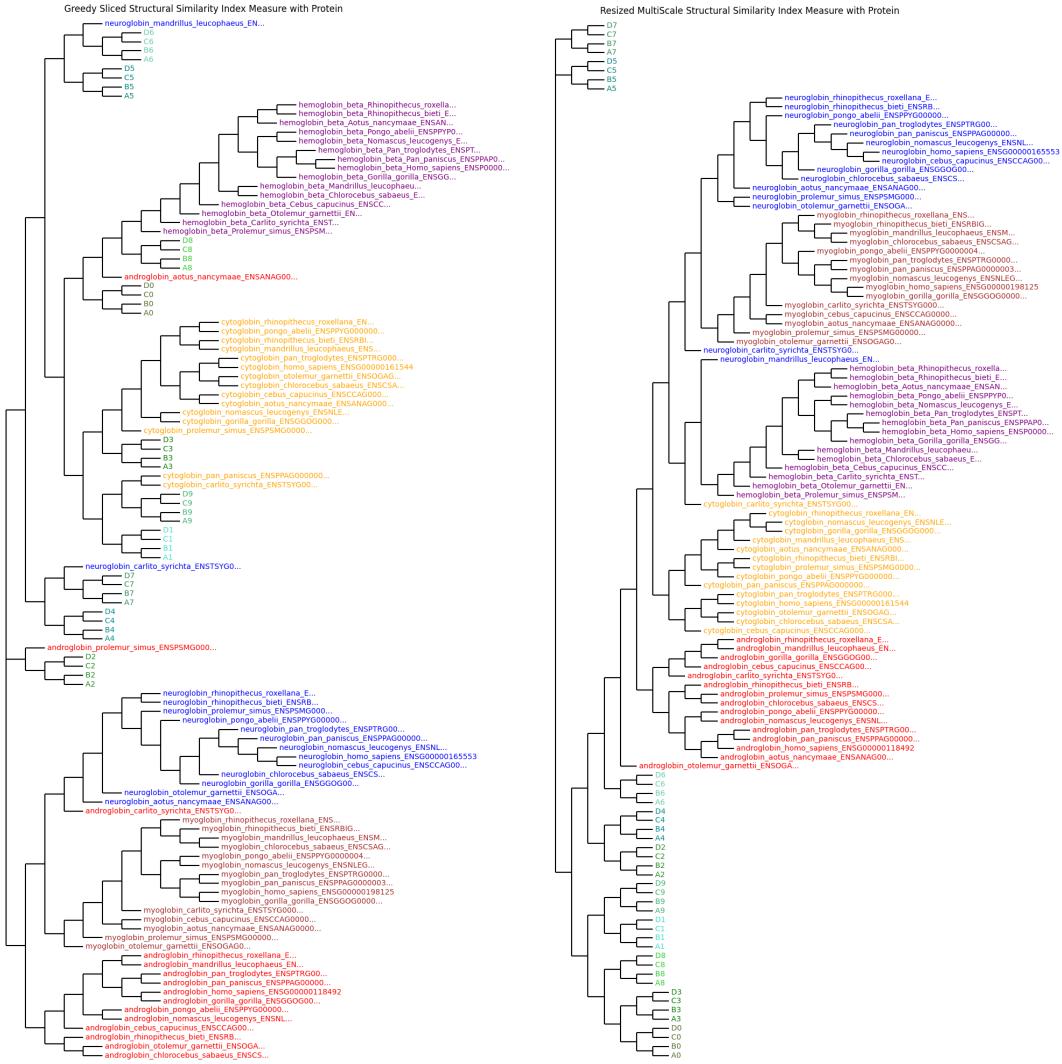
Os valores de RF também colaboram para o entendimento de como está a estrutura do dendrograma, mas não necessariamente descrevem devidamente o agrupamento. Como exemplo disso, foram ilustrados os agrupamentos feitos em PTN com o GS-SSIM e RMS-SSIM na Fig 4.17. Ambos estão com valor de RF inferior (0,45) ao que foi mostrado antes, no US-SSIM com DNA (0,3). Nesse caso, existe maior erro no agrupamento interno nas *Neuroglobin* (azul) do RMS-SSIM, assim como vários erros do lado do GS-SSIM.



(a) Clustal. (b) US-SSIM.

Figura 4.16: Agrupamentos de DNAs do US-SSIM comparativos ao Clustal.

Retornando na Tabela 4.5, observa-se que a diferença entre os dois algoritmos é o valor de MAP que também está influenciando no agrupamento, pois a performance do GS-SSIM é inferior, principalmente para *INDELible* (0,08 PTN), em tons de verde, e *Androglobin* (0,32 PTN), em vermelho, que são os mais desagrupados na Fig 4.17. Nesse caso, conclui-se a necessidade da medição do MAP para intuir se um algoritmo funciona para o agrupamento, uma vez que outros indivíduos não detectados, como no caso do *INDELible*, podem poluir o agrupamento. Durantes as buscas de homólogos, é importante conseguir separar os conjuntos de dados, sendo o mesmo o objetivo aqui, então a conclusão de que o MAP deve ser usado para avaliar agrupamentos é justificável.



(a) GS-SSIM, piores resultados de MAP. (b) RMS-SSIM, melhores resultados de MAP.

Figura 4.17: Agrupamentos de PTNs com resultados de BIM aceitáveis e RF abaixo do aceitável. Comparativo de MAPs.

4.3.2 Validação em AFProject

Uma vez que foram feitos os experimentos nos conjuntos de dados de Globinas e *INDELible*, fez-se necessário buscar mais locais para validação e comparação dos resultados com algum conjunto curado de controle. Com isso, é aplicada a tarefa de similaridade de homólogos, tentando encontrar as sequências mais similares em um conjunto de dados.

Para PTNs, é validado em onze conjuntos de dados com diversas famílias de PTNs do site AFProject, conforme descrito na Seção 3.1.2. Como os conjuntos de dados possuem um controle curado pelo AFProject, os controles dos experimentos anteriores (Clustal e Blast) são testados contra o controle

RF contra Controle		**ST01**	**ST02**	**ST03**	**ST04**	**ST05**
Clustal		0,21	0,13	0,42	0,19	0,12
SW Local		0,21	0,18	0,33	0,29	0,35
NW Global		0,41	0,24	0,39	0,35	0,69
UQI	**R**	0,85	0,71	0,83	0,79	0,88
	G	0,85	0,78	0,83	0,82	0,81
	B	0,85	0,71	0,81	0,79	0,88
	Full	0,8	0,71	0,83	0,78	0,88
US-SSIM	**R**	0,63	**0,36**	0,75	0,65	**0,54**
	G	0,68	**0,42**	0,75	0,62	**0,5**
	B	0,6	**0,4**	0,75	0,6	**0,46**
	Full	0,63	**0,31**	0,75	0,63	**0,46**
RMS-SSIM	**R**	0,85	0,67	0,81	0,79	0,85
	G	0,83	0,64	0,78	0,78	0,85
	B	0,85	0,64	0,81	0,79	0,85
	Full	0,83	0,67	0,81	0,76	0,85

(a) Os conjuntos de 01 até 05.

RF contra Controle		**ST07**	**ST08**	**ST09**	**ST10**	**ST11**	**ST12**
Clustal		0,27	0,56	0,61	0,19	0,30	0,35
SW Local		0,33	0,74	0,56	0,48	0,42	0,41
NW Global		0,52	0,79	0,72	0,65	0,65	0,41
UQI	**R**	0,84	0,90	0,89	0,81	0,81	0,76
	G	0,84	0,92	0,89	0,77	0,83	0,82
	B	0,78	0,90	0,86	0,81	0,83	0,76
	Full	0,80	0,87	0,86	0,81	0,82	0,76
US-SSIM	**R**	0,62	0,85	**0,72**	**0,48**	**0,60**	0,82
	G	0,62	**0,74**	**0,69**	**0,55**	**0,60**	0,88
	B	0,62	0,82	**0,69**	**0,45**	**0,58**	0,82
	Full	0,60	**0,72**	**0,67**	**0,48**	**0,58**	0,82
RMS-SSIM	**R**	0,76	0,87	0,92	0,81	0,80	0,82
	G	0,68	0,90	0,83	0,81	0,79	0,76
	B	0,76	0,87	0,83	0,77	0,80	0,71
	Full	0,74	0,85	0,86	0,77	0,79	0,76

(b) Os conjuntos de 07 até 12.

Tabela 4.7: Resultado de RF para dendrogramas gerados com diferentes algoritmos de similaridade em aminoácidos em comparação ao controle do AFProject. Valores em negrito indicam lugares onde os algoritmos se mostraram promissores.

curado do AFProject para validar nossa abordagem contra os métodos já existentes nos dados curados. Os nomes dos conjuntos de dados tiveram um zero removido para caberem na tabela, logo, ST001 virou ST01, ST002 virou ST02, e o mesmo para todos os outros.

Na Tabela 4.7, estão os resultados de RF para cada um dos conjuntos, cada um dos algoritmos e suas camadas. Só foram testados os algoritmos que

performaram melhor no conjunto de dados de PTN, sendo o UQI, US-SSIM e RMS-SSIM. Nesses dados, o algoritmo com resultados mais promissores foi o US-SSIM, especialmente para os conjuntos ST02 (0,31), ST05 (0,46), ST10 (0,48) e ST11 (0,58). Os conjuntos ST08 (0,72) e ST09 (0,67) estão com resultados muito discrepantes do controle, mas são muito próximos ou melhores do que os alinhamentos local (ST08: 0,74, ST09: 0,56), global (ST08: 0,79, ST09: 0,72) e Clustal (ST08: 0,56, ST09: 0,61), então, também podem ser resultados promissores.

	Clustal	**NW Global**	**SW Local**	**US-SSIM**			
				R	**G**	**B**	***Full***
ST001	0,04	0,13	0,04	0,24	0,27	0,23	0,24
ST002	0	0,05	0,02	**0,11**	**0,14**	**0,13**	**0,09**
ST003	0,07	0,05	0,03	0,23	0,23	0,23	0,23
ST004	0,09	0,17	0,15	0,32	0,31	0,30	0,31
ST005	0,06	0,33	0,17	**0,26**	**0,24**	**0,22**	**0,22**
ST007	0,06	0,18	0,09	0,23	0,23	0,23	0,22
ST008	0,28	0,39	0,36	0,41	0,36	0,40	0,35
ST009	0,30	0,35	0,27	0,35	0,34	0,34	0,32
ST010	0,09	0,31	0,23	**0,23**	**0,27**	**0,22**	**0,23**
ST011	0,15	0,32	0,21	**0,30**	**0,30**	**0,29**	**0,29**
ST012	0,14	0,17	0,17	0,36	0,39	0,36	0,36

Tabela 4.8: Resultado de BIM para dendrogramas gerados com US-SSIM em aminoácidos, em comparação aos alinhamentos e o controle do AFProject. Valores em negrito indicam lugares onde o algoritmo se mostrou promissor.

Do ponto de vista da métrica BIM, os mesmos conjuntos ST02 (0,09), ST05 (0,22), ST10 (0,23) e ST11 (0,29) possuem valores bem similares aos métodos de MSA local (ST02: 0,02, ST05: 0,17, ST10: 0,23, ST11: 0,21), global (ST02: 0,05, ST05: 0,33, ST10: 0,31, ST11: 0,32) e Clustal (ST02: 0, ST05: 0,06, ST10: 0,09, ST11: 0,15), sendo mais um indicador dos bons resultados do algoritmo US-SSIM aplicado nas representações de dados em PTN. Apesar da dificuldade, até para os algoritmos de alinhamento, o algoritmo US-SSIM conseguiu resultados competitivos em, pelo menos, 4 (ST002, ST005, ST010 e ST011) dos 11 conjuntos de dados, podendo inclusive ser promissor, caso seja otimizado em mais 2 (ST008 e ST009). Os dados estão representados na Tabela 4.8.

Na parte de DNA, só um conjunto de dados chamado FishMito também do AFproject foi avaliado. Para testar outros algoritmos, foram utilizados o R-SSIM e o WMS-SSIM somente com a camada *full*. O WMS-SSIM foi escolhido também por ter performado bem com sequências grandes nos casos de uso com a *Androglobin* e com *INDELible*. Os resultados estão sendo comparados com

Clustal	0,09
SW	0,23
NW	0,09
R-SSIM	0,86
WMS-SSIM	0,95

Tabela 4.9: Resultado de RF para dendrogramas gerados com algoritmos de similaridade em nucleotídeos com dados *full*, em comparação ao controle do AFProject para o conjunto de dados FishMito.

os outros algoritmos de alinhamento na Tabela 4.9. Os resultados são muito ruins, mas ao menos prova que é possível aplicar o algoritmo em sequências bem maiores. Era esperado um resultado ruim, uma vez que os algoritmos ainda não estão bem calibrados para sequências tão grandes, com tantos contextos e com tanta variação de tamanho. As sequências do FishMito possuem muitos genes lá dentro, enquanto o foco desta tese foi com sequências codificadoras de proteínas. Como o alinhamento global conseguiu um bom resultado de RF (0,09), era esperado que o algoritmo com uma característica mais global como o R-SSIM fosse melhor (0,86) do que o WMS-SSIM (0,95), que possui uma característica mais local.

5
Conclusões

As conclusões da tese estão divididas entre: um resumo passando pelos principais resultados do trabalho na Seção 5.1; depois do resumo, são pontuadas as principais contribuições na Seção 5.2; o trabalho deixou algumas oportunidades de pesquisa abertas, com a descrição delas na Seção 5.3; e por último, a tese é finalizada com as publicações realizadas durante o doutorado na Seção 5.4.

5.1
Resumo

Durante o trabalho, apareceram alguns pontos de limitação primordiais que existem ao lidar com dados de moléculas biológicas, sendo eles principalmente: mineração, extração, consolidação e compartilhamento de conhecimento dessas moléculas. Além desses pontos, existem outros mais atuais ligados à explicabilidade, à privacidade e ao seu uso mais simples em algoritmos de ML, visando *benchmarks* mais fáceis entre diversos algoritmos.

Na busca pela resolução desses problemas, surgiu a oportunidade para criação de uma nova modelagem física para o desenvolvimento de características em camadas, a fim de serem utilizadas na bioinformática. Essa modelagem física foi validada para nucleotídeos e aminoácidos, utilizando-a para criar representações de dados biológicos, relativos a DNA, RNA e PTN, que colaborem com a segurança e privacidade dos dados. A importância dessa representação é que ela é agregativa, logo, pesquisadores podem focar esforços em agregar informações de camadas com diferentes características e entender o que descreve melhor cada molécula. Como a modelagem física foi feita utilizando a premissa de que os blocos-padrão de nucleotídeos ou aminoácidos devem ser descritos de forma *bottom-up*, todas as características podem se encaixar nela.

Outro ponto interessante dessa representação é que tanto pode ser utilizada com uma única camada, como com várias combinadas, dependendo da semântica que deseja compor. Tanto uma camada possui significado sozinha, como em conjunto com outras. Além disso, as representações compondo outras características, além da sequência de caracteres feita nas PTNs, mostraram grande importância na resolução adequada dos casos de uso da bioinformática.

Com essa representação de dados em camadas tem-se a explicabilidade de como os resultados foram obtidos de forma clara e objetiva sem a necessidade da aplicação de um algoritmo diferente para inferir isso. Essa explicabilidade

ajudou a compreender o problema do algoritmo UQI, na camada G, que impactou todas as análises, como também pode ajudar com outras técnicas que sejam aplicadas na representação.

Visto que foram criadas essas representações de dados, é preciso validar se elas realmente representavam as entidades biológicas. Para isso, foram utilizadas duas alternativas: aplicar técnica de *embedding* nas representações através de algoritmos de DL, que consigam extrair um vetor através delas e; criar algoritmos de similaridade que consigam comparar as representações propostas.

A técnica de *embedding* testada com *Deep Search* se mostrou extremamente rápida e promissora, no caso de uso de similaridade de sequências, conseguindo tornar as camadas das representações mais eficazes quando combinadas. Os resultados de eficácia foram bastante ruins em outros casos de uso, comparado com outros algoritmos desenvolvidos e os métodos de MSA, deixando em aberto muitas oportunidades de melhoria.

Ao todo, além dos *embeddings*, foram desenvolvidos e apresentados três novos algoritmos melhores ou iguais aos algoritmos de redimensionamento aplicados. Dentro desses três algoritmos, um é de busca por máximos locais totalmente paralelizável (WMS-SSIM), mas ainda com um pouco de contexto global e outros dois realmente são glocais (GS-SSIM e US-SSIM), otimizando tanto contexto local como global de comparação. Desses glocais, o US-SSIM consegue ser paralelizável facilmente em GPUs por não ter dependências de processamento com comparações internas e não ter dependência de compartilhamento de memória durante o processamento. Do outro lado, o GS-SSIM é mais conservador e não possibilita que translocações e inversões ocorram, mas possui dependência de comparações sequenciais, sendo menos paralelizável.

Os algoritmos de similaridade tiveram bons resultados em todos os casos de uso de bioinformática aplicados neste trabalho, principalmente com *INDELible* simulando homólogos distantes. Observa-se que US-SSIM, WMS-SSIM, RMS-SSIM conseguiram detectar pequenas modificações no DNA e identificar melhor que o BLAST na busca de homólogos para DNA. Para PTN, somente o RMS-SSIM conseguiu detectar as diferenças de forma satisfatória. De todos os novos algoritmos, o GS-SSIM é o mais rápido, mas não teve resultados tão bons nos casos de uso, estando aberto a melhorias.

Como foi possível completar os casos de uso em bioinformática, introduziu-se mais uma contribuição com novas opções de metodologia para construção de algoritmos *alignment-free*. Essa metodologia *alignment-free* foi comparável aos resultados dos controles utilizados e consegue se beneficiar de outras informações além das letras das sequências, mantendo a semântica

das características das moléculas. Esses algoritmos de similaridade de dentro da metodologia conseguem ser comparáveis a um alinhamento, informando as áreas mais importantes da representação, contribuindo para explicabilidade dos resultados.

Como trabalho interdisciplinar, além da computação, existem muitas contribuições para biologia, uma vez que há uma representação de dados que facilite a colaboração entre os pesquisadores e representação de moléculas biológicas. Através da representação com características das sequências, já é possível trazer descobertas com um método *alignment-free* inspirado em *dot plots*, que são tão familiares aos biólogos. A técnica ter sido validada no AFProject, conseguindo bons resultados para representação de PTN em alguns conjuntos de dados já nos incentiva a buscar por melhorias no trabalho.

5.2 Principais Contribuições

Esta tese traz duas contribuições principais e diversos resultados secundários. A primeira contribuição é a modelagem física em camadas para moléculas biológicas, com as representações de dados de sequências para DNA e PTN. Dentro dessas representações existem contribuições secundárias:

- cada ponto em uma camada descreve a característica de um nucleotídeo (DNA ou RNA) ou aminoácido (PTN);
- cada camada é reutilizável e pode seguir um padrão de utilização por diversos programas;
- cada camada descreve uma característica da molécula biológica que pode ser compartilhada;
- cada camada é autocontida e, ao mesmo tempo possível de combinar com outras;
- ao desenvolver a camada pode ter privacidade maior com *data masking*;
- como cada camada representa uma característica, contribui para explicabilidade ao ser usada;
- a combinação das camadas habilita uma visão holística de todas as características de uma molécula, uma vez que sejam modeladas;
- além do uso computacional, as camadas podem ser combinadas e exploradas visualmente.

A segunda grande contribuição da tese são os métodos comparativos para representações de sequências biológicas. Nesses métodos existem algumas contribuições secundárias:

- três novos algoritmos (WMS-SSIM, GS-SSIM e US-SSIM) para medir similaridade entre sequências biológicas de tamanhos distintos;
- quatro aplicações de algoritmos só com redimensionamento nas representações (UQI, S-SSIM e RMS-SSIM, *Deep Search*) são eficientes para comparar sequências de mesmo tamanho;
- todos os algoritmos são flexíveis para dar pesos e lidar individualmente com cada camada;
- uso de otimização bayesiana, atingir eficácia ainda melhor com algoritmos de similaridade através de *tunning* de parâmetros;
- extração eficiente de características através de *embeddings*;
- uso de ANN (*Deep Search*) em representações com potencial melhoria na eficiência dos métodos de similaridade;
- criação de metodologia *alignment-free* com resultados comparáveis a métodos tradicionais de bioinformática;
- nova metodologia sugere maior eficácia comparado ao BLAST em busca de homólogos distantes.

5.3 Trabalhos Futuros

Assim como mostrado no resultado com FishMito, existem sequências muito grandes que os algoritmos ainda não conseguem lidar. Existe a possibilidade de quebrar as sequências em entidades menores, tal qual genes, para lidar com elas, ou pensar em como adaptar as representações para funcionar com sequências que tenham vários contextos. Com isso, fica em aberto a necessidade de explorar os algoritmos em uma maior diversidade de sequências, tais como genomas, outros tipos de proteínas ou até proteínas que não tenham estrutura secundária.

Podem existir genes que sejam ainda assim grandes com um mesmo contexto e isso continua sendo um ponto não resolvido neste trabalho. Uma vez que as representações de dados já estejam bem estabelecidas, é possível, futuramente, aplicar redução de dimensionalidade nas representações de dados para facilitar a troca e manuseio de sequências muito grandes.

Um outro ponto de melhoria das representações é adequar as formas de lidar com caracteres de ambiguidade nas camadas que não tem algo pronto. Existem padrões em biologia molecular para trabalhar com ambiguidade dentro de sequências de DNA, RNA e PTNs, mas só as camadas R e B de PTN estão lidando com essas ambiguidades atualmente nas representações.

Na representação de dados, caso sejam adicionadas mais camadas, é possível lidar individualmente com cada uma delas e criar tratativas específicas ou pesos para elas. Esse tratamento mais acurado para as camadas também é um trabalho em aberto, que influencia a forma como estão sendo montados os bancos de dados biológicos. Esses dados agora podem ter um peso para cada característica em cada organismo.

Sobre engenharia de *features* para criação de cada representação, seria possível utilizar as 134 características que o índice de *Sneath* leva em consideração separadas por camada, ou ainda avaliar características menos ligadas a sequência, mas sim a atividade da molécula nos organismos. Além disso, vale compreender se existem padrões que estão relacionados somente a moléculas reais e não artificiais como o *INDELible*.

As camadas foram implementadas nesse trabalho sem medir a correlação real entre elas. Seria interessante algum método para avaliar se uma característica possui alta correlação com outras implementadas para evitar duplicidades na representação completa da molécula. Inclusive, esse método poderia ser interessante para identificar interdependência entre camadas.

Ainda voltado para análise das camadas, vale entender se realmente existe uma assinatura única dessas camadas por molécula ou se podem existir duas ou mais moléculas com a mesma representação e o quanto isso impacta em aplicações da bioinformática. Pode ocorrer uma duplicidade caso as sequências possuam padrões parecidos, mas pb ou aa diferentes, como por exemplo as sequências AAATT e CCCGG caso não existam características distinguindo elas.

Estas representações em camadas trazem outras questões de bancos de dados ao abrir portas para padronização de características de sequências, indexação, busca e outras tarefas. Com isso, pode-se pensar ainda em um banco de dados, em camadas, que suporte a modelagem física proposta aqui.

A modelagem em camadas das moléculas possibilita a representação física de cada característica e abre espaço para o tratamento único de cada uma delas ou o uso de todas em conjunto dentro de um *Foundation Model* focado na resolução de tarefas de biologia molecular. Um dos maiores problemas atuais é a adição de diversas características nos LLMs sem um padrão.

Algumas das melhorias que são possíveis como novas áreas de pesquisa para o *Deep Search* são: implementação de nova medida de distância ao invés de euclideana; melhorar forma de lidar com representações de diversos tamanhos, dado que o redimensionamento com borda preta não necessariamente é a melhor forma de indexar as representações; o ajuste fino (*fine-tunning*) no modelo VGG utilizado com essas representações de dados ou a construção

de outros modelos melhores, podendo buscar até dentro da nova área de LLMs; e conseguir destacar no *embedding* as diferenças entre as sequências nos agrupamentos, que no trabalho foram os seus piores resultados.

Posto que os algoritmos de similaridade foram construídos, pensando em dimensões menores, e que a "maldição da dimensionalidade" é algo mais complexo de lidar para métodos de distância e similaridade (VERLEYSEN; FRANÇOIS, 2005), foi planejado deixar isso para trabalhos futuros. Outra melhoria nos algoritmos de similaridade pode ser a transformação deles em algoritmos de distância, levando em consideração a desigualdade triangular.

Os algoritmos aplicados no trabalho, demonstraram bons resultados ao serem otimizados com as representações de nucleotídeos, mas seria necessário um trabalho mais aprofundado nessas otimizações de hiperparâmetros para explorar algumas coisas: a melhor forma de otimizar, melhor comparativo, parâmetros por organismos, dentre outros *insights* que podem aparecer.

Olhando para otimização de performance computacional dos algoritmos e não só de resultados, existem oportunidades em aberto para tornar a metodologia totalmente paralelizável em GPU e não só em CPU. Essa tarefa está mais avançada porque grande parte do trabalho já foi desenvolvido, utilizando Tensorflow, que nativamente já possui ótimo suporte de paralelismo em GPU.

Refletindo sobre novas áreas de pesquisa, além das contribuições para biologia, essa nova metodologia é extensível a pesquisas em linguística. Utilizando uma abordagem bastante similar, é possível comparar qualquer sequência que não se saiba a semântica, como por exemplo, na identificação e tradução de sequências de caracteres em textos antigos, como foi estudado recentemente com o idioma Akadiano utilizando algoritmos de inteligência artificial (GUTHERZ et al., 2023).

Por fim, entende-se que este trabalho é um primeiro tijolo na construção de uma bioinformática mais colaborativa, segura e privativa, através das representações propostas em camadas. Foi dado mais um passo para conseguir utilizar algoritmos de bioinformática e ML em dados biológicos de forma reutilizável, ao invés de tentar adaptar metodologias existentes de texto e imagem para lidar com eles.

5.4 Publicações

Durante o desenvolvimento da tese foram publicados trabalhos em diversas áreas do conhecimento ligadas a bioinformática até o tema final dela surgir. Quatro trabalhos foram publicados em locais bem conceituados pelo *Qualis* da

CAPES e um *software* para a comunidade científica.

A jornada iniciou inspirada pela dificuldade para a aplicação de ML utilizando diversas *features* na área biológica, mas através de um tema mais atual na área médica e que, talvez no futuro, seja um problema também em biologia molecular. O primeiro artigo foi publicado na revista GigaScience (*Qualis* A1) e deu origem a um *framework* multi-agente para detecção de *drifts* (mudança de padrão) de dados chamado Driftage (VIEIRA et al., 2021). O *framework* permite que a camada de extração de dados e a camada de análise de *drifts* sejam desacopladas simplificando a forma de desenvolver detectores de *drift*. Ele foi validado com dados de monitoramento cardíaco abertos e publicado para a comunidade [1].

Enquanto compreendia o funcionamento dos *workflows* de processamento de dados biológicos, foi feita uma publicação em uma das maiores conferências em bioinformática do Brasil, o *Brazilian Symposium on Bioinformatics* (BSB) com *Qualis* B1, através de uma parceria com o INCA. A publicação é um sistema gerenciador de *workflows* focado especificamente na detecção de genes cancerígenos (VIEIRA et al., 2022). Como os dados biológicos e as técnicas aplicadas dependem de um especialista para análise no meio do *workflow*, ele foi desenhado com a proposta de trazer uma pessoa para o meio do processo (*Human in the Loop*). Com isso, o especialista pode interagir com os grafos gênicos, agregando conhecimento das escolhas e no futuro conseguindo prever genes através de dados históricos.

Depois disso, foi lançado o primeiro dos trabalhos, na BSB do ano seguinte, com o mesmo olhar *bottom-up*, mas com foco das moléculas em sistemas dinâmicos modelados através de redes de Petri (HAEUSLER et al., 2023). Logo em seguida, foi submetida e aceita uma prévia desta tese como *full paper* no Simpósio Brasileiro de Banco de Dados (SBBD) 2023 [2] (*Qualis* A4). Todo conteúdo desta tese ainda será publicado, mas a prévia de alguns resultados foi bem aceita pela comunidade científica até o momento.

[1] <https://github.com/dmvieira/driftage>
[2] <https://sbbd.org.br/2023/sbbd-fullpapers-program/>

6
Referências bibliográficas

ABOUELMEHDI, K.; BENI-HESSANE, A.; KHALOUFI, H. Big healthcare data: preserving security and privacy. **Journal of Big Data**, SpringerOpen, v. 5, n. 1, p. 1–18, 12 2018. ISSN 21961115. Citado na página 29.

AKIBA, T. et al. Optuna: A Next-generation Hyperparameter Optimization Framework. **Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, Association for Computing Machinery, p. 2623–2631, 7 2019. Disponível em: <https://dl.acm.org/doi/10.1145/3292500.3330701>. Citado na página 48.

ALCALÁ-CORONA, S. A. et al. Modularity in Biological Networks. **Frontiers in Genetics**, Frontiers Media S.A., v. 12, p. 701331, 9 2021. ISSN 16648021. Citado na página 29.

ALHIJAWI, B.; AWAJAN, A.; FRAIHAT, S. Survey on the Objectives of Recommender Systems: Measures, Solutions, Evaluation Methodology, and New Perspectives. **ACM Computing Surveys**, Association for Computing Machinery, v. 55, n. 5, p. 1–93, 12 2023. ISSN 15577341. Citado na página 45.

ALI, R. H. et al. Identifying Clusters of High Confidence Homologies in Multiple Sequence Alignments. **Molecular Biology and Evolution**, Oxford University Press, v. 36, n. 10, p. 2340, 10 2019. ISSN 15371719. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6933875/>. Citado na página 23.

ALTSCHUL, S. F. BLAST Algorithm. In: **eLS**. Wiley, 2014. Disponível em: <https://onlinelibrary.wiley.com/doi/10.1002/9780470015902.a0005253.pub2>. Citado na página 25.

AO, C. et al. Biological Sequence Classification: A Review on Data and General Methods. **Research**, American Association for the Advancement of Science (AAAS), v. 2022, 1 2022. ISSN 26395274. Disponível em: <https://spj.science.org/doi/10.34133/research.0011>. Citado na página 26.

ARSHAD, S. et al. Analysis of security and privacy challenges for DNA-genomics applications and databases. **Journal of Biomedical Informatics**, Academic Press, v. 119, p. 103815, 7 2021. ISSN 1532-0464. Citado na página 29.

ASHBURNER, M. et al. Gene Ontology: tool for the unification of biology. **Nature genetics**, NIH Public Access, v. 25, n. 1, p. 25, 5 2000. ISSN 10614036. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3037419/>. Citado na página 18.

BAKUROV, I. et al. Structural similarity index (SSIM) revisited: A data-driven approach. **Expert Systems with Applications**, Pergamon, v. 189, p. 116087, 3 2022. ISSN 0957-4174. Citado na página 43.

BALDAUF, S. L. Phylogeny for the faint of heart: a tutorial. **Trends in genetics : TIG**, Trends Genet, v. 19, n. 6, p. 345–351, 6 2003. ISSN 0168-9525. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/12801728/>. Citado 2 vezes nas páginas 9 e 24.

BALLARD, C. et al. Data warehousing, annotation and statistical analysis system. **Career: Data and Analytics**, Patrick Brown and David Botstein, v. 27, n. 11, p. 199, 10 2002. ISSN 2212-0173. Citado na página 27.

BEIS, J. S.; LOWE, D. G. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, IEEE, p. 1000–1006, 1997. ISSN 10636919. Citado na página 42.

BERG, S. et al. ilastik: interactive machine learning for (bio)image analysis. **Nature Methods**, Nature Publishing Group, v. 16, n. 12, p. 1226–1232, 9 2019. ISSN 1548-7105. Disponível em: <https://www.nature.com/articles/s41592-019-0582-9>. Citado na página 39.

BERMAN, D. S. et al. MutaGAN: A sequence-to-sequence GAN framework to predict mutations of evolving protein populations. **Virus Evolution**, Oxford University Press, v. 9, n. 1, 2023. ISSN 20571577. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10104372/>. Citado na página 56.

BERNASCONI, A. et al. PoliViews: A comprehensive and modular approach to the conceptual modeling of genomic data. **Data & Knowledge Engineering**, North-Holland, p. 102201, 6 2023. ISSN 0169-023X. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0169023X23000617>. Citado na página 30.

BERNSTEIN, F. C. et al. The Protein Data Bank: a computer-based archival file for macromolecular structures. **Journal of molecular biology**, J Mol Biol, v. 112, n. 3, p. 535–542, 5 1977. ISSN 0022-2836. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/875032/>. Citado na página 17.

BLACKSHIELDS, G. et al. Sequence embedding for fast construction of guide trees for multiple sequence alignment. **Algorithms for Molecular Biology**, BioMed Central, v. 5, n. 1, p. 1–11, 5 2010. ISSN 17487188. Disponível em: <https://almob.biomedcentral.com/articles/10.1186/1748-7188-5-21>. Citado na página 24.

BORNBERG-BAUER, E.; PATON, N. W. Conceptual data modelling for bioinformatics. **Briefings in bioinformatics**, Brief Bioinform, v. 3, n. 2, p. 166–180, 2002. ISSN 1467-5463. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/12139436/>. Citado na página 30.

BOROWIEC, M. Spruceup: fast and flexible identification, visualization, and removal of outliers from large multiple sequence alignments. **Journal of Open Source Software**, v. 4, n. 42, p. 1635, 10 2019. ISSN 2475-9066. Disponível em: <https://joss.theoj.org/papers/10.21105/joss.01635>. Citado na página 25.

BOZORGPANAH, A.; TORRA, V.; ALIAHMADIPOUR, L. Privacy and Explainability: The Effects of Data Protection on Shapley Values. **Technologies 2022, Vol. 10, Page 125**, Multidisciplinary Digital Publishing Institute, v. 10, n. 6, p. 125, 12 2022. ISSN 2227-7080. Disponível em: <https://www.mdpi.com/2227-7080/10/6/125>. Citado na página 29.

BRUDNO, M. et al. Glocal alignment: finding rearrangements during alignment. **Bioinformatics**, Oxford Academic, v. 19, n. suppl_1, p. i54–i62, 7 2003. ISSN 1367-4803. Disponível em: <https://academic.oup.com/bioinformatics/article/19/suppl_1/i54/227687>. Citado na página 65.

BRUNET, D.; VRSCAY, E. R.; WANG, Z. On the Mathematical Properties of the Structural Similarity Index. **IEEE TRANSACTIONS ON IMAGE PROCESSING**, v. 21, n. 4, 2012. Disponível em: <http://ieeexplore.ieee.org.> Citado na página 28.

BURMA, P. K. et al. Genome analysis: A new approach for visualization of sequence organization in genomes. **Journal of Biosciences**, Springer India, v. 17, n. 4, p. 395–411, 12 1992. ISSN 02505991. Disponível em: <https://link.springer.com/article/10.1007/BF02720095>. Citado na página 28.

CAPELLA-GUTIÉRREZ, S.; SILLA-MARTÍNEZ, J. M.; GABALDÓN, T. trimAl: A tool for automated alignment trimming in large-scale phylogenetic analyses. **Bioinformatics**, v. 25, n. 15, p. 1972–1973, 8 2009. ISSN 13674803. Citado na página 25.

CASTRESANA, J. Selection of Conserved Blocks from Multiple Alignments for Their Use in Phylogenetic Analysis. **Molecular Biology and Evolution**, Oxford Academic, v. 17, n. 4, p. 540–552, 4 2000. ISSN 0737-4038. Disponível em: <https://dx.doi.org/10.1093/oxfordjournals.molbev.a026334>. Citado na página 20.

CERI, S. et al. Overview of GeCo: A project for exploring and integrating signals from the genome. **Communications in Computer and Information Science**, Springer Verlag, v. 822, p. 46–57, 2018. ISSN 18650929. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-319-96553-6_4>. Citado na página 30.

CHURCH, K. W. Word2Vec. **Natural Language Engineering**, Cambridge University Press, v. 23, n. 1, p. 155–162, 1 2017. ISSN 1351-3249. Disponível em: <https://www.cambridge.org/core/journals/natural-language-engineering/article/word2vec/B84AE4446BD47F48847B4904F0B36E0B>. Citado 3 vezes nas páginas 9, 41 e 42.

COLLETT, J.; PEARCE, S. GCAT dotplots characterize precisely and imprecisely defined topological features of DNA. **bioRxiv**, Cold Spring Harbor Laboratory, p. 2021.07.29.454305, 7 2021. Disponível em: <https://www.biorxiv.org/content/10.1101/2021.07.29.454305v1>. Citado na página 39.

DALLA-TORRE, H. et al. The Nucleotide Transformer: Building and Evaluating Robust Foundation Models for Human Genomics. **bioRxiv**, Cold Spring Harbor

Laboratory, p. 2023.01.11.523679, 3 2023. Disponível em: <https://www.biorxiv.org/content/10.1101/2023.01.11.523679v2>. Citado na página 19.

DAUGELAITE, J.; DRISCOLL, A. O.; SLEATOR, R. D. An Overview of Multiple Sequence Alignments and Cloud Computing in Bioinformatics. **ISRN Biomathematics**, Hindawi Limited, v. 2013, p. 1–14, 8 2013. Citado na página 25.

DELIBAS, E. A new feature vector model for alignment-free DNA sequence similarity analysis. **Sigma Journal of Engineering and Natural Sciences – Sigma Mühendislik ve Fen Bilimleri Dergisi**, Kare Publishing, 0 2022. Citado na página 25.

DESCHAVANNE, P. J. et al. Genomic signature: characterization and classification of species assessed by chaos game representation of sequences. **Molecular Biology and Evolution**, Oxford Academic, v. 16, n. 10, p. 1391–1399, 10 1999. ISSN 0737-4038. Disponível em: <https://dx.doi.org/10.1093/oxfordjournals.molbev.a026048>. Citado na página 28.

DICK, K.; GREEN, J. R. Chaos Game Representations & Deep Learning for Proteome-Wide Protein Prediction. In: **2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)**. IEEE, 2020. p. 115–121. ISBN 978-1-7281-9574-2. Disponível em: <https://ieeexplore.ieee.org/document/9288051/>. Citado na página 27.

FITCH, W. M. Distinguishing Homologous from Analogous Proteins. **Systematic Zoology**, v. 19, n. 2, p. 99, 6 1970. ISSN 00397989. Disponível em: <https://academic.oup.com/sysbio/article-lookup/doi/10.2307/2412448>. Citado na página 23.

FLETCHER, W.; YANG, Z. INDELible: A flexible simulator of biological sequence evolution. **Molecular Biology and Evolution**, v. 26, n. 8, p. 1879–1888, 8 2009. ISSN 07374038. Citado na página 31.

GINALSKI, K. et al. ORFeus: detection of distant homology using sequence profiles and predicted secondary structure. **Nucleic Acids Research**, Oxford Academic, v. 31, n. 13, p. 3804–3807, 7 2003. ISSN 0305-1048. Disponível em: <https://academic.oup.com/nar/article/31/13/3804/2904126>. Citado na página 23.

GREENER, J. G. et al. A guide to machine learning for biologists. **Nature Reviews Molecular Cell Biology**, v. 23, n. 1, p. 40–55, 1 2022. ISSN 1471-0072. Disponível em: <https://www.nature.com/articles/s41580-021-00407-0>. Citado na página 26.

GUTHERZ, G. et al. Translating Akkadian to English with neural machine translation. **PNAS Nexus**, Oxford Academic, v. 2, n. 5, 5 2023. Disponível em: <https://dx.doi.org/10.1093/pnasnexus/pgad096>. Citado na página 96.

HAEUSLER, E. H. et al. Intentional Semantics for Molecular Biology. **Advances in Bioinformatics and Computational Biology. BSB 2023**, Springer, Cham, v. 13954, p. 94–105, 2023. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-031-42715-2_9>. Citado na página 97.

HAN, H.; LIU, X. The challenges of explainable AI in biomedical data science. **BMC Bioinformatics**, BioMed Central Ltd, v. 22, n. 12, p. 1–3, 1 2022. ISSN 14712105. Disponível em: <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-021-04368-1>. Citado na página 28.

HOLM, L. Dali server: structural unification of protein families. **Nucleic Acids Research**, Oxford Academic, v. 50, n. W1, p. W210–W215, 7 2022. ISSN 0305-1048. Disponível em: <https://dx.doi.org/10.1093/nar/gkac387>. Citado na página 25.

HUANG, Y. et al. JXP4BIGI: a generalized, Java XML-based approach for biological information gathering and integration. **Bioinformatics**, Oxford Academic, v. 19, n. 18, p. 2351–2358, 12 2003. ISSN 1367-4803. Disponível em: <https://dx.doi.org/10.1093/bioinformatics/btg327>. Citado na página 27.

HUERTA-CEPAS, J.; SERRA, F.; BORK, P. ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data. **Molecular Biology and Evolution**, v. 33, n. 6, p. 1635–1638, 2016. ISSN 15371719. Citado na página 46.

HULSEN, T. Explainable Artificial Intelligence (XAI): Concepts and Challenges in Healthcare. **AI**, Preprints, v. 4, n. 3, p. 652–666, 8 2023. ISSN 2673-2688. Disponível em: <https://www.mdpi.com/2673-2688/4/3/34>. Citado na página 29.

IBRAGIMOV, R. et al. GEDEVO: An Evolutionary Graph Edit Distance Algorithm for Biological Network Alignment. **German Conference on Bioinformatics**, v. 34, n. OpenAccess Series in Informatics (OASIcs), p. 68–79, 2013. ISSN 2190-6807. Citado na página 27.

IDREES, M.; KHAN, M. U. G. Conceptual Data Models for Biological Domain. **The Journal of Animal & Plant Sciences**, v. 25, n. 2, p. 337–345, 2015. ISSN 1018-7081. Citado na página 30.

IONESCU, R. T. Local rank distance. **Proceedings - 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2013**, IEEE Computer Society, p. 219–226, 2013. Citado na página 27.

JEFFREY, H. J. Chaos game representation of gene structure. **Nucleic Acids Research**, Oxford Academic, v. 18, n. 8, p. 2163–2170, 4 1990. ISSN 0305-1048. Disponível em: <https://academic.oup.com/nar/article/18/8/2163/2383530>. Citado na página 27.

JESCHKE, G. DEER Distance Measurements on Proteins. **https://doi.org/10.1146/annurev-physchem-032511-143716**, Annual Reviews, v. 63, p. 419–446, 4 2012. ISSN 0066426X. Disponível em: <https://www.annualreviews.org/doi/abs/10.1146/annurev-physchem-032511-143716>. Citado na página 27.

JIA, K. et al. New amino acid substitution matrix brings sequence alignments into agreement with structure matches. **Proteins: Structure, Function, and Bioinformatics**, John Wiley & Sons, Ltd, v. 89, n. 6, p. 671–682, 6 2021.

ISSN 1097-0134. Disponível em: <https://onlinelibrary.wiley.com/doi/full/10.1002/prot.26050>. Citado na página 54.

JONES, D. T.; THORNTON, J. M. The impact of AlphaFold2 one year on. **Nature Methods**, Nature Publishing Group, v. 19, n. 1, p. 15–20, 1 2022. ISSN 1548-7105. Disponível em: <https://www.nature.com/articles/s41592-021-01365-3>. Citado na página 18.

JUKES, T. H.; CANTOR, C. R. Evolution of Protein Molecules. **Mammalian Protein Metabolism**, Elsevier, p. 21–132, 1969. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/B9781483232119500097>. Citado na página 32.

JUMPER, J. et al. Highly accurate protein structure prediction with AlphaFold. **Nature**, Nature Publishing Group, v. 596, n. 7873, p. 583–589, 8 2021. ISSN 1476-4687. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/34265844>. Citado na página 18.

JUNYAN, Z.; CHENHUI, Y. Sequence Pattern Mining Based on Markov Chain. In: **7th International Conference on Information Technology in Medicine and Education (ITME)**. IEEE, 2015. p. 234–238. ISBN 978-1-4673-8302-8. Disponível em: <http://ieeexplore.ieee.org/document/7429136/>. Citado na página 27.

KAN, A. Machine learning applications in cell image analysis. **Immunology and Cell Biology**, John Wiley & Sons, Ltd, v. 95, n. 6, p. 525–530, 7 2017. ISSN 1440-1711. Disponível em: <https://onlinelibrary.wiley.com/doi/full/10.1038/icb.2017.16>. Citado na página 39.

KARAMICHALIS, R. et al. An investigation into inter- and intragenomic variations of graphic genomic signatures. **BMC Bioinformatics**, BioMed Central Ltd., v. 16, n. 1, p. 1–22, 8 2015. ISSN 14712105. Disponível em: <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-015-0655-4>. Citado 3 vezes nas páginas 28, 43 e 61.

KARIM, M. R. et al. Explainable AI for Bioinformatics: Methods, Tools, and Applications. **arXiv**, p. arXiv:2212.13261, 12 2022. Disponível em: <https://ui.adsabs.harvard.edu/abs/2022arXiv221213261R/abstract>. Citado na página 28.

KARP, P. D.; LEE, T. J.; WAGNER, V. BioWarehouse: Relational integration of eleven bioinformatics databases and formats. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, Springer, Berlin, Heidelberg, v. 5109 LNBI, p. 5–7, 2008. ISSN 03029743. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-540-69828-9_2>. Citado na página 27.

KASPRZYK, A. BioMart: driving a paradigm change in biological data management. **Database**, Oxford Academic, v. 2011, 1 2011. ISSN 17580463. Disponível em: <https://dx.doi.org/10.1093/database/bar049>. Citado na página 18.

KATOH, K. et al. MAFFT: A novel method for rapid multiple sequence alignment based on fast Fourier transform. **Nucleic Acids Research**, Oxford University Press, v. 30, n. 14, p. 3059–3066, 7 2002. ISSN 03051048. Citado na página 25.

KATOH, K.; ROZEWICKI, J.; YAMADA, K. D. MAFFT online service: multiple sequence alignment, interactive sequence choice and visualization. **Briefings in Bioinformatics**, Oxford Academic, v. 20, n. 4, p. 1160–1166, 7 2019. ISSN 1467-5463. Disponível em: <https://dx.doi.org/10.1093/bib/bbx108>. Citado na página 25.

KEMPEN, M. v. et al. Fast and accurate protein structure search with Foldseek. **bioRxiv**, Cold Spring Harbor Laboratory, p. 2022.02.07.479398, 3 2023. Disponível em: <https://www.biorxiv.org/content/10.1101/2022.02.07.479398v5>. Citado 2 vezes nas páginas 23 e 25.

KIELA, D.; BOTTOU, L. Learning Image Embeddings using Convolutional Neural Networks for Improved Multi-Modal Semantics. In: **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2014. p. 36–45. Disponível em: <http://aclweb.org/anthology/D14-1005>. Citado na página 42.

KLUSKA, K. et al. Non-Conserved Amino Acid Residues Modulate the Thermodynamics of Zn(II) Binding to Classical $\beta\beta\alpha$ Zinc Finger Domains. **International Journal of Molecular Sciences**, MDPI, v. 23, n. 23, p. 14602, 12 2022. ISSN 14220067. Disponível em: <https://www.mdpi.com/1422-0067/23/23/14602>. Citado na página 20.

LADUNGA, I. Finding Homologs in Amino Acid Sequences Using Network BLAST Searches. **Current protocols in bioinformatics**, John Wiley & Sons, Ltd, v. 59, n. 1, p. 1–3, 9 2017. ISSN 1934-340X. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/28902395>. Citado na página 23.

LANGMEAD, C. J.; DONALD, B. R. High-throughput 3D structural homology detection via NMR resonance assignment. **Proceedings. IEEE Computational Systems Bioinformatics Conference**, p. 278–89, 2004. ISSN 1551-7497. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/16448021>. Citado na página 23.

LATIF, J. et al. Medical Imaging using Machine Learning and Deep Learning Algorithms: A Review. In: **2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)**. IEEE, 2019. p. 1–5. ISBN 978-1-5386-9509-8. Disponível em: <https://ieeexplore.ieee.org/document/8673502/>. Citado na página 39.

LEONARD, S. A.; LITTLEJOHN, T. G.; BAXEVANIS, A. D. Common File Formats. **Current Protocols in Bioinformatics**, John Wiley & Sons, Ltd, v. 16, n. 1, p. A.1B.1–A.1B.9, 12 2006. ISSN 1934-340X. Disponível em: <https://currentprotocols.onlinelibrary.wiley.com/doi/10.1002/0471250953.bia01bs16>. Citado na página 17.

LI, Q. et al. The non-conserved region of MRP is involved in the virulence of Streptococcus suis serotype 2. **Virulence**, Taylor and Francis Inc., v. 8, n. 7, p. 1274–1289, 10 2017. ISSN 21505608. Disponível em: <https://www.tandfonline.com/doi/abs/10.1080/21505594.2017.1313373>. Citado na página 20.

LIFSCHITZ, S. et al. Bio-Strings: A Relational Database Data-Type for Dealing with Large Biosequences. **BioTech**, Multidisciplinary Digital Publishing Institute, v. 11, n. 3, p. 31, 7 2022. ISSN 2673-6284. Disponível em: <https://www.mdpi.com/2673-6284/11/3/31>. Citado na página 30.

LÖCHEL, H. F.; HEIDER, D. Chaos game representation and its applications in bioinformatics. **Computational and Structural Biotechnology Journal**, Research Network of Computational and Structural Biotechnology, v. 19, p. 6263, 1 2021. ISSN 20010370. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8636998/>. Citado na página 27.

LORENZON, L. N. Análise Comparada entre Regulamentações de Dados Pessoais no Brasil e na União Européia (LGPD e GDPR) e seus Respectivos Instrumentos de Enforcement. **Revista do Programa de Direito da União Europeia**, v. 1, p. 39–52, 3 2021. Disponível em: <https://hml-bibliotecadigital.fgv.br/ojs/index.php/rpdue/article/view/83423>. Citado na página 29.

MACEDO, J. A. F. D. et al. A conceptual data model language for the molecular biology domain. **Proceedings - IEEE Symposium on Computer-Based Medical Systems**, p. 231–236, 2007. ISSN 10637125. Citado na página 30.

MARCINKEVIČS, R.; VOGT, J. E. Interpretability and Explainability: A Machine Learning Zoo Mini-tour. **Arxiv**, 12 2020. Disponível em: <http://arxiv.org/abs/2012.01805>. Citado na página 28.

MCGINNIS, S.; MADDEN, T. L. BLAST: at the core of a powerful and diverse set of sequence analysis tools. **Nucleic Acids Research**, Oxford University Press, v. 32, n. Web Server issue, p. W20, 7 2004. ISSN 03051048. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC441573/>. Citado na página 25.

MILLS, L. Common File Formats. **Current Protocols in Bioinformatics**, John Wiley and Sons Inc., v. 45, n. 1, 3 2014. ISSN 1934-3396. Disponível em: <https://onlinelibrary.wiley.com/doi/10.1002/0471250953.bia01bs45>. Citado na página 17.

NAG, D. K. et al. Both conserved and non-conserved regions of Spo11 are essential for meiotic recombination initiation in yeast. **Molecular Genetics and Genomics**, Springer, v. 276, n. 4, p. 313–321, 10 2006. ISSN 16174615. Disponível em: <https://link.springer.com/article/10.1007/s00438-006-0143-7>. Citado na página 20.

NALDI, A. et al. Cooperative development of logical modelling standards and tools with CoLoMoTo. **Bioinformatics**, Oxford Academic, v. 31, n. 7, p. 1154–1159, 4 2015. ISSN 1367-4803. Disponível em: <https://dx.doi.org/10.1093/bioinformatics/btv013>. Citado na página 29.

NEEDLEMAN, S. B.; WUNSCH, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. **Journal of Molecular Biology**, Academic Press, v. 48, n. 3, p. 443–453, 3 1970. ISSN 0022-2836. Citado na página 32.

NILSSON, J.; NVIDIA, T. A.-M. Understanding SSIM. 6 2020. Disponível em: <https://arxiv.org/abs/2006.13846v2>. Citado na página 28.

NOVÈRE, N. L. et al. BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. **Nucleic acids research**, Oxford University Press, v. 34, n. Database issue, p. 689–91, 1 2006. ISSN 1362-4962. Disponível em: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC1347454>. Citado na página 18.

O'SULLIVAN, O. et al. 3DCoffee: Combining Protein Sequences and Structures within Multiple Sequence Alignments. **Journal of Molecular Biology**, Academic Press, v. 340, n. 2, p. 385–395, 7 2004. ISSN 0022-2836. Citado na página 25.

PORTIK, D. M.; WIENS, J. J. Do Alignment and Trimming Methods Matter for Phylogenomic (UCE) Analyses? **Systematic Biology**, Oxford Academic, v. 70, n. 3, p. 440–462, 4 2021. ISSN 1063-5157. Disponível em: <https://dx.doi.org/10.1093/sysbio/syaa064>. Citado na página 25.

RANWEZ, V.; CHANTRET, N. N. Strengths and Limits of Multiple Sequence Alignment and Filtering Methods. In: SCORNAVACCA, C.; DELSUC, F.; GALTIER, N. (Ed.). **Phylogenetics in the Genomic Era**. No commercial publisher | Authors open access book, 2020. v2, cap. 2.2, p. 1–36. Disponível em: <https://hal.inria.fr/PGE.> Citado na página 19.

REN, J. et al. Alignment-Free Sequence Analysis and Applications. **Annual review of biomedical data science**, NIH Public Access, v. 1, n. 1, p. 93, 7 2018. ISSN 2574-3414. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6905628/>. Citado 2 vezes nas páginas 20 e 25.

ROBINSON, D. F.; FOULDS, L. R. Comparison of phylogenetic trees. **Mathematical Biosciences**, Elsevier, v. 53, n. 1-2, p. 131–147, 2 1981. ISSN 0025-5564. Citado na página 45.

RÖHLING, S. et al. The number of k-mer matches between two DNA sequences as a function of k and applications to estimate phylogenetic distances. **PLoS ONE**, PLOS, v. 15, n. 2, 2 2020. ISSN 19326203. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7010260/>. Citado na página 27.

ROZEWICKI, J. et al. MAFFT-DASH: integrated protein sequence and structural alignment. **Nucleic Acids Research**, Oxford Academic, v. 47, n. W1, p. W5–W10, 7 2019. ISSN 0305-1048. Disponível em: <https://dx.doi.org/10.1093/nar/gkz342>. Citado na página 25.

SAIFULLAH, S. et al. Privacy Meets Explainability: A Comprehensive Impact Benchmark. **Arxiv**, 11 2022. Disponível em: <http://arxiv.org/abs/2211.04110>. Citado na página 29.

SAITOU, N.; NEI, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. **Molecular Biology and Evolution**, Oxford Academic, v. 4, n. 4, p. 406–425, 7 1987. ISSN 0737-4038. Disponível em: <https://academic.oup.com/mbe/article/4/4/406/1029664>. Citado na página 47.

SATRA, S. et al. Octopus: A Novel Approach for Health Data Masking and Retrieving Using Physical Unclonable Functions and Machine Learning. **Sensors 2023, Vol. 23, Page 4082**, Multidisciplinary Digital Publishing Institute, v. 23, n. 8, p. 4082, 4 2023. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/23/8/4082>. Citado na página 29.

SCHÖNBACH, C.; KOWALSKI-SAUNDERS, P.; BRUSIC, V. Data warehousing in molecular biology. **Briefings in Bioinformatics**, Oxford Academic, v. 1, n. 2, p. 190–198, 5 2000. ISSN 1467-5463. Disponível em: <https://dx.doi.org/10.1093/bib/1.2.190>. Citado na página 27.

SEIBT, K. M.; SCHMIDT, T.; HEITKAM, T. FlexiDot: Highly customizable, ambiguity-aware dotplots for visual sequence analyses. **Bioinformatics**, Oxford University Press, v. 34, n. 20, p. 3575–3577, 10 2018. Citado na página 39.

SIDDARTHA, B. K.; RAVIKUMAR, G. K. A Novel Data Masking Method for Securing Medical Image. In: **International Conference on Smart Systems and Inventive Technology (ICSSIT)**. IEEE, 2019. p. 30–34. ISBN 978-1-7281-2119-2. Disponível em: <https://ieeexplore.ieee.org/document/8987835/>. Citado na página 29.

SIDDARTHA, B. K.; RAVIKUMAR, G. K. An efficient data masking for securing medical data using DNA encoding and chaotic system. **International Journal of Electrical and Computer Engineering (IJECE)**, v. 10, n. 6, p. 6008, 12 2020. ISSN 2722-2578. Disponível em: <http://ijece.iaescore.com/index.php/IJECE/article/view/20515>. Citado na página 29.

SIEVERS, F.; HIGGINS, D. G. Clustal Omega for making accurate alignments of many protein sequences. **Protein Science : A Publication of the Protein Society**, Wiley-Blackwell, v. 27, n. 1, p. 135, 1 2018. ISSN 1469896X. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5734385/>. Citado na página 35.

SIEVERS, F.; HIGGINS, D. G. The Clustal Omega Multiple Alignment Package. **Methods in Molecular Biology**, Humana Press Inc., v. 2231, p. 3–16, 2021. ISSN 19406029. Disponível em: <https://link.springer.com/protocol/10.1007/978-1-0716-1036-7_1>. Citado na página 23.

SIEVERS, F. et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. **Molecular Systems Biology**, John Wiley & Sons, Ltd, v. 7, n. 1, p. 539, 1 2011. ISSN 1744-4292. Disponível em: <https://onlinelibrary.wiley.com/doi/full/10.1038/msb.2011.75https://onlinelibrary.wiley.com/doi/abs/10.1038/msb.2011.75https://www.embopress.org/doi/10.1038/msb.2011.75>. Citado na página 23.

SIMONYAN, K.; ZISSERMAN, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. **3rd International Conference on Learning Representations, ICLR - Conference Track Proceedings**, International Conference on Learning Representations, ICLR, 9 2014. Disponível em: <http://arxiv.org/abs/1409.1556>. Citado na página 42.

SMITH, T. F.; WATERMAN, M. S. Identification of common molecular subsequences. **Journal of Molecular Biology**, Academic Press, v. 147, n. 1, p. 195–197, 3 1981. ISSN 0022-2836. Citado na página 46.

SNEATH, P. H. Relations between chemical structure and biological activity in peptides. **Journal of Theoretical Biology**, Academic Press, v. 12, n. 2, p. 157–195, 11 1966. ISSN 0022-5193. Citado na página 56.

SÖDING, J. Protein homology detection by HMM–HMM comparison. **Bioinformatics**, Oxford Academic, v. 21, n. 7, p. 951–960, 4 2005. ISSN 1367-4803. Disponível em: <https://dx.doi.org/10.1093/bioinformatics/bti125>. Citado 2 vezes nas páginas 23 e 24.

SOMEKH, J.; CHODER, M.; DORI, D. Conceptual Model-Based Systems Biology: Mapping Knowledge and Discovering Gaps in the mRNA Transcription Cycle. **PLOS ONE**, Public Library of Science, v. 7, n. 12, p. e51430, 12 2012. ISSN 1932-6203. Disponível em: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0051430>. Citado na página 29.

STEENWYK, J. L. et al. ClipKIT: A multiple sequence alignment trimming software for accurate phylogenomic inference. **PLOS Biology**, Public Library of Science, v. 18, n. 12, p. e3001007, 12 2020. ISSN 1545-7885. Disponível em: <https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.3001007>. Citado na página 25.

STYCZYNSKI, M. P. et al. BLOSUM62 miscalculations improve search performance. **Nature biotechnology**, Nature Publishing Group, v. 26, n. 3, p. 274–5, 3 2008. ISSN 1546-1696. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/18327232>. Citado na página 54.

SUN, Z. et al. A novel numerical representation for proteins: Three-dimensional Chaos Game Representation and its Extended Natural Vector. **Computational and Structural Biotechnology Journal**, Elsevier, v. 18, p. 1904–1913, 1 2020. ISSN 2001-0370. Citado na página 28.

TIAN, D. P. A Review on Image Feature Extraction and Representation Techniques. **International Journal of Multimedia and Ubiquitous Engineering**, v. 8, n. 4, p. 385–396, 2013. Citado na página 41.

TIWARI, D. D. et al. BioModelsML: Building a FAIR and reproducible collection of machine learning models in life sciences and medicine for easy reuse. **bioRxiv**, Cold Spring Harbor Laboratory, p. 2023.05.22.540599, 5 2023. Disponível em: <https://www.biorxiv.org/content/10.1101/2023.05.22.540599v1>. Citado na página 18.

TRIVEDI, R.; NAGARAJARAM, H. A. Substitution scoring matrices for proteins - An overview. **Protein Science**, John Wiley & Sons, Ltd, v. 29, n. 11, p. 2150–2163, 11 2020. ISSN 1469-896X. Disponível em: <https://onlinelibrary.wiley.com/doi/full/10.1002/pro.3954>. Citado na página 54.

VERLEYSEN, M.; FRANÇOIS, D. The curse of dimensionality in data mining and time series prediction. **Lecture Notes in Computer Science**, Springer

Verlag, v. 3512, p. 758–770, 2005. ISSN 03029743. Disponível em: <https://link.springer.com/chapter/10.1007/11494669_93>. Citado na página 96.

VIEIRA, D. M. et al. Driftage: a multi-agent system framework for concept drift detection. **GigaScience**, Oxford Academic, v. 10, n. 6, p. 1–10, 6 2021. ISSN 2047217X. Disponível em: <https://dx.doi.org/10.1093/gigascience/giab030>. Citado na página 97.

VIEIRA, D. M. et al. Scientific Workflow Interactions: An Application to Cancer Gene Identification. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, Springer Science and Business Media Deutschland GmbH, v. 13523 LNBI, p. 14–19, 2022. ISSN 16113349. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-031-21175-1_2>. Citado na página 97.

VINGA, S.; ALMEIDA, J. Alignment-free sequence comparison—a review. **Bioinformatics**, Oxford Academic, v. 19, n. 4, p. 513–523, 3 2003. ISSN 1367-4803. Disponível em: <https://academic.oup.com/bioinformatics/article/19/4/513/218529>. Citado na página 20.

WÄLDCHEN, J.; MÄDER, P. Machine learning for image based species identification. **Methods in Ecology and Evolution**, John Wiley & Sons, Ltd, v. 9, n. 11, p. 2216–2225, 11 2018. ISSN 2041-210X. Disponível em: <https://onlinelibrary.wiley.com/doi/full/10.1111/2041-210X.13075>. Citado na página 39.

WANG, L.; JIANG, T. On the complexity of multiple sequence alignment. **Journal of computational biology : a journal of computational molecular cell biology**, J Comput Biol, v. 1, n. 4, p. 337–348, 1994. ISSN 1066-5277. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/8790475/>. Citado na página 23.

WANG, Z.; BOVIK, A. C. A universal image quality index. **IEEE Signal Processing Letters**, v. 9, n. 3, p. 81–84, 2002. Citado na página 43.

WANG, Z. et al. Image quality assessment: From error visibility to structural similarity. **IEEE Transactions on Image Processing**, v. 13, n. 4, p. 600–612, 4 2004. ISSN 10577149. Citado 2 vezes nas páginas 43 e 78.

WANG, Z.; SIMONCELLI, E.; BOVIK, A. Multiscale structural similarity for image quality assessment. In: **The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers**. IEEE, 2003. v. 2, p. 1398–1402. ISBN 0-7803-8104-1. ISSN 10586393. Disponível em: <http://ieeexplore.ieee.org/document/1292216/>. Citado na página 43.

WOLFF, J. Approximate nearest neighbor query methods for large scale structured datasets. **Albert Ludwig University of Freiburg Germany**, 2016. Citado 3 vezes nas páginas 9, 43 e 59.

WOLSTENCROFT, K. et al. SEEK: A systems biology data and model management platform. **BMC Systems Biology**, BioMed Central Ltd., v. 9, n. 1, p. 1–12, 7 2015. ISSN 17520509. Disponível em: <https://bmcsystbiol.biomedcentral.com/articles/10.1186/s12918-015-0174-y>. Citado na página 18.

WOOLEY, J. C.; LIN, H. S. On the Nature of Biological Data. In: **Catalyzing Inquiry at the Interface of Computing and Biology.** Washington, D.C.: National Academies Press, 2005. cap. 3. ISBN 978-0-309-09612-6. Disponível em: <http://www.nap.edu/catalog/11480>. Citado na página 17.

XU, C.; JACKSON, S. A. Machine learning and complex biological data. **Genome Biology**, BioMed Central Ltd., v. 20, n. 1, p. 1–4, 4 2019. ISSN 1474760X. Disponível em: <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1689-0>. Citado na página 18.

YANG, A. et al. Review on the Application of Machine Learning Algorithms in the Sequence Data Mining of DNA. **Frontiers in Bioengineering and Biotechnology**, Frontiers Media SA, v. 8, p. 1032, 9 2020. ISSN 22964185. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7498545/>. Citado na página 26.

ZHANG, D. et al. DNAGPT: A Generalized Pre-trained Tool for Versatile DNA Sequence Analysis Tasks. **Arxiv**, 7 2023. Disponível em: <http://arxiv.org/abs/2307.05628>. Citado na página 19.

ZIELEZINSKI, A. et al. Benchmarking of alignment-free sequence comparison methods. **Genome Biology**, BioMed Central Ltd., v. 20, n. 1, p. 1–18, 7 2019. ISSN 1474760X. Disponível em: <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1755-7>. Citado na página 37.

ZIELEZINSKI, A. et al. Alignment-free sequence comparison: benefits, applications, and tools. **Genome biology**, Genome Biol, v. 18, n. 1, 10 2017. ISSN 1474-760X. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/28974235/>. Citado na página 20.