UNIVERSIDADE DO ESTADO DA BAHIA (UNEB)
DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA, CAMPUS I
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

RICKSON ALVES DA SILVA

ESTUDO DE VIABILIDADE DO ALGORITMO CODON BASED UNSUPERVISED CLASSIFICATION (CBUC)

SALVADOR
2024

RICKSON ALVES DA SILVA

ESTUDO DE VIABILIDADE DO ALGORITMO CODON BASED UNSUPERVISED CLASSIFICATION (CBUC)

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação do Departamento de Ciências Exatas e da Terra (DCET) - Campus I, da Universidade do Estado da Bahia (UNEB), como requisito à obtenção do grau de bacharel em Sistemas de Informação.
Área de concentração: Ciências da Computação.

Orientador: Prof. Dra. Maria Inês Valderrama Restovic

SALVADOR
2024

TERMO DE ANUÊNCIA DO ORIENTADOR

Declaro para os devidos fins que li e revisei este trabalho e atesto sua qualidade como resultado final desta monografia. Confirmo que o referencial teórico apresentado é completo e suficiente para fundamentar os objetivos propostos e que a metodologia científica utilizada e os resultados finais são consistentes e com qualidade suficiente para submissão à banca examinadora final do Trabalho de Conclusão de Curso II do curso de Bacharelado em Sistemas de Informação.

Prof. Dra. Maria Inês Valderrama Restovic
Orientador

RICKSON ALVES DA SILVA

ESTUDO DE VIABILIDADE DO ALGORITMO CODON BASED UNSUPERVISED CLASSIFICATION (CBUC)

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação do Departamento de Ciências Exatas e da Terra (DCET) - Campus I, da Universidade do Estado da Bahia (UNEB), como requisito à obtenção do grau de bacharel em Sistemas de Informação.
Área de concentração: Ciências da Computação.

Aprovada em: 09/07/2024

BANCA EXAMINADORA

Prof. Dra. Maria Inês Valderrama Restovic
Orientadora

Prof. Dr. Diego Gervásio Frias Suarez
(DCET-I/UNEB)

Prof. Dr. Vagner de Souza Fonseca
(DCET-I/UNEB)

AGRADECIMENTOS

É com profunda gratidão que dedico esta seção de agradecimentos do meu Trabalho de Conclusão de Curso. Esta jornada acadêmica não teria sido possível sem o apoio e incentivo de pessoas incríveis que estiveram ao meu lado.

Agradeço primeiramente à minha família, por ser a minha base, pelo amor incondicional, e por sempre acreditarem no meu potencial. Vocês foram a força que impulsionou cada passo desta caminhada.

Ao meu orientador, agradeço pela paciência, sabedoria e orientação ao longo de todo o processo de pesquisa. Seu apoio foi fundamental para o desenvolvimento deste trabalho.

Aos professores que contribuíram com seus conhecimentos e experiências, a minha sincera gratidão. Cada aula, conselho e feedback foram fundamentais para o meu crescimento acadêmico.

Aos amigos e colegas de curso, pela troca de ideias, momentos de estudo e pelo apoio mútuo, meu muito obrigado. Compartilhamos desafios e conquistas que tornaram esta jornada ainda mais significativa.

Por fim, agradeço a todos que, de alguma forma, contribuíram para o sucesso deste trabalho. Cada palavra de encorajamento, cada gesto de apoio, foi fundamental para chegar até este momento.

Este TCC não é apenas um marco acadêmico, mas uma realização coletiva. A todos vocês, o meu mais sincero agradecimento.

"A educação é a arma mais poderosa que você pode usar para mudar o mundo."

(Nelson Mandela)

RESUMO

Este trabalho validou o algoritmo CBUC para identificação de sequências de arbovírus, com ênfase no vírus Zika. Desenvolvido pelo Dr. Diego Gervásio Frias Suaréz, o CBUC utiliza aprendizado de máquina baseado no algoritmo PSRM para detectar padrões e agrupamentos genéticos. Utilizou-se um dataset do ABVdb contendo sequências genotipadas de Zika para agrupar genótipos de arbovírus e identificar novas sequências do Zika. Foram desenvolvidas duas interfaces, uma desktop e outra web, para uso do CBUC. Os resultados mostraram que o CBUC identificou 100% das sequências completas de até 8000 bases no dataset de teste, sugerindo sua eficácia na identificação de sequências completas do vírus Zika quando de tamanho similar às utilizadas no treinamento. Para avaliação da precisão, os resultados do CBUC foram comparados com os do *Genome Detective*, uma ferramenta que emprega métodos tradicionais. A comparação demonstrou que o CBUC apresenta resultados promissores e competitivos na identificação precisa de genótipos do Zika

Palavras-chave: Bioinformatica; Arbovírus; Genotipagem; Filogenia.

ABSTRACT

This study validated the CBUC algorithm for arbovirus sequence identification, focusing on the Zika virus. Developed by Dr. Diego Gervásio Frias Suaréz, CBUC employs machine learning based on the PSRM algorithm to detect genetic patterns and clusters. A dataset from ABVdb containing Zika genotyped sequences was used to cluster arbovirus genotypes and identify new Zika sequences. Two interfaces, a desktop and a web version, were developed for CBUC usage. Results demonstrated that CBUC identified 100% of complete sequences up to 8000 bases in the test dataset, suggesting its effectiveness in identifying complete Zika virus sequences of similar length to those used in training. To assess its accuracy, CBUC results were compared with those from Genome Detective, a tool utilizing traditional methods. The comparison showed that CBUC delivers promising and competitive outcomes in precise Zika genotype identification.

Keywords: Bioinformatics; Arboviruses; Genotyping; Phylogeny.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

CBUC	*Codon Based Usupervised Classification*
DNA	Ácido Desoxirribonucleico
RNA	Ácido Ribonucleico
A	Adenina
G	Guanina
T	Timina
C	Citosina
U	Uracila
ABVdb	Arthropod Borne Virus database
PSRM	Parametric State Recognition Method
DENV	Vírus Dengue
CHIKV	Vírus Chikungunya
ZIKV	Vírus Zika

SUMÁRIO

1 INTRODUÇÃO

A genotipagem desempenha um papel crucial na identificação de variantes genéticas associadas a doenças e na compreensão das variações genéticas que podem causar mutações. Essas variantes podem influenciar a suscetibilidade a doenças, a resposta a medicamentos e até mesmo a progressão de certas condições médicas. Através da análise de marcadores genéticos específicos, a genotipagem permite a identificação de alelos e genótipos individuais, fornecendo informações valiosas sobre a composição genética de indivíduos e populações.

Além disso, a bioinformática desempenha um papel importante no processamento e análise dos dados gerados pela genotipagem. Métodos computacionais e algoritmos são aplicados para mapear sequências de DNA ou RNA, identificar variações genéticas e correlacionar essas informações com fenótipos observados (Kockum; Huang; Stridh, 2023).

Dentro da bioinformática, utilizam-se árvores filogenéticas para agrupar linhagens genômicas, essenciais para investigar a origem e a disseminação de vírus. No entanto, as árvores filogenéticas não podem ser observadas diretamente, sendo inferidas a partir dos dados disponíveis. A análise filogenética permite entender as relações evolutivas entre diferentes linhagens virais, incluindo arbovírus. Métodos como a máxima parcimônia, a máxima verossimilhança e a inferência bayesiana são aplicados para analisar sequências genômicas de arbovírus, permitindo não apenas inferir relações filogenéticas precisas, mas também estimar datas de divergência entre linhagens, identificar eventos de recombinação genética e traçar rotas de disseminação geográfica desses agentes patogênicos (Yang; Rannala, 2012).

Para o contexto deste trabalho de conclusão de curso (TCC), é crucial compreender o conceito de arbovírus. Segundo (Lopes; Nozawa; Linhares, 2014), arbovírus são vírus transmitidos por artrópodes e se replicam parcialmente nos insetos. Transmitidos aos seres humanos e outros animais pela picada de artrópodes hematófagos, os arbovírus pertencem a cinco famílias virais principais: *Bunyaviridae, Togaviridae, Flaviviridae, Reoviridae e Rhabdoviridae*.

Dentro desse contexto, o algoritmo desenvolvido pelo professor Dr. Diego Gervasio Frías Suárez, chamado *CBUC*, que é baseado em PSRM, é uma abordagem baseada em aprendizado de máquina e pretende auxiliar na genotipagem de sequências genômicas. Esse algoritmo possui a capacidade de identificar padrões em dados genômicos e realizar o agrupamento dessas sequências em famílias, de forma similar às árvores filogenéticas. O *CBUC* busca classificar as sequências de forma rápida e exigindo menos poder computacional. O *CBUC* foi utilizado apenas em um Trabalho de Conclusão de Curso (TCC) que analisou a proteína spike do vírus Sars-Cov-2 (Nascimento, 2021).

A pergunta norteadora deste trabalho é: É possível utilizar os agrupamentos (famílias) do *CBUC* para fazer a classificação de sequências genéticas do vírus Zika com a mesma precisão dos métodos tradicionais?

O objetivo é avaliar a precisão do algoritmo *CBUC* na genotipagem de sequências de arbovírus, com foco no vírus Zika, e compará-la com os métodos tradicionais. Para isso, são definidos os seguintes objetivos específicos:

- Treinamento do *CBUC*: Obtenção de sequências completas já genotipadas do vírus Zika do ABVdb;
- Adaptação do *CBUC*: Realização de alterações necessárias no algoritmo;
- Desenvolvimento de Interfaces: Criação de interfaces amigáveis para o uso do *CBUC*;
- Coleta de Dados: Coleta de sequências genômicas do vírus Zika do GenBank para testes;
- Análise Comparativa: Comparação e validação dos resultados do *CBUC* em relação aos métodos tradicionais de genotipagem.

A motivação deste trabalho baseia-se no potencial do algoritmo *CBUC* para a identificação de novas variantes de vírus, bem como em sua capacidade de oferecer uma alternativa de baixo custo em comparação com os métodos tradicionais de filogenia. A utilização do *CBUC* pode permitir uma detecção mais ágil e eficiente de variantes virais.

Ao validar o algoritmo *CBUC*, buscamos fornecer uma alternativa para a identificação de genótipos de arbovírus. Isso contribuirá para avançar nosso conhecimento sobre a epidemiologia e a evolução desses vírus, permitindo uma análise mais detalhada das relações filogenéticas e identificação de variantes virais de importância clínica. Além disso, a validação do *CBUC* como uma alternativa aos métodos tradicionais de genotipagem pode abrir portas para o desenvolvimento de abordagens mais avançadas no campo da bioinformática viral.

Nos próximos capítulos, haverá o desenvolvimento deste trabalho. No Capítulo 2, será apresentado o referencial teórico, abordando os conceitos e metodologias essenciais para a genotipagem, bioinformática e análise filogenética, incluindo uma revisão dos principais algoritmos de classificação de sequências genômicas, além de discutir o ABVdb e o *Genome Detective*. O Capítulo 3 detalhará o desenvolvimento do projeto, descrevendo as etapas de treinamento e adaptação do algoritmo *CBUC*, a criação das interfaces e a metodologia para a coleta de dados e análise comparativa dos resultados. No Capítulo 4, serão apresentados os resultados obtidos, comparando a eficácia do *CBUC* com os métodos tradicionais de genotipagem em termos de precisão, incluindo uma análise do

tempo de execução do algoritmo. Por fim, na conclusão, serão discutidas as principais descobertas e contribuições do trabalho, avaliando a validade e viabilidade do *CBUC* como uma alternativa aos métodos tradicionais e explorando as implicações e direções futuras para pesquisas adicionais na área de bioinformática e genotipagem de arbovírus.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, será apresentada a fundamentação teórica para este trabalho, que se concentra principalmente na área de bioinformática. A bioinformática é uma área multidisciplinar que engloba diversos campos de atuação. O foco principal do estudo são os arbovírus.

2.1 Biologia Molecular

Biologia molecular é um ramo que estuda os processos moleculares, com foco principalmente em *DNA*, *RNA* e síntese de proteínas.

Assim como os computadores utilizam a codificação binária representada pelos dígitos 0 e 1, o *DNA* possui unidades de informação chamadas nucleotídeos, sendo eles A, G, T, C, responsáveis por sua codificação (Alberts, 2017).

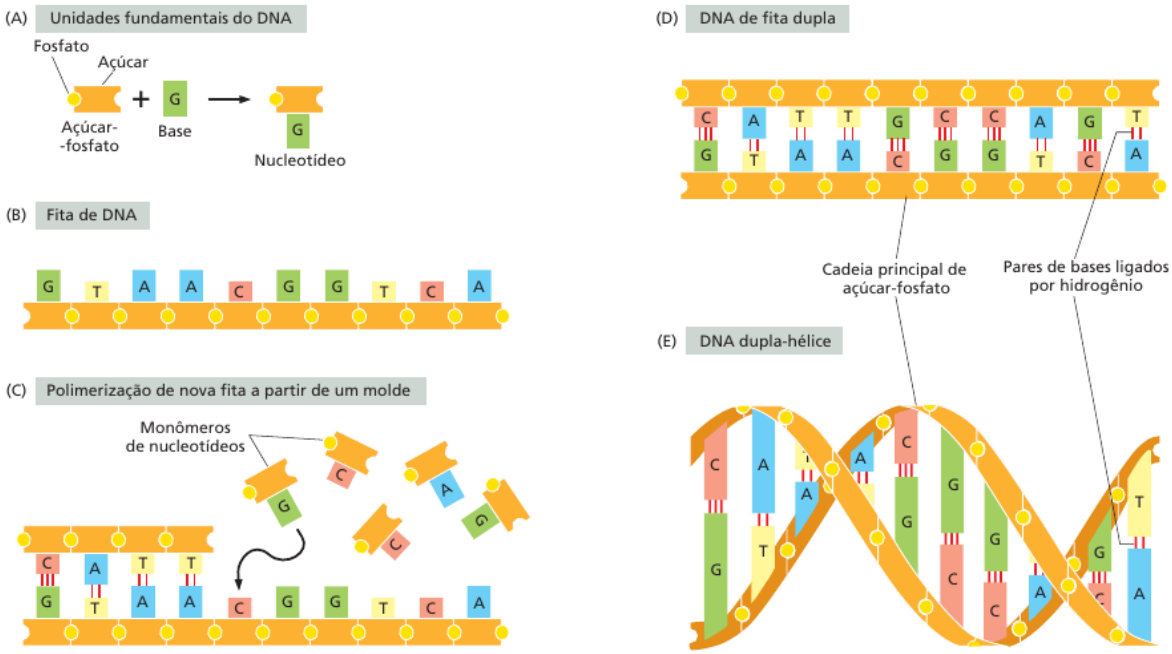


Figura 1 – Imagem extraída de (Alberts, 2017)

A biologia molecular busca entender como as informações genéticas são armazenadas, transmitidas e expressas nos organismos vivos. Isso inclui a investigação dos mecanismos moleculares envolvidos na replicação do *DNA*, transcrição do *DNA* para a síntese de *RNA* e tradução do *RNA* em proteínas. Além disso, a biologia molecular também estuda a regulação dos genes e a expressão gênica, ou seja, como os genes são ativados ou desativados em diferentes células e momentos. A biologia molecular é um ramo da biologia que se

concentra no estudo das estruturas e funções das moléculas biológicas que estão envolvidas nos processos celulares. Ela abrange uma variedade de tópicos relacionados à genética, expressão gênica, replicação do *DNA*, transcrição, tradução, regulação gênica e interações moleculares (Alberts, 2017).

2.2 Genotipagem

Genotipagem é o processo de determinar o genótipo de um indivíduo, ou seja, a composição genética de um organismo em termos de variantes genéticas específicas. A genotipagem pode ser usada para identificar variantes genéticas associadas a características ou doenças específicas, bem como para estudos genealógicos, pesquisas em animais e humanos, investigação forense e medicina personalizada. Existem várias técnicas de genotipagem disponíveis, sequenciamento de *DNA*, análise de fragmentos de restrição, entre outras. A genotipagem é uma ferramenta importante para a pesquisa genética com aplicações em uma ampla variedade de campos, incluindo medicina, biologia, antropologia e forense (Kockum; Huang; Stridh, 2023).

2.3 Bioinformática

Bioinformática é um campo multidisciplinar que combina biologia e ciência da computação para coletar, armazenar, analisar e interpretar dados biológicos. Envolve o desenvolvimento e a aplicação de métodos computacionais e estatísticos para entender melhor os processos biológicos e resolver problemas relacionados à vida.

Este campo lida com uma abundância de dados biológicos, como sequências de *DNA*, proteínas, estruturas tridimensionais, informações genéticas e expressão gênica. Esses dados são coletados por meio de técnicas experimentais, como sequenciamento de *DNA*, cristalografia de proteínas, microarranjos de *DNA* e outros métodos de biologia molecular.

As principais ferramentas utilizadas são *softwares* e a os softwares, sendo que muitos deles estão disponiveis publicamente na *internet*, que permitem o desenvolvimento de algoritmos e ferramentas computacionais para armazenar e analisar dados biológicos. Profissionais da área utilizam técnicas de programação, aprendizado de máquina, estatística e outras abordagens computacionais para extrair informações úteis dos dados biológicos. Essas informações são usadas para entender a estrutura e função de biomoléculas, estudar a evolução biológica, identificar genes relacionados a doenças, projetar novos medicamentos e realizar muitas outras aplicações na biologia.

O campo desempenha um papel fundamental na medicina personalizada, agricultura, biotecnologia e várias outras áreas da pesquisa biológica. Ele ajuda os cientistas a lidarem

com a crescente quantidade de dados gerados pelas tecnologias modernas e a obterem *insights* valiosos para avançar nosso conhecimento sobre a vida.

2.4 Arbovírus

Os arbovírus são um grupo de vírus transmitidos por artrópodes, como mosquitos e carrapatos. O termo "arbovírus"é uma abreviação de "arthropod-borne viruses". Esses vírus são encontrados em todo o mundo e podem causar uma variedade de doenças tanto em humanos quanto em animais. Alguns exemplos conhecidos de arbovírus incluem o vírus da dengue, o vírus Zika, o vírus Chikungunya, o vírus do Nilo Ocidental e o vírus da febre amarela. A transmissão dos arbovírus ocorre quando um artrópode vetor, como um mosquito infectado, se alimenta do sangue de um hospedeiro infectado e, em seguida, transmite o vírus para um novo hospedeiro durante a alimentação subsequente. Os arbovírus podem se replicar tanto no vetor como no hospedeiro vertebrado, podendo causar doenças em ambos. As doenças causadas por arbovírus variam em gravidade, desde infecções assintomáticas até doenças graves. Os sintomas comuns incluem febre, dor de cabeça, fadiga, dores articulares e musculares, erupções cutâneas e manifestações neurológicas. Em alguns casos, as infecções por arbovírus podem levar a complicações sérias, como encefalite (inflamação do cérebro) ou síndrome neurológica grave (Lima-Camara, 2016) (Lopes; Nozawa; Linhares, 2014).

2.4.1 Dengue

A dengue (Khan *et al.*, 2023) é causada por um vírus do gênero *Flavivirus*, da família *Flaviviridae*. O vírus da dengue possui um genoma de *RNA* de cadeia simples, que é encapsulado em uma cápside proteica. A cápside é cercada por uma membrana lipídica derivada da célula hospedeira e contém proteínas de envelope na superfície. O vírus da dengue possui quatro sorotipos principais, denominados DENV-1, DENV-2, DENV-3 e DENV-4, que são geneticamente distintos. Isso significa que uma pessoa pode ser infectada por um dos sorotipos e desenvolver imunidade específica apenas para esse sorotipo, tornando-se suscetível a infecções subsequentes pelos outros sorotipos. A resposta imune a um sorotipo específico pode aumentar o risco de desenvolver dengue grave em infecções subsequentes com outros sorotipos. Quando um mosquito fêmea do gênero *Aedes*, principalmente o *Aedes aegypti*, se alimenta de sangue de uma pessoa infectada com o vírus da dengue, ela se torna vetor do vírus. O vírus se replica no mosquito e migra para as glândulas salivares. Quando o mosquito infectado pica outra pessoa, o vírus é injetado na corrente sanguínea e inicia a infecção. Uma vez dentro do organismo humano, o vírus da dengue infecta células hospedeiras, principalmente células do sistema imunológico, como células dendríticas e macrófagos. O vírus utiliza receptores celulares para entrar nas células e, em seguida, começa a replicar seu material genético, produzindo novas partículas virais.

A resposta imune do hospedeiro é ativada em resposta à infecção viral, levando à produção de citocinas e outras moléculas inflamatórias. A gravidade da dengue está associada a uma resposta imune desregulada, que pode levar a danos teciduais, inflamação excessiva e aumento da permeabilidade dos vasos sanguíneos, resultando em complicações como dengue grave e síndrome do choque da dengue.

2.4.2 Chikungunya

A febre chikungunya (Galán-Huerta *et al.*, 2015) é uma doença viral transmitida principalmente pelos mosquitos *Aedes aegypti* e *Aedes albopictus*. Ela é causada pelo vírus chikungunya (CHIKV), que pertence ao gênero *Alphavirus*, da família *Togaviridae*. O vírus chikungunya possui um genoma de *RNA* de cadeia simples, encapsulado em uma cápside proteica. Essa cápside é cercada por uma camada lipídica derivada da membrana celular do hospedeiro infectado. O vírus contém proteínas de envelope na superfície, que desempenham um papel importante na entrada do vírus nas células do hospedeiro. Quando um mosquito infectado pica uma pessoa, o vírus é injetado na corrente sanguínea e inicia a infecção. O vírus se replica inicialmente nos tecidos da pele e, em seguida, dissemina-se para outras partes do corpo, como as articulações. A infecção pelo vírus chikungunya pode resultar em uma variedade de sintomas. Os sintomas mais comuns incluem febre alta, dores articulares intensas (artralgia), dores musculares, dor de cabeça, fadiga e erupções cutâneas. Esses sintomas podem ser debilitantes e durar semanas ou até meses em alguns casos. Em alguns casos raros, a febre chikungunya pode levar a complicações graves, como encefalite, problemas cardíacos e neurológicos.

2.4.3 Zika

O vírus Zika (Tropical *et al.*, 2022) é um vírus transmitido principalmente pelo mosquito *Aedes aegypti*, embora também possa ser transmitido por outros mosquitos do gênero *Aedes*, como o *Aedes albopictus*. O vírus Zika pertence à família *Flaviviridae* e ao gênero *Flavivirus*, o mesmo gênero que inclui os vírus da dengue, da febre amarela e da febre do Nilo Ocidental. O vírus Zika possui um genoma de *RNA* de cadeia simples e sentido positivo. Ele é composto por uma cápside proteica que envolve o material genético viral, e essa cápside é envolvida por uma membrana lipídica. Na superfície do vírus, há proteínas de envelope cruciais para a entrada do vírus nas células do hospedeiro. A infecção pelo vírus Zika pode ser assintomática em muitos casos. No entanto, quando os sintomas estão presentes, eles geralmente são leves e incluem febre baixa, erupção cutânea maculopapular, dores nas articulações, conjuntivite não purulenta, dores de cabeça e mal-estar geral. Esses sintomas podem durar de alguns dias a algumas semanas.

2.5 Filogenia

Filogenia é um modelo da história genealógica que representa as relações evolutivas entre espécies ou genes. É uma árvore que contém nós conectados por ramos, onde cada ramo representa a persistência de uma linhagem genética ao longo do tempo e cada nó representa o surgimento de uma nova linhagem. A filogenia é inferida a partir de dados de sequência ou outros dados e pode ser construída usando métodos baseados em distância ou em caracteres, como parcimônia, distância, verossimilhança e métodos bayesianos. A filogenia pode ser utilizada para o estudo de epidemiologia, rastrear surtos virais e incidência de novas variantes de vírus (Yang; Rannala, 2012).

Figura 2 – Árvore filogenética

Retirada de (Nascimento, 2021)

2.5.1 Máxima Verossimilhança

A máxima verossimilhança (Kishino; Hasegawa, 2001) é um método estatístico utilizado para estimar os parâmetros de um modelo a partir de dados observados. Na filogenética, a máxima verossimilhança é frequentemente usada para inferir a árvore filogenética que melhor explica os dados moleculares observados. O método de máxima verossimilhança envolve a escolha da árvore que maximiza a probabilidade de observar os dados moleculares, dada a árvore e o modelo evolutivo. Embora a máxima verossimilhança seja computacionalmente intensiva, ela é considerada uma das abordagens mais precisas para inferir árvores filogenéticas a partir de dados moleculares. A inferência de árvores por máxima verossimilhança envolve duas etapas de otimização: a otimização dos comprimentos dos ramos para calcular o escore da árvore para cada árvore candidata e uma busca no espaço de árvores para encontrar a árvore de máxima verossimilhança.

2.5.2 Máxima Parcimônia

A máxima parcimônia (Kannan; Wheeler, 2012) é um método de inferência filogenética que busca minimizar o número de mudanças necessárias para explicar a evolução de um conjunto de sequências. Isso é feito atribuindo estados de caracteres a nós interiores na árvore filogenética e calculando a pontuação da árvore, a soma dos comprimentos dos caracteres em todos os lugares. A árvore de máxima parcimônia é a árvore que minimiza essa pontuação. No entanto, a parcimônia não leva em consideração a probabilidade de diferentes mudanças em diferentes lugares, o que pode levar a árvores incorretas em alguns casos.

2.5.3 Inferência Bayesiana

A inferência bayesiana (Rannala; Yang, 1996) é uma abordagem estatística que utiliza a distribuição posterior para inferir sobre parâmetros desconhecidos. Essa abordagem permite a incorporação de informações a priori sobre os parâmetros, mas é importante realizar análises de robustez bayesiana para avaliar o impacto da priori nas estimativas posteriores. Os métodos bayesianos são amplamente utilizados em filogenética molecular para inferir a topologia das árvores filogenéticas e outros parâmetros evolutivos. A fórmula básica da inferência bayesiana é:

$$P(\theta|D) = \frac{P(D|\theta) \cdot P(\theta)}{P(D)}$$

onde θ é o parâmetro desconhecido, D é o conjunto de dados observados, $P(\theta|D)$ é a distribuição posterior, $P(D|\theta)$ é a verossimilhança, $P(\theta)$ é a distribuição a priori e $P(D)$ é a probabilidade marginal dos dados. Essa fórmula é conhecida como Teorema de Bayes e é a base da inferência bayesiana. A distribuição posterior é proporcional à verossimilhança multiplicada pela distribuição a priori, normalizada pela probabilidade marginal dos dados (Yang; Rannala, 2012).

2.5.4 Bootstrap

A análise *bootstrap* (Henderson, 2005) é um procedimento comumente utilizado para avaliar a confiança em uma estimativa de topologia de árvore. Nessa abordagem, os sítios no alinhamento de sequência são amostrados com reposição tantas vezes quanto o comprimento da sequência, gerando uma pseudo-amostra *bootstrap* que é do mesmo tamanho que o conjunto de dados original. Tipicamente, 100 ou 1.000 amostras *bootstrap* são geradas dessa maneira, e cada uma é analisada da mesma forma que o alinhamento de sequência original. As árvores inferidas a partir dessas amostras *bootstrap* são então tabuladas para calcular os valores de suporte *bootstrap*. Para cada clado na árvore estimada, seu valor de suporte *bootstrap* é simplesmente a proporção de árvores *bootstrap* que incluem esse clado.

2.6 *Arthropod Borne Virus database* - ABVdb

O ABVdb (RESTOVIC, 2018) é um banco de dados relacional desenvolvido por Dra. Maria Inês Valderrama Restovic que armazena genomas de vírus da dengue, Zika e chikungunya, fornecendo suporte para mineração de dados relacionada a esses três tipos de vírus. Sendo um banco de dados relacional, seus dados são armazenados em tabelas que se relacionam entre si.

Os dados armazenados no ABVdb são provenientes do banco de dados mundial *GenBank* e incluem genomas presentes em mosquitos, humanos e símios. Os dados mais relevantes armazenados são:

- Código de acesso no *GenBank* (Locus)
- Nome do isolado
- Data da coleta
- Origem da amostra (humano, mosquito ou símio)
- Região genômica
- Região geográfica
- Sorotipo (para o vírus da dengue)
- Sequência de nucleotídeos
- Tamanho da sequência

No caso da espécie ser *Homo sapiens*, são adicionadas informações adicionais, como:

- Gênero do paciente
- Idade
- Tipo da doença (especificado para cada tipo de vírus)

Essas informações são provenientes de publicações científicas indexadas no *PubMed*. O ABVdb é projetado para permitir a integração com ferramentas de filogenia, fornecendo uma plataforma abrangente para análises filogenéticas de vírus da dengue, Zika e chikungunya. Atualmente, o ABVdb está hospedado no domínio http://www.abvdb.uneb.br, onde os usuários podem acessar e utilizar os recursos disponíveis para pesquisar, visualizar e analisar os genomas de vírus armazenados no banco de dados.

2.7 *Genome Detective*

O *Genome Detective* é um software baseado na web que automatiza a identificação de vírus a partir de dados de sequenciamento de alta capacidade. Projetado para ser acessível e eficiente, ele utiliza um método inovador de alinhamento para construir genomas virais vinculando contigs de novo por referência, combinando pontuações de aminoácidos e nucleotídeos. Esta abordagem permite uma análise detalhada e precisa, facilitando a identificação de vírus em diversos tipos de amostras (Vilsker *et al.*, 2018).

O processo de funcionamento da plataforma envolve várias etapas principais. O usuário inicia o processo fazendo o upload dos dados de sequenciamento, que podem ser provenientes de várias tecnologias, como *Illumina* ou *nanopore*. Uma vez carregados, o software executa uma série de análises para identificar possíveis vírus, utilizando algoritmos avançados que combinam informações de nucleotídeos e aminoácidos para alinhar contigs com genomas virais de referência. Em seguida, os contigs alinhados são comparados com uma extensa base de dados de genomas virais conhecidos, permitindo a identificação precisa de vírus presentes na amostra, mesmo em casos de coinfecção ou presença de múltiplos agentes virais. Após a identificação, a ferramenta constrói os genomas virais completos, processo crucial para a caracterização detalhada do vírus, incluindo a detecção de novas variantes ou mutações. Finalmente, são gerados relatórios detalhados e visualizações dos dados analisados, facilitando a interpretação dos resultados pelo usuário (Vilsker *et al.*, 2018).

O sistema foi otimizado e validado com dados de sequenciamento de centenas de vírus, demonstrando alta precisão e eficiência. O Dr. Vagner Fonseca, um dos colaboradores deste projeto, é professor da Universidade do Estado da Bahia (UNEB) (Vilsker *et al.*, 2018).

A capacidade da plataforma de processar e analisar rapidamente grandes volumes de dados de sequenciamento torna-o uma ferramenta valiosa em situações de surto, onde a identificação rápida de agentes patogênicos é crucial. Além disso, sua aplicação pode ser estendida para a pesquisa em virologia, epidemiologia e desenvolvimento de vacinas, proporcionando uma plataforma robusta para a investigação e monitoramento de doenças virais.

A tecnologia representa um avanço significativo na identificação e análise de vírus a partir de dados de sequenciamento. Seu desenvolvimento e otimização continuam a aprimorar a capacidade dos cientistas em responder a desafios emergentes na área da virologia, oferecendo uma solução eficiente e precisa para a análise de dados complexos e está disponível online em: http://www.genomedetective.com/app/typingtool/virus/.

2.8 PSRM

O Método de Reconhecimento de Estados Paramétricos (*PSR - Parametric State Recognition Method*) (Grieves; Vickers, 2017) é uma técnica de aprendizado não supervisionado utilizada para analisar e agrupar dados, especialmente séries temporais. Ele foi inicialmente desenvolvido para o monitoramento de equipamentos industriais por meio de um conceito chamado de *Digital Twin*, que é uma representação virtual em tempo real de um objeto ou processo físico.

No PSR Method, os dados de entrada consistem em uma lista de números normalizados e características categóricas misturadas, que são os parâmetros analisados. Antes de aplicar o algoritmo, é realizada uma etapa de pré-processamento, que envolve duas camadas. A primeira camada realiza cálculos para obter diferenças finitas nas séries temporais, enquanto a segunda camada realiza médias acumulativas com base em janelas de tempo.

Os hiperparâmetros do método incluem o número de variáveis e as combinações e tipos de camadas de pré-processamento. Além disso, existem parâmetros de conhecimento, como o mapa de agrupamentos (*MOC - Map of Clusters*), o dicionário de estados (*CDIC - State Dictionary*) e o dicionário de famílias (*FDIC - Family Dictionary*). Esses parâmetros ajudam a representar os microestados e as conexões entre eles, formando os agrupamentos em níveis micro e macro.

Em resumo, o PSR Method permite agrupar e analisar dados usando características paramétricas. Ele é particularmente útil para identificar padrões ou estados específicos nos dados de entrada. Esses agrupamentos podem fornecer informações valiosas sobre o comportamento de sistemas ou objetos, possibilitando uma melhor compreensão e auxiliando na tomada de decisões informadas.

Durante a etapa de treinamento do Método de Reconhecimento de Estados Paramétricos (*PSRM*), o conjunto de dados brutos é processado em duas etapas principais: a etapa de partição e a etapa aglomerativa.

Na etapa de partição, as amostras de treinamento são analisadas individualmente. Para cada característica numérica normalizada, um histograma é construído e um algoritmo de busca de agrupamentos é aplicado. Os picos identificados no histograma representam agrupamentos para cada característica. Uma distribuição de probabilidade é ajustada para cada pico, utilizando o intervalo entre os vales adjacentes. Essas distribuições são usadas como funções de pertinência para atribuir um valor a uma característica pertencente a um determinado agrupamento. Os limites de cada agrupamento são armazenados no Mapa de Agrupamentos (*MOC*).

Para características categóricas, é atribuído um número de agrupamentos igual ao

número de classes de características categóricas. Uma parte do *MOC* contém as associações entre classe e agrupamento. Com base nisso, é criada uma assinatura de micro-agrupamento para cada amostra de dados, representada por um vetor de valores para cada característica. Essas micro-assinaturas são armazenadas no Dicionário de Estados (*CDIC*), recebendo um índice ordinal.

Na etapa aglomerativa, as assinaturas de micro-agrupamentos são comparadas entre si para calcular a distância utilizando uma métrica adequada para dados híbridos (numéricos e categóricos). As distâncias são organizadas em uma matriz triangular superior. Em seguida, é aplicado um método hierárquico de agrupamento, conhecido como método de vizinho mais próximo, para unir os micro-agrupamentos em macro-agrupamentos. Esse processo envolve a criação de um Dicionário de Famílias (*FDIC*), onde as linhas representam as famílias e as colunas são as assinaturas de micro-agrupamentos. As distâncias entre os micro-agrupamentos são percorridas e, com base nas condições definidas, os micro-agrupamentos são atribuídos à mesma família ou uma nova família é criada.

Após a etapa de treinamento, na etapa pós-treino, são realizados cálculos adicionais para cada macro-agrupamento. Isso inclui a coleta da distribuição de distância dos micro-agrupamentos internos, o cálculo da média e variância das distâncias e a determinação do maior valor de distância entre os membros do macro-agrupamento. Além disso, são calculados os centroides dos macro-agrupamentos como o centro de massa. Uma matriz de distâncias é então calculada entre os centroides para avaliar a qualidade do agrupamento.

O método pode ser operado e retreinado usando o *FDIC* e o *CDIC* preenchidos. Durante a operação, para uma nova instância de dados, a assinatura de micro-agrupamento é procurada no *CDIC*. Se encontrada, é atribuída à mesma família. Caso contrário, é adicionada ao *CDIC* e a distância entre a nova assinatura e as outras é calculada para encontrar a assinatura mais próxima. Com base em critérios de distância, a nova instância é atribuída ao macro-agrupamento correspondente ou um novo macro-agrupamento é criado no *FDIC*.

O pré-processamento dos dados pode ser realizado de forma serial, paralela ou híbrida. No pré-processamento paralelo, as características brutas são normalizadas e selecionadas, e em seguida são calculadas as derivadas normalizadas e as médias integrais normalizadas das características selecionadas. Os conjuntos resultantes são agrupados para formar o vetor de entrada. No pré-processamento serial, o processo é dividido em dois fluxos: pré-processamento D-I e pré-processamento I-D. Em ambos os casos, as características brutas são normalizadas e selecionadas. No pré-processamento D-I, são calculadas as derivadas normalizadas das características selecionadas, seguidas pelo cálculo das médias integrais normalizadas. No pré-processamento I-D, é realizado o cálculo das médias integrais normalizadas, seguido pelo cálculo das derivadas normalizadas. Os conjuntos resultantes

são agrupados para formar o vetor de entrada (Nascimento, 2021).

2.9 CBUC

O algoritmo *Codon Based Unsupervised Classification* (*CBUC*) é uma adaptação do algoritmo *PSRM*, desenvolvido pelo Dr. Diego Gervásio Frias Suaréz, utilizando a linguagem de programação *Scilab* na versão 6.0. O *CBUC* foi projetado para agrupar sequências em famílias, de maneira semelhante a uma árvore filogenética, e tem como objetivo classificar de forma rápida e econômica uma nova sequência em um grupo monofilético.

No Trabalho de Conclusão de Curso (*TCC*) de (Nascimento, 2021), o algoritmo *CBUC* foi implementado na linguagem de programação *Python*, utilizando o ambiente do *Google Colab*, que fornece um *notebook Jupyter* na nuvem (Kluyver *et al.*, 2016). O programa recebe um arquivo no formato *FASTA*, que é um formato de arquivo usado para representar sequências biológicas, como sequências de *DNA*, *RNA* ou proteínas. O formato *FASTA* consiste em uma linha de cabeçalho iniciada com o caractere »", contendo informações descritivas sobre a sequência, seguida pelas linhas de sequência em si.

O programa realiza a leitura do arquivo *FASTA*, transformando as sequências em uma lista de números normalizados. Em seguida, seleciona conjuntos de três caracteres e os transforma em números de 2 a 65, representando os códons, que são sequências de três bases nitrogenadas.

Com a lista de sequências normalizadas, é construída uma base de conhecimento, a partir da qual é calculada a matriz de frequência de códons das sequências. Essa matriz é utilizada para gerar agrupamentos, resultando em um mapa de agrupamentos. Além disso, é calculada a quantidade de agrupamentos em cada posição da sequência. Cada agrupamento recebe um código único.

A frequência de cada padrão de agrupamento é calculada, e com base nesses cálculos, é gerada uma matriz de distância. Essa matriz de distância é usada para realizar agrupamentos de padrões em famílias.

Os resultados dos agrupamentos de códons e dos agrupamentos de padrões de códons (famílias) são exibidos no console utilizando a biblioteca *matplotlib* do *Python* (Nascimento, 2021).

2.10 Trabalhos Correlatos

Nascimento implementa o *CBUC* em *Python*, que é o algoritmo utilizado e a principal motivação para a realização deste trabalho. Karim *et al.* utiliza *Convolutional Embedded Network* (*CEN*) para identificar variantes genéticas. Essa rede combina duas

redes neurais *Convolutional Embedded Clustering* (*CEC*) e *Convolutional Autoencoder* (*CAE*) para prever etnias genéticas com variantes genéticas (*GV*). Essa abordagem é comparada com outros algoritmos tais como *VariantSpark* e *ADMIXTURE*. Utilizando as métricas índice *Rand* ajustado (*ARI*), informação mútua normalizada (*NMI*), precisão de clusterização (*ACC*), homogeneidade, completude, tempo de execução.

Wuyun *et al.* propõe o *PHiMM* que é um algoritmo que faz inferência sobre sequências genéticas utilizando um modelo de deriva genética, substituições, recombinação e fluxo gênico, combinando uma técnica de aproximação baseada em coalescência.

Kim *et al.* propõe um algoritmo para identificar associações entre os recursos genéticos (*SNP*) e as medidas derivadas de imagens de ressonância magnética utilizando o método *Tree-Guided Sparse Learning* (*TGSL*). O *TGSL* é um método de seleção de recursos que utiliza uma abordagem de regularização baseada em árvore, que leva em consideração a estrutura hierárquica natural dos *SNPs*. A construção da árvore é baseada em conhecimento prévio sobre a relação funcional e genética entre os *SNPs*. Cada nó da árvore representa um grupo de *SNPs* com relação funcional semelhante, e os diferentes níveis da árvore representam diferentes níveis de agrupamento.

Cleemput *et al.* propõe o *Genome Detective* é uma aplicação *web* para a montagem de todos os genomas de vírus conhecidos a partir de dados de sequenciamento de nova geração. Este aplicativo permite a identificação de *clusters* filogenéticos a partir de genomas montados no formato *FASTA*.

3 METODOLOGIA

Este trabalho utilizou uma metodologia de estudo comparativo (Collier, 1993) para validar o algoritmo de bioinformática CBUC, focando em sua capacidade de identificar genótipos de sequências do Vírus Zika. Para isso, empregou-se um conjunto de sequências de genótipos conhecidos como referência e comparou-se a concordância dos resultados obtidos pelo CBUC com os do método tradicional de genotipagem, representado pela ferramenta *Genome Detective*.

3.1 Desenvolvimento do projeto

O diagrama da Figura 3 descreve a arquitetura da solução utilizada neste trabalho. O processo começa com a base de dados ABVdb, que armazena sequências genotipadas. Essas sequências servem como referência para o CBUC, um componente responsável por identificar e analisar novas sequências.

As sequências de Zika são recebidas de uma fonte externa e, antes de serem processadas pelo *CBUC*, passam por um estágio de pré-processamento. Durante o pré-processamento, as sequências são ajustadas adicionando-se caracteres '-' (hífens) para garantir que todas tenham o mesmo tamanho e sejam múltiplas de 3, o que é necessário para o CBUC poder trabalhar corretamente com elas.

Uma vez pré-processadas, as sequências de Zika são enviadas para o *CBUC*. O *CBUC* então utiliza as sequências de referência do ABVdb para identificar as sequências de Zika. Esse processo de identificação envolve a comparação das sequências de Zika com as referências no ABVdb, permitindo determinar correspondências ou diferenças.

Em resumo, o fluxo de trabalho começa com a recepção das sequências de Zika, segue para o pré-processamento onde são ajustadas, e culmina no CBUC, que as compara com as sequências de referência do ABVdb para identificar as novas sequências.

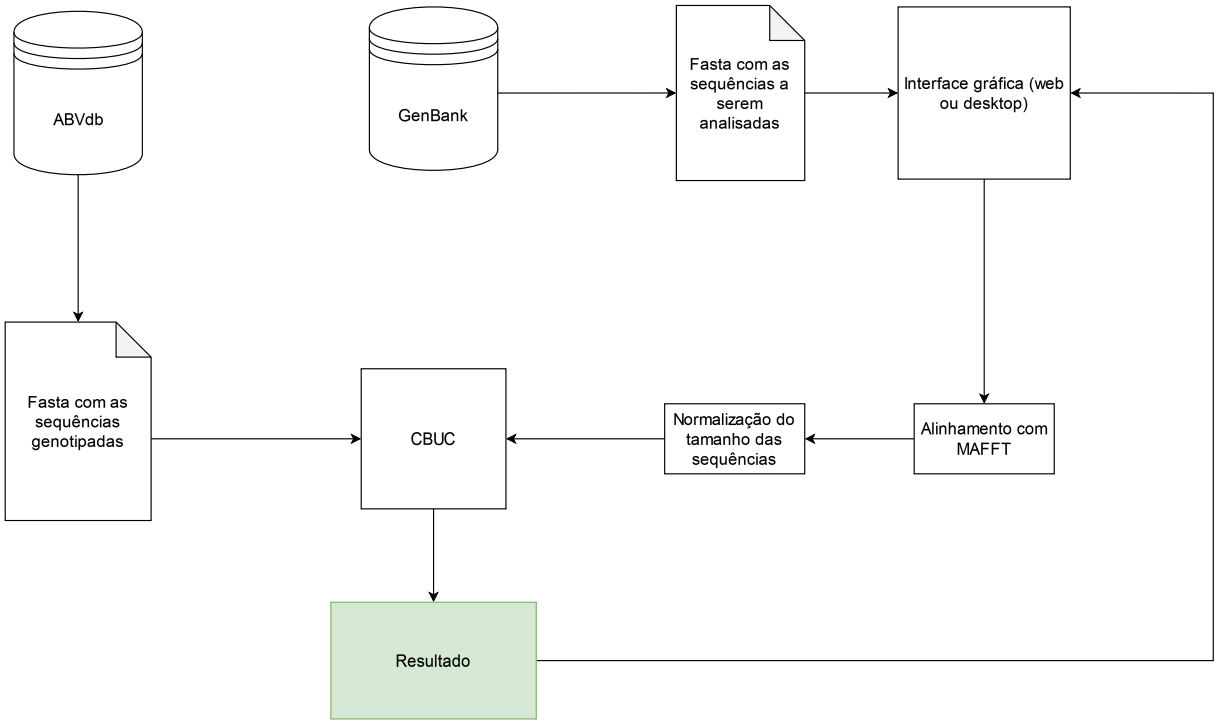


Figura 3 – Arquitetura da solução

Fonte: O autor

3.1.1 Especificação da máquina utilizada

Os experimentos foram realizados em um laptop com as seguintes especificações:

- **Sistema operacional:** *Windows 11 Home Single Language*;
- **Processador:** *Intel Core i5* (11ª geração, 11400h), hexa-core, 2,7 GHz;
- **Gpu:** *Nvidia GeForce GTX* 1650;
- **Memória RAM:** 24 GB DDR4 2933 MHz;
- **Armazenamento:** *SSD NVMe* 512 GB + HD 1 TB.

3.1.2 Ferramentas de software utilizadas

- ***Python*:** Linguagem de programação interpretada, amplamente utilizada em desenvolvimento web, científico e automação de tarefas.
- ***JavaScript*:** Linguagem de script amplamente utilizada para o desenvolvimento de aplicações web interativas.
- ***MAFFT***: Ferramenta de alinhamento de sequências utilizada para alinhar as sequências genômicas.

- ***React (Javascript)*:** Biblioteca JavaScript utilizada para desenvolver a interface web.
- ***Next.js (Javascript)*:** Framework JavaScript integrado ao React para otimizar a interface web.
- ***FastAPI (Python)*:** Framework Python utilizado para criar endpoints para processamento de sequências em formato FASTA.
- ***PyQt5 (Python)*:** Framework Python utilizado para construir a interface desktop.

3.1.3 Preparação do Conjunto de Dados

Para os experimentos, foram retiradas 514 sequências completas genotipadas do Vírus Zika do banco de dados ABVdb, das quais foram selecionadas 100 para que o CBUC possa utilizá-las como referências em seus agrupamentos.

Para os testes, foi utilizado um conjunto de dados com 702 sequências completas, junto com outros dois conjuntos de dados, um com 29 sequências e outro com 26 sequências que, entre as completas, também possuem sequências de tamanho parcial.

3.1.4 Adaptações

Devido às limitações do algoritmo *CBUC*, como a exigência de um arquivo FASTA com sequências de mesmo tamanho e a falta de uma interface amigável para o usuário, foi necessário realizar adaptações no código-fonte do *CBUC*. Primeiramente, reduzimos o número de clusters (famílias) para um cluster para as sequências africanas e outro para asiáticas, mas isso pode ser modificado de forma simples no código, o que foi essencial para a identificação precisa do genótipo dos dados genômicos do Vírus Zika. Além disso, incorporamos uma funcionalidade de alinhamento utilizando o *MAFFT* (Katoh; Standley, 2013). Após o alinhamento, as sequências foram preenchidas com o caractere '-' para garantir que todas tivessem o mesmo tamanho, possibilitando assim o processamento pelo *CBUC*. Essas adaptações foram fundamentais, pois o algoritmo só funciona corretamente com sequências de tamanhos iguais.

3.2 Desenho experimental

Primeiramente, deve-se obter as sequências do Vírus Zika a partir do *GenBank*, selecionando tanto sequências completas quanto parciais. Em seguida, todas as sequências coletadas devem ser compiladas em um único arquivo FASTA. Com o arquivo FASTA pronto, deve-se executar o algoritmo *CBUC*. Posteriormente, o mesmo arquivo FASTA deve ser submetido à ferramenta *Genome Detective* (Vilsker *et al.*, 2018). O objetivo é

verificar a precisão do CBUC na identificação das sequências, comparando os resultados obtidos com os do *Genome Detective*. A métrica de precisão utilizada para essa comparação é:

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}} \tag{3.1}$$

onde VP representa os verdadeiros positivos e FP representa os falsos positivos.

3.2.1 Coleta de Sequências

Para iniciar o experimento, devem-se obter as sequências do Vírus Zika a partir do banco de dados *GenBank*. A seleção deve incluir tanto sequências completas quanto parciais do vírus. Esse passo é crucial para garantir a representatividade e a diversidade das amostras utilizadas na análise.

3.2.2 Compilação das Sequências

Após a coleta, todas as sequências devem ser compiladas em um único arquivo no formato *FASTA*. Este arquivo servirá como entrada tanto para o algoritmo CBUC quanto para a ferramenta de comparação.

3.2.3 Execução do Algoritmo CBUC

Com o arquivo *FASTA* preparado, o próximo passo é executar o algoritmo *CBUC*. Este algoritmo processará as sequências e fornecerá os genótipos correspondentes, além de outros dados relevantes para a análise.

3.2.4 Comparação com Genome Detective

O mesmo arquivo *FASTA* deve ser submetido à ferramenta *Genome Detective* (Vilsker *et al.*, 2018). Esta etapa é fundamental para validar a precisão do *CBUC* na identificação das sequências de RNA do Vírus Zika.

3.2.5 Análise da Precisão

Para avaliar a precisão do algoritmo *CBUC*, os resultados obtidos devem ser comparados com os resultados do *Genome Detective*. A métrica de precisão a ser utilizada é definida pela seguinte fórmula:

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}} \tag{3.2}$$

onde:

- VP representa os verdadeiros positivos,
- FP representa os falsos positivos.

3.2.6 Interpretação dos Resultados

A comparação dos resultados de ambos os algoritmos permitirá avaliar a eficácia do *CBUC* em identificar corretamente as sequências de RNA do Vírus Zika. A métrica de precisão fornecerá uma medida quantitativa da precisão do algoritmo, indicando a proporção de verdadeiros positivos em relação ao total de identificações positivas.

3.3 Interface Web

A interface web foi projetada para otimizar o acesso e a utilização do *CBUC* através de navegadores. Esta plataforma permite a submissão de sequências de RNA, a execução de análises de genótipos e a visualização dos resultados de maneira intuitiva. A interface foi desenvolvida utilizando a biblioteca JavaScript React, integrada ao framework Next.js (Thakkar, 2020). Essa integração possibilita a comunicação com dois endpoints criados na linguagem de programação Python utilizando o framework FastAPI: */process_fasta/* e */upload_and_process_fasta/*, como pode ser observado na Figura 4.

3.3.1 Endpoints

- */process_fasta/*: Utilizado para processar texto no formato FASTA.
- */upload_and_process_fasta/*: Utilizado para processar arquivos no formato FASTA.

3.3.2 Funcionalidades da Interface

- **Upload de Arquivos FASTA**: Os usuários podem fazer o upload de arquivos no formato FASTA para processamento.
- **Inserção de Sequências de RNA**: É possível inserir sequências de RNA do Zika em formato FASTA diretamente no campo de texto.
- **Visualização dos Resultados**: Após o processamento, os resultados apresentam o nome das sequências juntamente com seus genótipos.
- **Download dos Resultados**: Há a opção de baixar os resultados no formato CSV ou em um formato aceito pelo Microsoft Excel (xlsx) para análise posterior.

A interface proporciona uma experiência de usuário fluida e eficiente, facilitando o processo de análise de dados genotípicos e a obtenção de resultados de forma acessível e organizada.

Figura 4 – Api para fazer o processamento das sequências

Fonte: O autor

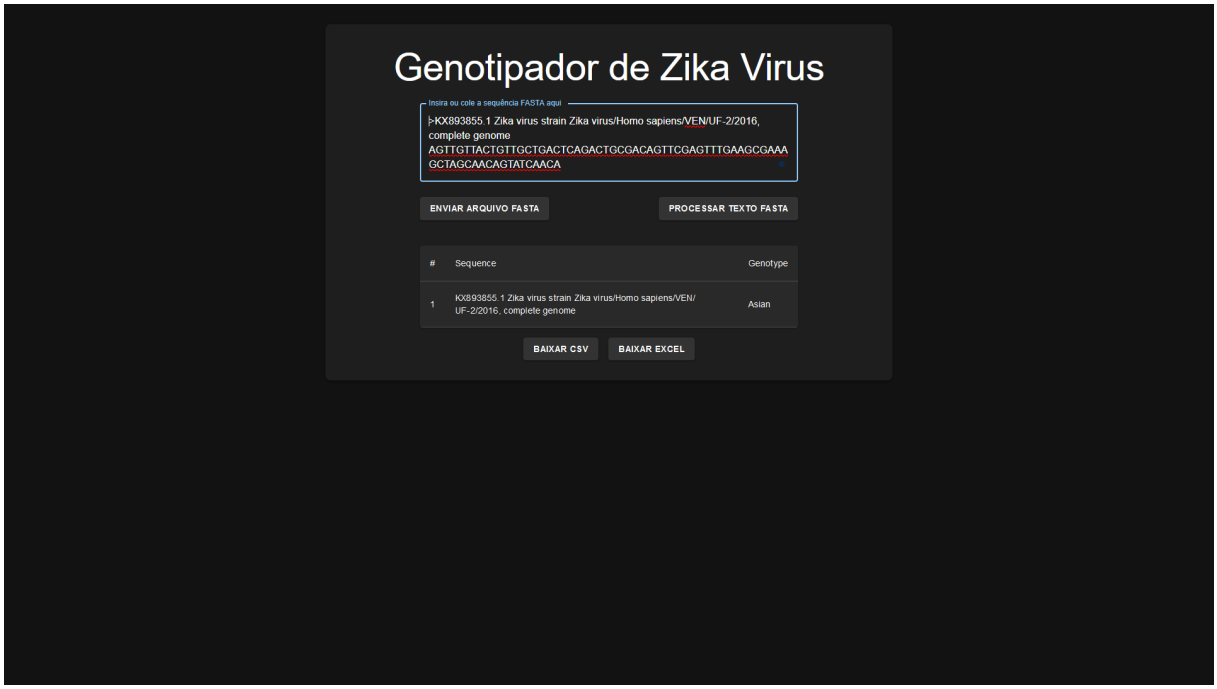


Figura 5 – Interface web

Fonte: O autor

3.4 Interface Desktop

A interface desktop foi desenvolvida para oferecer uma experiência semelhante à versão web, permitindo o acesso e a utilização do *CBUC* de forma intuitiva. Esta aplicação permite a submissão de sequências de RNA, a execução de análises de genótipos e a visualização dos resultados de maneira eficiente. A interface foi construída utilizando o framework PyQt5 em Python. De forma semelhante à sua versão web, a interface desktop permite que o usuário faça o upload de arquivos FASTA para processamento ou insira sequências de RNA do Zika diretamente na interface. Após o processamento, o resultado é apresentado com o nome das sequências e seus genótipos. Adicionalmente, os resultados podem ser exportados no formato CSV ou em um formato compatível com o Microsoft Excel (xlsx) para análise posterior.

3.4.1 Funcionalidades da Interface

- **Upload de Arquivos FASTA**: Os usuários podem fazer o upload de arquivos no formato FASTA para processamento.
- **Inserção de Sequências de RNA**: É possível inserir sequências de RNA do Zika em formato FASTA diretamente na interface.
- **Visualização dos Resultados**: Após o processamento, os resultados apresentam o nome das sequências juntamente com seus genótipos.
- **Download dos Resultados**: Há a opção de baixar os resultados no formato CSV ou em um formato aceito pelo Microsoft Excel (xlsx) para análise posterior.

A interface desktop proporciona uma experiência de usuário fluida e eficiente, facilitando o processo de análise de dados genotípicos e a obtenção de resultados de forma acessível e organizada.

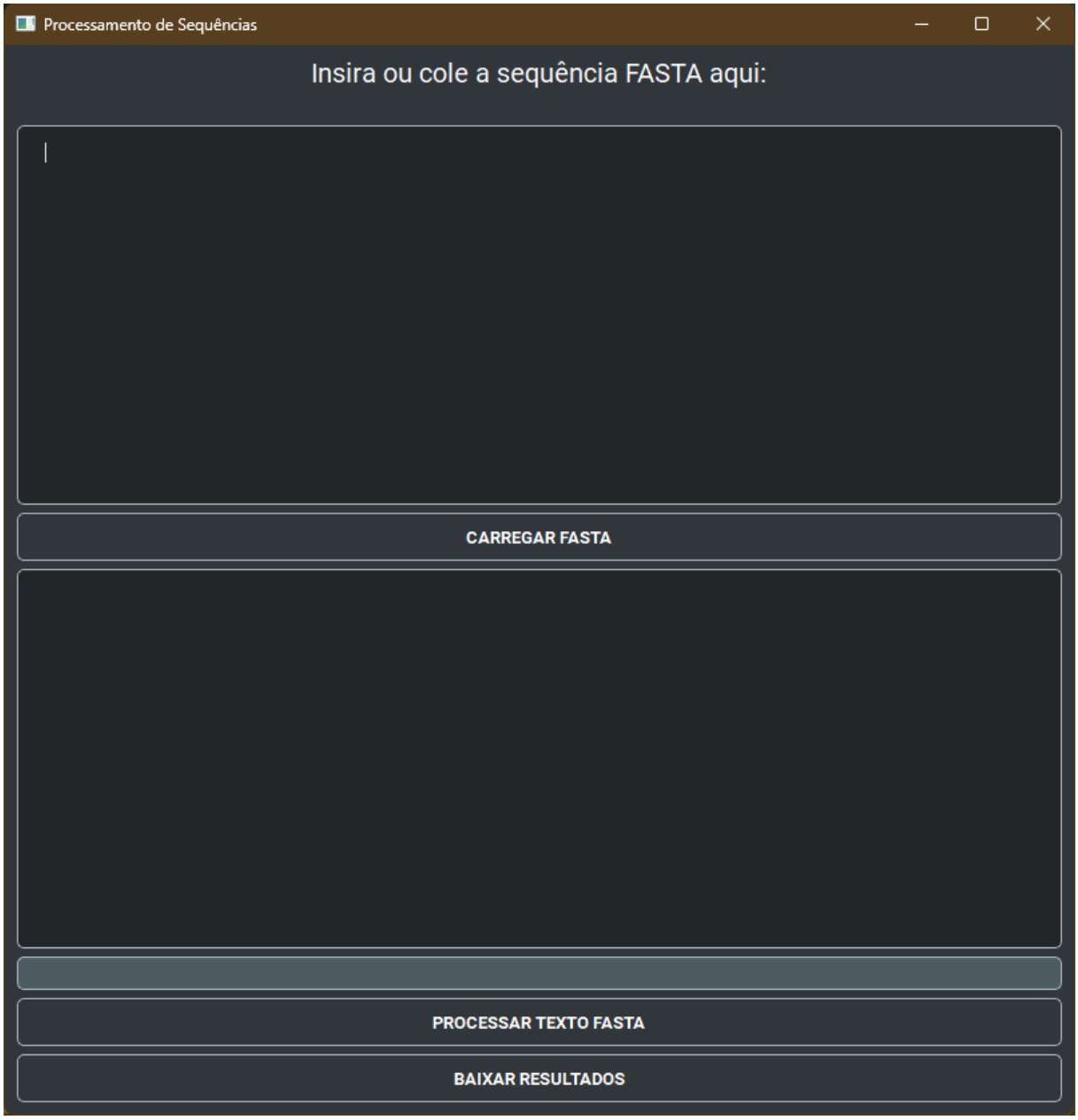


Figura 6 – Interface Desktop

Fonte: O autor

4 RESULTADOS

4.1 Primeiro experimento

O primeiro experimento utilizou 702 sequências completas do Vírus Zika. O algoritmo *CBUC* identificou 116 sequências do genótipo africano e 586 do genótipo asiático assim como *Genome Detective*. Os resultados foram comparados com os obtidos pela ferramenta *Genome Detective*, alcançando uma precisão de 100% para as sequências completas do Vírus Zika.

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}} = \frac{702}{702 + 0} = 1 \tag{4.1}$$

(a) Resultado *CBUC* (b) Resultado *Genome Detective*

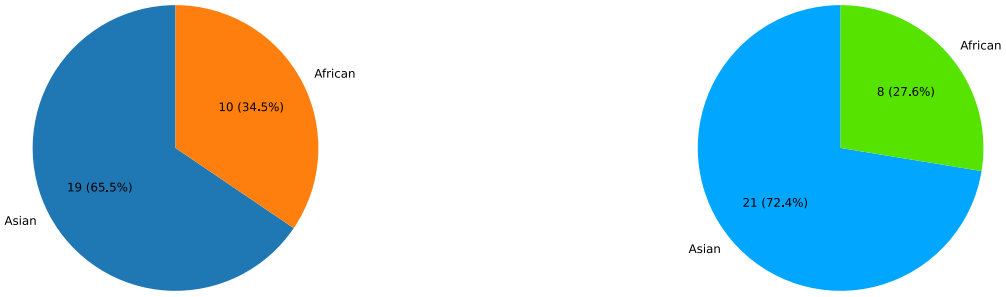
Figura 7 – Comparação dos resultados do primeiro experimento

Fonte: O autor

4.2 Segundo experimento

O segundo experimento usou um dataset com 27 sequências completas e 2 sequências parciais, ambas de tamanho 2778 nucleotídeos. O *CBUC* identificou 19 sequências asiáticas e 10 africanas, enquanto que o Genome detective 21 asiáticas e 8 africanas .

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}} = \frac{27}{27 + 2} = \frac{27}{29} \approx 0.931 \tag{4.2}$$



(a) Resultado *CBUC* (b) Resultado *Genome Detective*

Figura 8 – Comparação dos resultados do segundo experimento

Fonte: O autor

4.3 Terceiro experimento

O terceiro experimento contou com um dataset de 20 sequências completas e 6 sequências parciais, com tamanhos variados (1921, 1512, 2 de 2778, 2 de 2601 nucleotídeos). O *CBUC* identificou 20 sequências asiáticas e não conseguiu classificar as 6 restantes, entretanto o *Genome Detective* identificou todas como asiáticas.

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}} = \frac{20}{20 + 6} = \frac{20}{26} \approx 0.769 \tag{4.3}$$

(a) Resultado CBUC (b) Resultado *Genome Detective*

Figura 9 – Comparação dos resultados do terceiro experimento

Fonte: O autor

4.4 Quarto experimento

O experimento consiste em refazer o terceiro experimento, porém, as sequências de treinamento serão recortadas da posição 0 até 3000, que é aproximadamente o tamanho

das sequências parciais de teste. Além disso, essa região corresponde ao mesmo trecho do genoma das sequências, visto que elas estão localizadas no início do genoma. O resultado foi que o *CBUC* foi capaz de identificar corretamente as sequências de tamanho parcial, entretanto, não conseguiu identificar as completas. Com isso, é possível observar que os gráficos ficaram invertidos em comparação ao experimento que é feito utilizando as sequências completas.

(a) Resultado CBUC com dataset de sequências completas

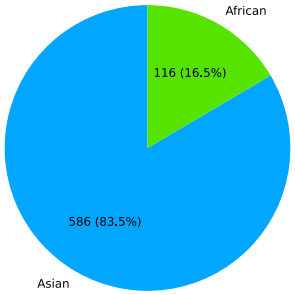
(b) Resultado com dataset com as sequências recortadas

Figura 10 – Comparação com dataset completo de sequências completas com as recortdas experimento

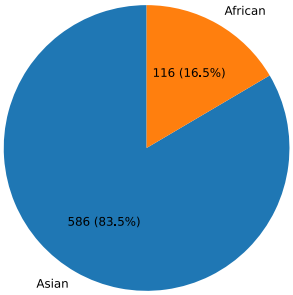
Fonte: O autor

4.5 Quinto experimento

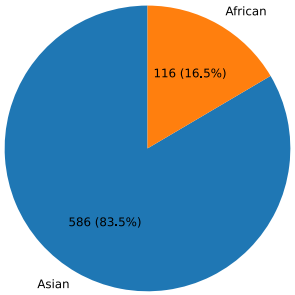
Neste experimento, o objetivo foi avaliar a precisão do *CBUC* ao diminuir gradativamente o tamanho das sequências em 1000 nucleotídeos até que o algoritmo perca a precisão. Os experimentos foram feitos diminuindo a quantidade de nucleotídeos do arquivo *Fasta* a partir do final, e o resultado observado foi que, até o tamanho de 8000 nucleotídeos, ele obtém 100% de acerto. Contudo, com sequências menores, o algoritmo não funciona corretamente, pois as sequências passadas se agrupam em um único grupo.



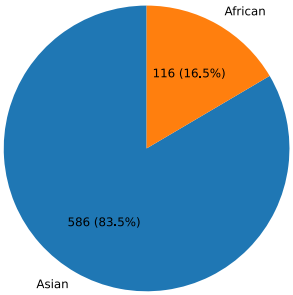
(a) Resultado *Genome Detective*



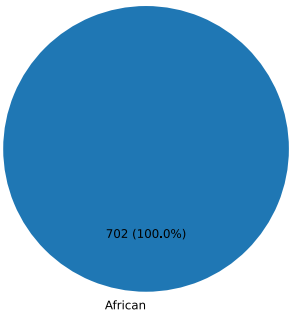
(b) Tamanho 10000



(c) Tamanho 9000



(d) Tamanho 8000



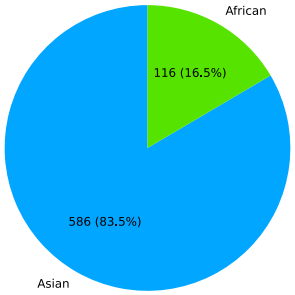
(e) Tamanho 7000

Figura 11 – Análise da performance para o tamanho das sequências

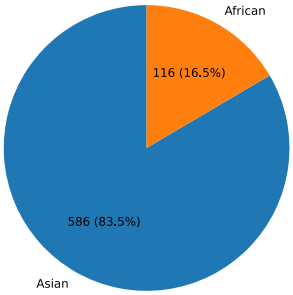
Fonte: O autor

4.6 Sexto experimento

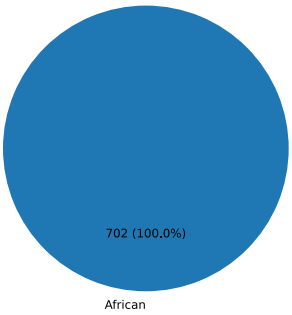
Semelhante ao experimento anterior, desta vez a parte recortada foi o começo do genoma do dataset utilizado. Os resultados foram precisos até que o recorte atingiu 1000 nucleotídeos.



(a) Resultado *Genome Detective*



(b) removido 1000



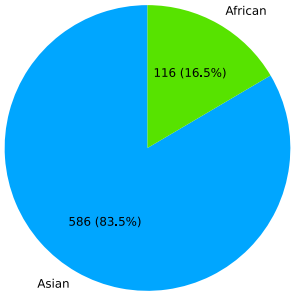
(c) removido 2000

Figura 12 – Análise da performance para o tamanho das sequências quando recortados no começo

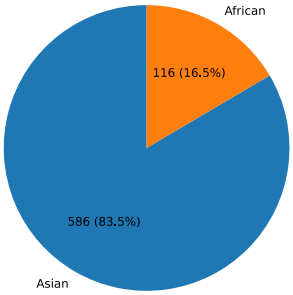
Fonte: O autor

4.7 Experimento com o Envelope

Esse experimento foi realizado com uma abordagem diferente: ao invés de utilizar as sequências completas como referência no *CBUC*, foi usada uma região específica do genoma, que contém o envelope, indo de 108 até 2478 do genoma alinhado. Essa região foi usada tanto para a identificação das sequências de teste, que também foram recortadas na mesma região. O que se observa é que, quando as sequências têm um tamanho parecido com a sequência utilizada como referência e estão na mesma região do genoma, o *CBUC* apresenta uma precisão de 100%.



(a) Resultado CBUC com dataset de sequências completas



(b) Resultado com dataset com as sequências recortadas

Figura 13 – Comparação com dataset completo de sequências completas com as recortadas experimento

Fonte: O autor

4.8 Analise de desempenho

O experimento foi realizado com o objetivo de medir o tempo de execução do algoritmo ao processar diferentes quantidades de sequências. Para obter resultados robustos, cada experimento foi repetido 10 vezes para três diferentes conjuntos de dados: 1, 10 e 100 sequências completas.

Após a execução dos experimentos, foram calculadas as seguintes estatísticas para o tempo de execução dos algoritmos: a média, a mediana e o desvio padrão. A média dos tempos de execução fornece uma estimativa do tempo médio necessário para o algoritmo processar as sequências. A mediana dos tempos de execução representa o valor central dos dados e é menos suscetível a valores atípicos (outliers) em comparação com a média. O desvio padrão dos tempos de execução quantifica a variabilidade dos tempos de execução, indicando a consistência do desempenho do algoritmo.

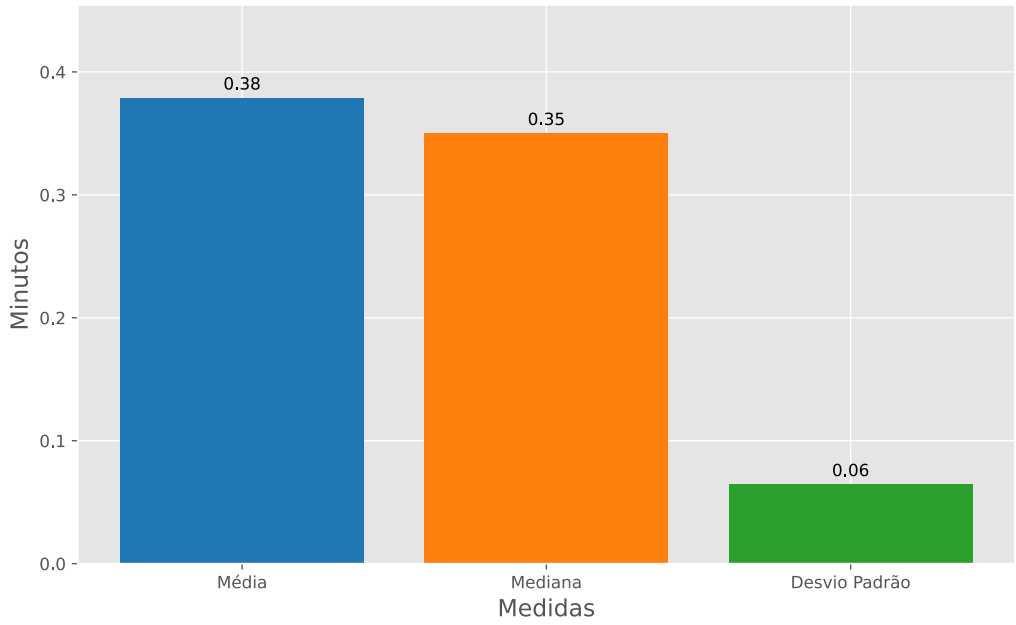


Figura 14 – Tempo de execução do algoritmo para processar uma sequência completa 10 vezes

Fonte: O autor

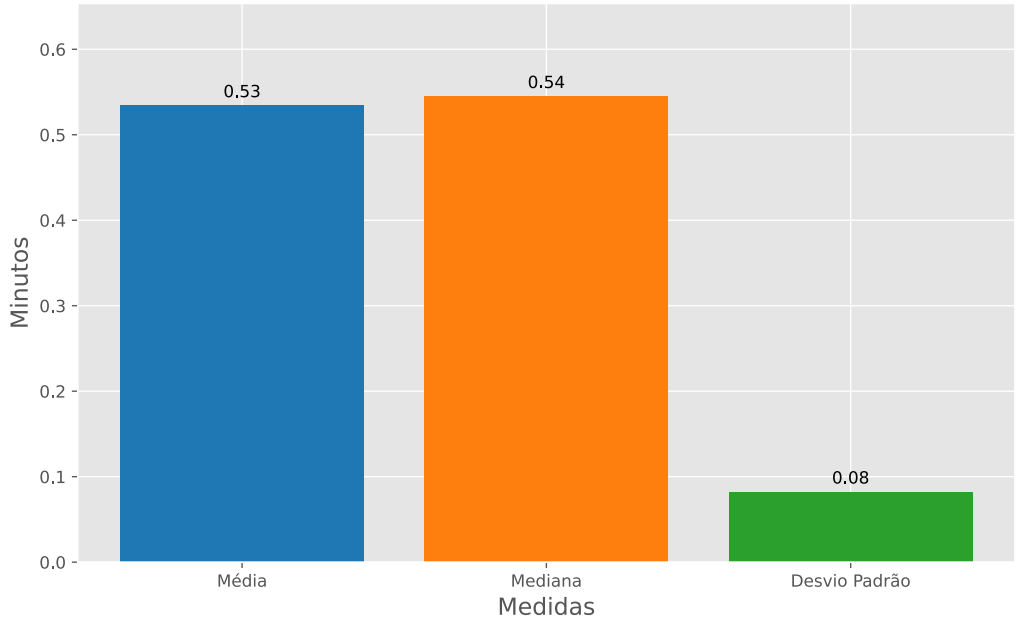


Figura 15 – Tempo de execução do algoritmo para processar 10 sequências completas 10 vezes

Fonte: O autor

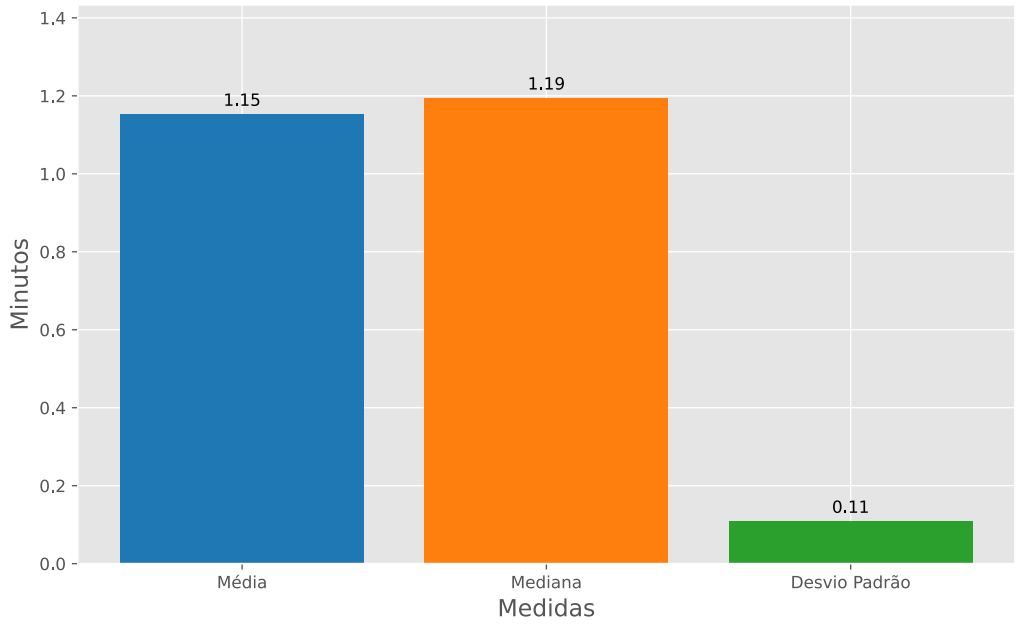


Figura 16 – Tempo de execução do algoritmo para processar 100 sequências completas 10 vezes

Fonte: O autor

4.9 Análise de Resultados

Os experimentos realizados mostraram que o *CBUC* apresenta uma precisão de 100% na identificação de sequências de tamanho semelhante às utilizadas no treinamento, especialmente quando as sequências estão na mesma região do genoma. No entanto, a precisão do *CBUC* diminui significativamente quando se trata de sequências de tamanhos diferentes. Observou-se que o *CBUC* não lida bem com datasets contendo sequências de tamanhos variados, uma vez que foi projetado para operar com sequências de tamanho uniforme. Para abordar essa limitação, foi implementada uma função que adiciona o caractere "-" ao final das sequências alinhadas, com o objetivo de padronizar o tamanho das sequências. Apesar dessa medida, o algoritmo ainda não alcança a mesma eficiência na identificação de sequências de tamanhos diferentes. Mesmo com essas limitações, o *CBUC* demonstrou ser uma ferramenta eficaz para a identificação de sequências completas do vírus Zika, apresentando um desempenho comparável ao *Genome Detective* para sequências completas. Essa constatação reforça a viabilidade do *CBUC* como uma ferramenta de identificação em cenários específicos, onde as sequências de entrada são de tamanhos uniformes.

5 CONCLUSÃO

Neste estudo, investigamos o desempenho do algoritmo *CBUC* na identificação de sequências do Zika vírus. Descobrimos que o *CBUC* enfrenta limitações significativas quando confrontado com sequências de tamanhos diferentes das utilizadas no treinamento. Esse problema é especialmente pronunciado em datasets com uma grande variação de tamanhos de sequência, resultando em uma diminuição da precisão da identificação.

No entanto, é importante destacar que o *CBUC* demonstrou uma capacidade notável de identificar 100% das sequências completas presentes no dataset de teste. Isso sugere que, quando as sequências do dataset de teste têm um tamanho semelhante às utilizadas no treinamento, o *CBUC* pode ser uma ferramenta eficaz para a identificação de sequências completas do Zika vírus.

Uma consideração importante a ser levada em conta é que o sucesso do *CBUC* na identificação de sequências completas está intrinsecamente ligado à composição do dataset de treinamento. Como o dataset de treinamento consistiu principalmente de sequências completas, o *CBUC* foi bem-sucedido em identificar essas sequências com alta precisão.

Em suma, este estudo destaca tanto as capacidades quanto as limitações do *CBUC* na identificação de sequências do vírus Zika. Embora tenha mostrado um desempenho excepcional na identificação de sequências completas, sua eficácia pode ser comprometida em conjuntos de dados com uma variação significativa de tamanhos de sequência. Futuras pesquisas podem se concentrar em abordagens para melhorar a robustez do *CBUC* diante dessa variabilidade de tamanho de sequência. Trabalhos futuros podem explorar a implementação do algoritmo para outros tipos de vírus e desenvolver métodos para lidar com sequências de diferentes tamanhos.

REFERÊNCIAS

ALBERTS, B. **Molecular Biology of the Cell**. W.W. Norton, 2017. ISBN 9781317563754. Disponível em: https://books.google.com.br/books?id=jK6UBQAAQBAJ. Citado nas páginas 8, 15 e 16.

CLEEMPUT, S. *et al.* Genome detective coronavirus typing tool for rapid identification and characterization of novel coronavirus genomes. 2020. Citado na página 26.

COLLIER, D. The comparative method. In: _____. [S.l.: s.n.], 1993. p. 105–119. Citado na página 27.

GALáN-HUERTA, K. *et al.* Chikungunya virus: A general overview. **Medicina Universitaria**, Elsevier, v. 17, p. 175–183, 7 2015. ISSN 1665-5796. Disponível em: https://www.elsevier.es/en-revista-medicina-universitaria-304-articulo-chikungunya-virus-a-general-overview-S1665579615000587. Citado na página 18.

GRIEVES, M.; VICKERS, J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In: KAHLEN, F.-J.; FLUMERFELT, S.; ALVES, A. (Ed.). **Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches**. Cham: Springer International Publishing, 2017. p. 85–113. ISBN 978-3-319-38756-7. Disponível em: https://doi.org/10.1007/978-3-319-38756-7_4. Citado na página 23.

HENDERSON, A. R. The bootstrap: a technique for data-driven statistics. using computer-intensive analyses to explore experimental data. **Clinica chimica acta; international journal of clinical chemistry**, Clin Chim Acta, v. 359, p. 1–26, 9 2005. ISSN 0009-8981. Disponível em: https://pubmed.ncbi.nlm.nih.gov/15936746/. Citado na página 20.

KANNAN, L.; WHEELER, W. C. Maximum parsimony on phylogenetic networks. **Algorithms for Molecular Biology**, BioMed Central, v. 7, p. 1–10, 5 2012. ISSN 17487188. Disponível em: https://almob.biomedcentral.com/articles/10.1186/1748-7188-7-9. Citado na página 20.

KARIM, M. R. *et al.* Convolutional embedded networks for population scale clustering and bio-ancestry inferencing. **IEEE/ACM transactions on computational biology and bioinformatics**, IEEE/ACM Trans Comput Biol Bioinform, v. 19, p. 369–382, 2022. ISSN 1557-9964. Disponível em: https://pubmed.ncbi.nlm.nih.gov/32750845/. Citado na página 25.

KATOH, K.; STANDLEY, D. M. Article fast track mafft multiple sequence alignment software version 7: Improvements in performance and usability. 2013. Disponível em: https://academic.oup.com/mbe/article/30/4/772/1073398. Citado na página 29.

KHAN, M. B. *et al.* Dengue overview: An updated systemic review. **Journal of infection and public health**, J Infect Public Health, v. 16, p. 1625–1642, 10 2023. ISSN 1876-035X. Disponível em: https://pubmed.ncbi.nlm.nih.gov/37595484/. Citado na página 17.

KIM, C. *et al.* Bwa-mem-scale: Accelerating genome sequence mapping on commodity servers. **ACM International Conference Proceeding Series**, Association for Computing Machinery, 8 2022. Disponível em: https://parsif.al/Rickson/metodos-de-tipagem-de-sequencias/conducting/studies/. Citado na página 26.

KISHINO, H.; HASEGAWA, M. Maximum likelihood. **Encyclopedia of Genetics**, Academic Press, p. 1157–1160, 1 2001. Citado na página 19.

KLUYVER, T. *et al.* Jupyter notebooks – a publishing format for reproducible computational workflows. In: LOIZIDES, F.; SCHMIDT, B. (Ed.). **Positioning and Power in Academic Publishing: Players, Agents and Agendas**. [S.l.], 2016. p. 87 – 90. Citado na página 25.

KOCKUM, I.; HUANG, J.; STRIDH, P. Overview of genotyping technologies and methods. **Current Protocols**, Wiley, v. 3, n. 4, abr. 2023. Disponível em: https://doi.org/10.1002/cpz1.727. Citado nas páginas 12 e 16.

LIMA-CAMARA, T. N. Emerging arboviruses and public health challenges in brazil. **Revista de Saúde Pública**, Faculdade de Saúde Pública da Universidade de São Paulo, v. 50, p. 36, 2016. ISSN 0034-8910. Disponível em: https://doi.org/10.1590/S1518-8787.2016050006791. Citado na página 17.

LOPES, N.; NOZAWA, C.; LINHARES, R. E. C. Características gerais e epidemiologia dos arbovírus emergentes no brasil. **Revista Pan-Amazônica de Saúde**, Instituto Evandro Chagas, v. 5, n. 3, ago. 2014. Disponível em: https://doi.org/10.5123/s2176-62232014000300007. Citado nas páginas 12 e 17.

NASCIMENTO, C. **Análise da proteína spike do SARS-CoV-2 utilizando o algoritmo CBUC.** 2021. Citado nas páginas 12, 19 e 25.

Rannala, B.; Yang, Z. Probability distribution of molecular evolutionary trees: A new method of phylogenetic inference. **Journal of Molecular Evolution**, v. 43, n. 3, p. 304–311, set. 1996. Citado na página 20.

RESTOVIC, M. I. V. Arthropod borne virus database – abvdb - um banco de dados de epidemiologia molecular para os vÍrus dengue, zika e chikungunya. 9 2018. Citado na página 21.

THAKKAR, M. Next.js. In: _____. [S.l.: s.n.], 2020. p. 93–137. ISBN 978-1-4842-5868-2. Citado na página 31.

TROPICAL doi Revista da Sociedade Brasileira de M. *et al.* An overview of zika virus genotypes and their infectivity. **Revista da Sociedade Brasileira de Medicina Tropical**, Sociedade Brasileira de Medicina Tropical - SBMT, v. 55, p. e0263–2022, 9 2022. ISSN 0037-8682. Disponível em: https://www.scielo.br/j/rsbmt/a/XyjvFRdM4bxFz3SkT4xD8xs/?lang=en. Citado na página 18.

VILSKER, M. *et al.* Genome Detective: an automated system for virus identification from high-throughput sequencing data. **Bioinformatics**, v. 35, n. 5, p. 871–873, 08 2018. ISSN 1367-4803. Disponível em: https://doi.org/10.1093/bioinformatics/bty695. Citado nas páginas 22, 29 e 30.

WUYUN, Q. *et al.* Scalable statistical introgression mapping using approximate coalescent-based inference. Association for Computing Machinery (ACM), p. 504–513, 9 2019. Citado na página 26.

YANG, Z.; RANNALA, B. Molecular phylogenetics: Principles and practice. **Nature reviews. Genetics**, v. 13, p. 303–14, 03 2012. Citado nas páginas 12, 19 e 20.

APÊNDICE A – SEQUÊNCIAS UTILIZADAS

A.1 Sequencias todos os experimentos (exceto no segundo e terceiro experimentos)

KY241759 KY241747 MF801381 KU365778 KX694533 KY241702 KX548902 KY014322 MN566107 KF268948 KY241753 KY014300 MH130098 MF783073 KU955589 KY241749 KY241765 KY241777 MF801387 MK028858 MH882533 MF574563 MH061911 KY325469 MF574573 KY765324 MF167360 KY241689 KY241697 KY075938 KY328290 KX601166 KX051560 KY785472 MH882544 KY785459 KY415990 MN185324 KY003153 MH130109 KY348640 MH061853 MN566104 KY014295 MH157213 MH130097 MH061886 KX922705 KY785429 KY075934 LC331561 MH916802 MH055376 KU509998 MF593625 KY288905 MH061860 MH675620 MK049247 KU820899 MF574558 MF574583 MH158236 MF801410 KX156774 MH061866 MK238035 KY785476 MH061872 MK238038 KY241708 KY785423 KX446951 KY241768 MF801384 MH063261 KU963573 KU761560 KY785415 MF073358 KY241736 KY558999 KY014310 MK269360 KY241701 KY785462 MG770187 KY317938 KU955594 KX262887 MK269357 KY785445 MH061859 KY328289 KY241716 KX197192 KY765321 KX601169 KY241788 MF574566 KY415988 KY241746 KY241752 KY241758 KY785456 MH882541 MG674719 KY241684 KX694534 KY241762 KU681082 MH675621 KY014296 MH061889 MH061892 KY693679 KU501215 KY241711 KY325473 MF574580 MK216713 MF098769 KY241677 MF664436 KY241715 KY785475 KY272987 KY014319 KX247646 KU527068 MH061856 MF801398 MH130106 KY241725 KY241770 KY325477 MF574553 MH061907 KY003154 MT078740 KY014323 KY014298 MH061883 MH061890 KX421194 MF801413 MF801417 MH157202 MH061865 MF988734 MF996804 MG494697 KU955593 KY785426 KX447510 KU729218 KX051561 MF434522 MN100039 KY785479 KY014312 LC369584 MF098768 MF574586 MG548661 MK049248 HQ234500 KY241679 MF801408 MN185327 KX838906 MT078739 MF098771 MK105975 KY765326 KU820897 KU720415 MF574556 KX830960 KX447509 KY241757 KY559005 MN185330 KX893855 KU963574 MH061899 KY317937 AY632535 MH130101 MH061881 KY241707 KY241741 KX842449 KY785448 MH179341 MG770188 KY241733 MK713750 MF073357 MH130094 MH061858 KU926309 KX922707 KY765320 MG758785 KY241683 KY785442 MF574575 KY241691 KY241735 KY317940 MF574561 MH130096 MN473453 KY325467 MH061873 MH061879 KY241763 KY241787 MF801391 MH061913 LC190723 KY241751 KX185891 MH882528 KY014306 MH061868 KY241728 KU647676 MH675622 KX051562 MH882531 KY631494 KU707826 MH882548 MH061855 KX447513 KY325472 EU545988 KX827268 MH130107 KX247632 KY241712 KY241726 MF801389 KU501216 MH061904 MH675628 KY241714 KY317936 MH061884 MG807647 KY325476 MH061862 KX377336

MN611472 KY014324 KY014325 KY241678 MN124091 MH061897 MH544701 KY014315
HQ234501 HQ234499 KY766069 KU955590 MF801412 MF801426 KY559007 MF574585
KY241738 KY606273 KX694532 KY441401 KY241748 KX922708 KY014303 MH882536
MH061861 MN566106 MF574562 KY785464 MH130104 KY415986 KY559021 MH061878
KF268949 KX421193 KY241694 KY241756 KY785433 KY241766 KY559015 KY693680
KJ776791 MF574570 MK028859 KX811222 KX601167 KY126351 MN190155 MF574564
MF036115 KY241760 KY241782 KY075939 MF574576 MH061910 KU853013 KY241774
MH882543 KY785469 KY765323 MK216745 KY631493 MH882545 MH061852 KX906952
KY241721 KY241750 KY989511 KX447512 KU922960 KY348860 KY325479 MT078742
MH675623 KY317939 MH061887 KY785418 KY241780 KX253996 KY014301 KY120349
KX087101 KU955595 MH513600 MH061905 MF434516 KX156775 MH061867 MF098766
MF574588 MF574582 KY379148 MK241415 KY785420 MN185325 KY967711 KY241783
KY785422 MN185332 MF574554 KY241699 KX056898 KY120353 KU729217 KY606272
MH061871 MH063262 KY014304 KY241709 KY241743 KY241745 KU937936 KY325464
MF574571 KU922923 MN577550 KY241737 MH882537 MF073359 KU926310 KX827309
KX856011 KY785453 KY241685 KX838904 KX832731 MF510857 KF383116 MH130105
KY241693 KY241779 KY415989 KY241761 MF574577 MH061915 MN473451 MH675630
MH882539 MH882540 MF384325 MF574567 KY325465 MH061877 KY241773 KF383119
KY241710 KX013000 KU681081 KY693676 MH061902 LC002520 MK216727 KY241722
MH675624 KY241690 KY241724 KY241771 MH061908 MH513599 KY075932 KX198134
KX447515 MH061857 KX702400 KU312312 KX087102 KU179098 MF801414 KU997667
MF574552 KY014318 MH061882 MF434517 KY441403 KY241676 MF434521 MH675619
KX421195 MF574587 KX198135 MH513598 KU963796 MN185328 KX156776 MH061891
MH061864 KX673530 MH675629 MF159531 KY014313 MH916806 KY241742 KY559027
MH061914 KY785450 KU365777 KX377337 KY241703 KY559013 KX369547 MF574557
MH130102 KY553111 MH061870 KY241732 KY241704 KY785441 MF574572 MH061912
JN860885 KX922704 MK028860 MH882534 KY785452 KY325468 MH061876 KY785466
MF574560 KY241680 KY241686 KY241784 KY075937 MH882527 MH882529 KY415991
KY241776 KF383118 MG827392 MF574569 KY765317 KX266255 KY765325 MN473454
KU940228 MN577544 KY241696 KY241754 KY014307 LC219720 KY241729 MF574578
KY241675 KX447514 MN566105 KX280026 MN124090 MH061854 MH061894 KY241719
MH882547 KR872956 KU501217 MH061903 MH130108 KU870645 MH013290 MH675625
MF801402 KX879603 KX766029 KY241717 KY075935 KY014317 MH061885 MN566108
KY241740 KU497555 HQ234498 KY241673 KU955591 KY014314 MH916803 MH255601
KY241730 KY014321 MF574559 MK566202 MF801406 MH061869 MH061896 MH158237
MK241417 KX117076 MF574584 MF988743 KY241731 KX813683 KY014302 KY441402
KY272991 KY241700 MH882535 KY648934 MF801395 KU761561 KY785424 MF352141
KY785414 KX446950 MH675627 KY989971 MH063264 MH130103 KY241727 KU866423
MH061875 KY415987 KX806557 KY785455 MN577543 KY241767 KY241775 KY785465

MK238037 MK713748 KY241739 KY241755 KU365780 KX601168 KY241695 MG674718 MH882542 KF383117 MF574565 MF574568 KY765318 KY765322 KU853012 KY241687 KY241785 KX922703 MK028861 MF574579 MH119185 KY241718 MF801403 MF964216 MH061888 MH061893 KY241720 MH882546 KX447511 MH368551 KY785484 KY014316 MH675626 MH061906 MH061900 MT078741 KX766028 KY927808 KY631492 KY241781 KY014297 MG548660 MF801418 KY075936 KU955592 KY785419 KY785427 KX879604 MH061895 KU321639 KY120348 MH900227 KY014299 KY014320 KY693678 DQ859059 MF574581 MN185326 KU758877 KU820898 KY241772 MK241416 KY241688 MH130100 MN185331 MF574555 MH061874 KY241734 KY014305 KY785435 KU365779 MH061880 KY241706 KX197205 MH882538 KX447517 KY241744 KX838905 KF383115 MF794971 KY241764 KY241778 KY785468 MG595216 MK269359 KU761564 KY241682 KY241786 MF801396 KY241692 KX922706 MH130099 MH130095 KY120352 MN025403 MK028857 MH882530 MH882532 KU991811 KX269878 MF574574 MN473452 KY241698 MF438286 KX520666 KU744693 MF783072 MH763833 MH061909 MF692778 KY241713 KU740184 KY241681 KY693677 MG758786 KY075933 KX447516 KX377335 KY241723 KX051563 MK696551 KY241705 MF801378 MH061901 KX830930 LC191864 MG807646 MN185329 MG645981 KY241671 MH157208 MH061863 KF268950 KY765327 MH061898

A.2 Sequências utilizadas no segundo experimento

KU647676 KU509998 KU312314 KU312313 KU729217 KU729218 KU497555 KU527068 KU501215 KU501216 KU501217 KU365777 KU365778 KU365779 KU365780 KU312312 KU321639 KJ776791 KF383115 KF383116 KF383118 KF383119 KF268948 HQ234499 HQ234500 HQ234501 JN860885 DQ859059 EU545988

A.3 Sequências utilizadas no terceiro experimento

KU647676 KU509998 KJ634273 KF993678 KU312315 KU312314 KU312313 KU646828 KU646827 KU729217 KU729218 KU497555 KU527068 KU501215 KU501216 KU501217 KU365777 KU365778 KU365779 KU365780 KU312312 KU321639 KJ776791 HQ234499 JN860885 EU545988