



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

Matheus Carreira Andrade

Construção de Redes *One-step* para Estudos Evolucionários Utilizando GPU

São José do Rio Preto
2020

Matheus Carreira Andrade

Construção de Redes *One-step* para Estudos Evolucionários Utilizando GPU

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Financiadora: CAPES

Orientador:

Prof. Dr. Geraldo Francisco Donegá
Zafalon

**São José do Rio Preto
2020**

A553c Andrade, Matheus Carreira
Construção de Redes One-step para Estudos Evolucionários
Utilizando GPU / Matheus Carreira Andrade. -- São José do Rio
Preto, 2020
66 f. : il., tabs., fotos

Dissertação (mestrado) - Universidade Estadual Paulista
(Unesp), Instituto de Biociências Letras e Ciências Exatas, São
José do Rio Preto
Orientador: Geraldo Francisco Donegá Zafalon

1. One-step. 2. Grafo. 3. Bioinformática. 4. Biossequência. 5.
Caminho Evolucionário.. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do
Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto. Dados
fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Matheus Carreira Andrade

Construção de Redes *One-step* para Estudos Evolucionários Utilizando GPU

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Financiadora: CAPES

Comissão Examinadora:

Prof. Dr. Geraldo Francisco Donegá Zafalon
UNESP – Câmpus de São José do Rio Preto
Orientador

Prof. Dr. Walter Filgueira de Azevedo Júnior
PUCRS - Escola de Ciências da Saúde e da Vida

Prof. Dr. Leandro Alves Neves
UNESP – Câmpus de São José do Rio Preto

**São José do Rio Preto
12 de Agosto de 2020**

Aos meus pais.

Agradecimentos

Agradeço ao meu pai e minha mãe pelo apoio, carinho, compreensão, amor, em todos esses anos da minha vida.

Agradeço ao meu orientador Geraldo F. D. Zafalon e doutorando Anderson Rici pelo apoio, auxílio e companheirismo, dando a mim a oportunidade de trabalhar e realizar um projeto nesta área.

Agradeço aos companheiros de laboratório, Álvaro, Guilherme F., Guilherme B., Allan, pela amizade proporcionada nestes anos de pós-graduação. Sentirei saudades de nossas conversas e risadas.

Agradeço aos professores Leandro, Rogéria, Geraldo Z., os quais foram muito importantes para minha formação.

Agradeço a Bruna de Andrade Moreira, pelo amor, carinho, apoio, nestes quase dois anos ao seu lado. Muito obrigado!

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

"O inimigo mais perigoso que você poderá encontrar será sempre você mesmo."

-Nietzsche, Friedrich (YOUNG, 2010)

Resumo

Os avanços nos estudos na área biológica resultaram na dependência de profissionais desta área em relação a recursos computacionais, devido a crescente quantidade de dados a serem analisados. A Bioinformática, como um segmento da Ciência da Computação, é uma das áreas que se utiliza de recursos matemáticos, como a teoria dos grafos. Esta pode ser empregada na construção de árvores filogenéticas, mapeamento de famílias de proteínas, entre outras aplicações. Dentre as diversas técnicas utilizadas para a construção de redes de biossequências, uma que se destaca é a estratégia *One-step*, a qual indica relações de proximidade, analisando pequenas variações entre as biossequências. Porém, ao se comparar milhares de biossequências é necessário uma abordagem eficiente, devido ao volume de cálculos necessários. Um método computacional que não utiliza somente o processamento da CPU pode ser uma solução para este problema. Assim, o presente trabalho propôs a modelagem e o desenvolvimento de um método computacional para estudos evolucionários, a partir de redes *One-step*. Este método utiliza-se de GPU para a paralelização dos cálculos realizados, a fim de diminuir o tempo de processamento ao se comparar milhares, ou até milhões, de biossequências. Tal trabalho demonstra-se importante para profissionais da área biológica, de modo a auxiliá-los no entendimento da evolução genética de diferentes indivíduos, ou seus estágios de vida, analisando as relações entre as biossequências.

Palavras-chave: *One-step*, Grafo, Bioinformática, Biossequência, Caminho Evolucionário.

Abstract

Advances in biological studies have resulted in the dependence of professionals in this area about computational resources due to the increasing amount of data to be analyzed. Bioinformatics, as a segment of Computer Science, is one of the areas that use mathematical resources, such as graph theory. This can be used in the construction of phylogenetic trees, protein family mapping, among other applications. There are several techniques used for the biosequence networks construction, such as the One-step strategy, which indicates proximity relations, analyzing small variations between biosequences. However, when comparing thousands of biosequences an efficient approach is necessary because of the large numbers of calculations required. A computational method that does not use only CPU processing can be a way for the problem solution. Thus, the present work proposed the modeling and development of a computational method for evolutionary studies, from One-step networks. This method uses GPU to parallelize the calculations, aimed to reduce the processing time when comparing a large number of biosequences. This work is important to professionals in the biological area to understand the genetic evolution of different individuals or stages, analyzing relations between biosequences.

Keywords: *One-step, Graph, Bioinformatics, Biosequences, Evolutionary Path.*

Lista de Figuras

2.1	Interligação entre as bases que compõem o DNA	20
2.2	Exemplo de haplótipo	21
4.1	Exemplo de sequências de DNA.	33
4.2	Exemplo de abstração da GPU com uma e duas dimensões de blocos e <i>threads</i>	35
4.3	Exemplo de abstração da GPU com três dimensões de blocos e <i>threads</i>	36
4.4	Diagrama da etapa de cálculo de distâncias.	39
4.5	Diagrama da etapa de construção da rede.	40
4.6	Exemplo de haplótipos iguais em arquivos diferentes.	41
4.7	Rede <i>One-step</i> construída utilizando dois arquivos.	42
5.1	Gráfico da abordagem com CPU vs abordagem com GPU.	46
5.2	Gráfico de Regressão da abordagem com CPU vs abordagem com GPU.	47
5.3	Gráfico da abordagem com CPU vs abordagem com GPU de 100 mil a 1 milhão de sequências.	50
5.4	Gráfico de regressão da abordagem com CPU vs abordagem com GPU de 100 mil a 1 milhão de sequências.	51
5.5	Gráfico da abordagem sequencial vs abordagem com GPU vs <i>Median-joining</i>	55

Lista de Tabelas

5.1	Tempos de execução para a abordagem <i>One-step</i>	45
5.2	Tempos de execução 1-5 para o caso com 60000 sequências.	47
5.3	Tempos de execução 6-10 para o caso com 60000 sequências.	48
5.4	<i>Speedup</i> de todos os casos de testes.	48
5.5	Tempos de execução para a abordagem <i>One-step</i> de 100 mil a 1 milhão de sequências.	49
5.6	<i>Speedup</i> de todos os casos de testes de 100 mil a 1 milhão.	51
5.7	Média dos tempos de execução do método <i>median-joining</i>	53
5.8	Média dos tempos de execução do método <i>median-joining</i> - Sequências com alteração randômica de um nucleotídeo.	53
5.9	Média dos tempos de execução do método <i>median-joining</i> - Sequências com alteração randômica de dois nucleotídeo.	53
5.10	Média dos tempos de execução do método <i>median-joining</i> - Sequências com alteração randômica de três nucleotídeo.	54
5.11	Média dos tempos de execução, em segundos, da abordagem com CPU vs abordagem com GPU em todos os cenários.	55
5.12	<i>Speedup</i> abordagem com CPU e GPU vs método <i>Median-joining</i> . . .	56

Sumário

1	Introdução	12
1.1	Justificativa	16
1.2	Objetivo	17
2	Fundamentação Teórica	19
2.1	Macromoléculas biológicas	19
2.2	Haplótipos	20
2.3	Reconhecimento de Padrões	21
2.4	Teoria dos grafos	21
2.5	Árvore Filogenética	23
2.6	Redes Filogenéticas	23
2.7	<i>Median-joining</i>	24
2.7.1	<i>Network</i>	25
2.8	<i>One-step</i>	26
2.9	Computação Paralela	27
3	Trabalhos Relacionados	29
3.1	Estudos sobre <i>Evolutionary Pathway</i>	29
3.2	<i>One-step Network</i>	30
4	Desenvolvimento do Trabalho	33
4.1	Cálculos de distâncias	33

4.2	Construção da rede	38
5	Resultados	44
5.1	Plataforma de testes	44
5.2	Testes e Resultados	45
5.2.1	Abordagem com CPU vs Abordagem com GPU - Primeiro caso	45
5.2.2	Abordagem com CPU vs Abordagem com GPU - Segundo caso	49
5.2.3	Método <i>median-joining</i>	52
6	Conclusão	57
6.1	Contribuições Obtidas	58
6.2	Trabalhos Futuros	58
	Referências	59

Capítulo 1

Introdução

Os avanços dos estudos na área biológica, principalmente em genômica, propiciaram evoluções consideráveis no aspecto científico e tecnológico (EDGAR; BATZOGLOU, 2006). Consequentemente, a quantidade de dados genômicos disponíveis e que devem ser analisados cresceu consideravelmente, de modo que as análises manuais se tornaram impraticáveis (RIDDER; RIDDER; REINDERS, 2013). Com isso, houve a necessidade da utilização de recursos computacionais para auxiliar os biólogos em suas análises, para que posteriores inferências possam ser realizadas sobre essa crescente quantidade de dados genômicos. Estes fatores originaram, por conseguinte, a Bioinformática.

A Bioinformática tem por pressuposto realizar análises sobre as sequências biológicas, também classificadas como biossequências. Diversas operações podem ser realizadas para se proceder estas análises, como alinhamentos de sequências (HUANG, 1994), reconhecimentos de padrões (WANG et al., 2011), construções de árvores filogenéticas (GOUY; GUINDON; GASCUEL, 2009), construção de redes de haplótipos (HUSON; SCORNAVACCA, 2011), entre outras. Uma variedade de técnicas podem ser empregadas para se concluir tais operações como, por exemplo, algoritmos de programação dinâmica (ZHANG et al., 2004), algoritmos evolutivos (VESTERSTROM; THOMSEN, 2004), algoritmos bioinspirados (GANDOMI; ALAVI, 2012) e algorit-

mos baseados em teoria dos grafos (HARTUV; SHAMIR, 2000).

A análise filogenética visa descobrir as relações evolutivas entre diferentes espécies ou ancestrais, a fim de entender a evolução da vida na terra (HUSON; SCORNAVACCA, 2011). Árvores filogenéticas são amplamente utilizadas para resolver esta tarefa, de modo a representar histórias evolutivas nas quais os principais eventos são especiações¹ e descendentes com modificações.

No entanto, devido ao padrão bifurcado da árvore (KONG; SÁNCHEZ-PACHECO; MURPHY, 2016), cenários evolutivos complexos são mal descritos por árvores filogenéticas (HUSON; BRYANT, 2005), como mecanismos de evolução reticulada (SNEATH, 1975; SYVANEN, 1985), entre estes, hibridização (RIESEBERG, 1997), recombinação horizontal² (DELWICHE; PALMER, 1996; GRIFFITHS; MARJORAM, 1996) ou rearranjo genético (DELWICHE; PALMER, 1996). Além disso, mecanismos como a separação de linhagens incompletas ou padrões complicados de duplicação e perda de genes podem levar a incompatibilidades que não podem ser representadas em uma árvore (HUSON; SCORNAVACCA, 2011).

Com isso, o uso de redes filogenéticas, em contraste à árvore, tornou-se uma alternativa favorável neste aspecto. Assim, é possível visualizar variações entre sequências genéticas de organismos (TEACHER; GRIFFITHS, 2011), analisando conjuntos de dados, os quais a evolução envolve quantidades significativas de eventos reticulados. Dentre as redes filogenéticas, as não enraizadas são as mais utilizadas, devido a dificuldade nos cálculos e falta de usabilidade e otimização de métodos computacionais para trabalhar com redes enraizadas (HUSON; SCORNAVACCA, 2011; HUSON; LINZ, 2018). No que se refere a redes não enraizadas, tem-se, por exemplo, redes de haplótipos, na qual os nós representam diferentes haplótipos, geralmente relacionados, e as arestas conectam os que são muito semelhantes (MATSCHINER, 2015). Esta rede é utilizada em estudos evolucionários, possibilitando a análise na variação genética

¹Termo usado para referir-se à divisão de uma linhagem que produz duas ou mais espécies diferentes.

²Evento evolutivo que ocorre quando há transferência de material genético entre organismos de espécies diferentes.

entre organismos da mesma ou diferentes espécies (HUANG et al., 2017; NISAR et al., 2019). A partir disso, ressalta-se que a análise na variação genética anteriormente citada também é aplicada em estudos virológicos (FAZ et al., 2019; LADNER et al., 2015).

O método *median-joining* é o que recebe maior destaque (LEIGH; BRYANT, 2015; KONG; SÁNCHEZ-PACHECO; MURPHY, 2016) para a construção de redes de haplótipos. Este método constrói, inicialmente, árvores de geração mínima, e seleciona as que possuem menor peso. Posteriormente, adiciona-se novas sequências não existentes na rede, as quais não estão no conjunto inicial, denominadas de vetores medianos (BANDELT; FORSTER; RÖHL, 1999). Estes vetores são considerados um ancestral extinto ou sequências não amostradas no sequenciamento do material genético. No entanto, a desvantagem deste método está na própria adição destes vetores, os quais influenciam na organização da rede e não representam uma característica autêntica do organismo, o que prejudica a qualidade da visualização (TRONTELJ; MACHINO; SKET, 2005; ANDRESIUK et al., 2013; KONG, 2015).

Além disso, nos trabalhos nos quais o método *median-joining* (VULTOS et al., 2008; WOOD; GRAVE; DANIELS, 2018; XUE et al., 2018; BUJÁN et al., 2018; HAIDER et al., 2018) é utilizado, aplica-se um baixo número de sequências para a construção da rede, poucas dezenas ou centenas, os quais não representam um cenário atual, com milhares de sequências. Este cenário é possível, devido aos sequenciadores de nova geração (NGS) (SHENDURE; JI, 2008; GOODWIN; MCPHERSON; MCCOMBIE, 2016), que geram até milhões de sequências. Diante disso, uma alternativa para sanar problemas de desempenho que, possivelmente, limitam o número de sequências utilizados nestes trabalhos, dada a natureza do *median-joining*, é a utilização de outro método baseado em distância. Um método que pode ser utilizado para a construção de redes de haplótipos é o TCS (CLEMENT; POSADA; CRANDALL, 2000), o qual é um método probabilístico. Porém, ao se trabalhar com centenas ou mi-

lhares de sequências, há perda de desempenho, sendo esta agravada conforme se trabalha com sequências mais divergentes (maior distância entre as sequências) (SALZBURGER; EWING; HAESELER, 2011). Assim, para sanar os problemas relatados sobre os métodos *median-joining* e TCS, neste trabalho é utilizado o método *One-step* (CAMPO et al., 2014; PALMER et al., 2016) para estudos evolucionários.

O *One-step* foi utilizado por autores, como Campo et al. (2014), na análise da evolução do vírus HCV (Hepatite C). O autor relata que esta estratégia fornece uma estrutura natural para exibir a relação entre populações heterogêneas, ao analisar-se pequenas variações nas biossequências. Não obstante, o método *One-step* conecta sequências estreitamente relacionadas sem qualquer suposição sobre sua origem ancestral (CAMPO et al., 2014), o qual não ocorre com o método *median-joining*. Além disso, de forma a potencializar ganhos no desempenho ao se comparar milhares de sequências, e assim, retornar uma resposta mais rápida ao usuário, pode-se utilizar o processamento de uma GPU (Unidade Gráfica de Processamento) para computar os cálculos de distância necessários para a construção da rede *One-step*.

Como relatado por Nvidia (NVIDIA, 2019b), a computação acelerada por GPU libera porções do aplicativo com uso intenso de computação para a placa de vídeo, enquanto o restante do código ainda é executado na CPU (Unidade Central de Processamento). Portanto, a aplicação é executada em um menor tempo, uma vez que a GPU possui milhares de núcleos eficientes criados para lidar com múltiplas tarefas simultaneamente. Assim, o presente trabalho destaca-se pelo desenvolvimento do método construção de redes *One-step* utilizando-se GPU para a paralelização dos cálculos realizados. Com isso, reduziu-se de forma considerável o tempo de processamento ao se comparar um considerável número de biossequências, na escala de milhares ou milhões.

Este trabalho está organizado como segue: no Capítulo 1 apresenta-se a problemática desta pesquisa, além de sua motivação, escopo, objetivos e justificativa; quanto

ao Capítulo 2, fundamenta-se os elementos sob pesquisa, desde a contextualização biológica aos aspectos computacionais que tangem este trabalho; no Capítulo 3, tem-se a descrição de trabalhos correlatos à linha de pesquisa sob o âmbito de redes de haplótipos; ademais, no Capítulo 4, apresenta-se a metodologia adotada para atingir os objetivos propostos neste trabalho; por conseguinte, no Capítulo 5, os resultados de pesquisa são apresentados e discutidos; neste caminho, no Capítulo 6, encontra-se o fechamento deste trabalho com a conclusão da pesquisa realizada e, por fim, os artigos produzidos acerca do tema.

1.1 Justificativa

A justificativa se dá pelo fato de que uma abordagem utilizando GPU para o processamento de cálculos intensos é importante, devido a velocidade que os dados são computados utilizando uma arquitetura massivamente paralela (NVIDIA, 2019b). O uso de GPU se tornou comum na otimização de aplicações, devido ao seu poder computacional e baixo custo de *hardware* (LANGDON et al., 2017). Shehab et al. (2017) compararam o desempenho de alinhamentos múltiplos de sequências, utilizando os algoritmos predecessores de alinhamento de sequências par-a-par, Needleman-Wunsch e Smith-Waterman. Estes foram implementados apenas utilizando a CPU e em uma abordagem híbrida, a qual utiliza CPU e GPU para a execução do algoritmo. Portanto, Shehab et al. (2017) concluiu que a utilização de GPU em conjunto com a CPU na realização dos cálculos, a execução dos algoritmos foi 100 vezes mais rápida que a abordagem sequencial utilizando apenas CPU. Dessa forma, destaca-se a importância de um método computacional massivamente paralelo na área da bioinformática, devido a quantidade de dados de sequenciamentos gerados pelas tecnologias NGS.

Ademais, a utilização da estratégia *One-step* para estudos evolucionária é relevante, devido a disposição de sua rede. Como relatado por Campo et al. (CAMPO et

al., 2014), o *One-step* quando utilizado para conectar variantes virais³ fornece uma estrutura natural para organizar estas sequências em uma rede, a qual agrupa os haplótipos que possuem ligações evolutivas mais próximas (PALMER et al., 2016). Portanto, a conexão de sequências utilizando a abordagem *One-step* é vantajoso pois provê um critério biológico simples para identificar a estrutura genética de subpopulações virais, conectando variantes estreitamente relacionadas sem qualquer suposição sobre sua origem ancestral (CAMPO et al., 2014). Esta suposição é uma desvantagem do método *median-joining*, o qual adiciona possíveis ancestrais a rede (TRONTELJ; MACHINO; SKET, 2005; ANDRESIUK et al., 2013; KONG, 2015).

Dessa forma, o método proposto no presente trabalho, além do valor biológico que a estratégia *One-step* trás ao relacionar biossequências com pequenas variações (CAMPO et al., 2014), também foca em desempenho, possibilitando assim o biólogo dispendar seu tempo para a análise dos resultados encontrados e não na espera para obter estes resultados. Este método computacional para construção de redes *One-step* é de importância em estudos evolucionários, o qual auxiliará biólogos e profissionais da área médica a compreender o comportamento de diferentes organismos e sua evolução.

1.2 Objetivo

Este trabalho tem como objetivos:

- A modelagem de um método computacional que construirá redes *One-step*. Esta construção tem como pressuposto auxiliar biólogos e profissionais da área médica a compreender a evolução de um ou mais indivíduos, no decorrer do tempo, por meio da análise das variações entre suas biossequências ao exibir os resultados em uma rede;
- A criação de um método computacional paralelizável utilizando plataforma Cuda,

³Sequências virais que possuem variações entre elas, as quais podem causar consequências, como a resistência do vírus a uma determinada droga(s) farmacêutica(s).

comparando milhares de biossequências em paralelo, e conseqüentemente, diminuir o tempo de processamento necessário para a construção da rede;

Capítulo 2

Fundamentação Teórica

Esta seção tem o objetivo de apresentar conceitos fundamentais na área de Bioinformática, desde fundamentos básicos da biologia molecular e grafos, até estratégias e utilizações destes conceitos na resolução de problemas. A compreensão destes assuntos é importante para o entendimento dos trabalhos relacionados que envolvem o assunto aqui tratado.

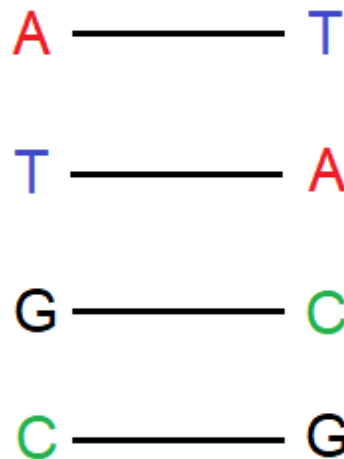
2.1 Macromoléculas biológicas

O entendimento das macromoléculas de DNA, RNA e também das proteínas é indispensável para a compreensão das estruturas e funções de cada organismo. No escopo da Bioinformática, estas macromoléculas são chamadas de biossequências ou sequências biológicas (EIDHAMMER; JONASSEN; TAYLOR, 2000; LEMOS; ARAGAO; CASANOVA, 2003).

Assim, tanto as sequências de DNA como as de RNA são classificadas como sequências de nucleotídeos. O DNA é a sigla para ácido desoxirribonucleico, que é um composto orgânico cujas moléculas contêm as instruções genéticas que coordenam o desenvolvimento e funcionamento de todos os seres vivos. Segundo estudos, uma sequência de DNA é composta por uma fita dupla de nucleotídeos e quatro bases

nitrogenadas: Adenina (A), Timina (T), Citosina (C) ou Guanina (G), que se ligam por meio de uma ponte de hidrogênio, como representado na Figura 2.1 (WATSON; CRICK, 1953).

Figura 2.1: Interligação entre as bases que compõem o DNA



Fonte: Figura Elaborada pelo autor.

2.2 Haplótipos

Um haplótipo, é definido como um grupo de genes dentro de um organismo que foram herdados de um único pai com mais frequência do que o esperado (DURBIN, 2014). Ao examinar os haplótipos, os cientistas podem identificar padrões de variação genética associados à saúde e estados de doença. Por exemplo, se um haplótipo estiver associado a uma certa doença, então os cientistas podem examinar extensões de DNA próximas a uma região onde ocorre variações, também conhecidas como *Single Nucleotide Polymorphism* (SNP), para tentar identificar o gene ou os genes responsáveis pela causa da doença (NATURE, 2017). Um exemplo de haplótipo é exibido na Figura 2.2.

Assim, entende-se por um haplótipo uma mesma sequência de DNA que está presente em um sequenciamento com uma determinada frequência. Como pode ser ob-

Figura 2.2: Exemplo de haplótipo

P1-3 Freq 5
ACCCGTTCA

Fonte: Figura Elaborada pelo autor.

servado na Figura 2.2, tem-se uma sequência P1-3 que ocorre 5 vezes (Freq 5) no sequenciamento de uma certa amostra estudada.

2.3 Reconhecimento de Padrões

Com os avanços dos estudos e tecnologias nas áreas da biologia molecular e celular, possibilitou-se o entendimento de uma série de fatores e características que são propagadas pelos diversos seres vivos ao longo das gerações. Neste cenário, a filogenia e a análise de padrões são áreas que se preocupam em identificar, respectivamente, o caminho de um processo evolucionário e as características que foram predominantemente preservadas pelos organismos ao longo das gerações (GOULD, 1977). Diversas abordagens algorítmicas foram desenvolvidas no decorrer dos anos, com uma ampla diversidade de aplicações, comparando-se biossequências e identificando variações nos organismos (HUANG et al., 2017).

2.4 Teoria dos grafos

A Bioinformática, como um segmento da Ciência da Computação, é uma das áreas que se beneficia dos recursos oferecido pela teoria dos grafos. Pode-se aplicar grafos para a construção de árvores filogenéticas (DEYASI; BANERJEE; DEB, 2015), para o mapeamento de famílias de proteínas (MITCHELL et al., 2014), ou para a análise e visualização das sequências de DNA, como versa a ideia do presente trabalho.

A teoria dos grafos é um conceito matemático que estuda as relações e interações

entre objetos de um determinado conjunto. Tal teoria pode-se dizer que foi redescoberta em vários momentos da história, devido a remota comunicação antes do surgimento da rede de computadores e telefonia. O primeiro registro de um problema relacionado a teoria dos grafos foi em 1736, conhecido como problema de Euler (WEST, 2001), vindo posteriormente o problema de Kirchhoff (FIGUEIREDO, 2013), Cayley (FERGUSON, 1989), entre outros problemas (READ, 2014).

Um grafo G é definido como um par V e A (READ, 2014; BIGGS; LLOYD; WILSON, 1976), em que:

- V é um conjunto de vértices (ou nós), em que cada vértice é representado por um círculo;
- A é um conjunto de arestas (ou arcos), em que cada aresta é representada por um arco conectando dois vértices.
- O grau de um vértice é definido como o número de arestas que o incidem;
- Uma cadeia é uma sequência qualquer de arestas adjacentes que ligam dois vértices;
- Um grafo $G(V,A)$ é dito conexo se há pelo menos uma cadeia ligando cada par de vértice deste grafo G ;
- Um grafo $G(V,A)$ é dito desconexo se há pelo menos um par de vértice que não está ligado por nenhuma cadeia;
- Um subgrafo de um grafo G é um grafo cujo conjunto dos vértices é um subconjunto do conjunto de vértices de G , e o conjunto de arestas é um subconjunto do conjunto de arestas de G .

2.5 Árvore Filogenética

Como relatado por Bear et al. (2016), uma árvore filogenética é um diagrama que representa as relações evolutivas entre os organismos. Estas são hipóteses, não são fatos definitivos. O padrão de ramificação em uma árvore filogenética reflete como espécies ou outros grupos evoluíram a partir de uma série de ancestrais comuns (BAUM, 2008). Nas árvores, duas espécies são mais relacionadas se tiverem um ancestral comum mais recente e menos relacionadas se tiverem um ancestral comum menos recente. Por definição, as árvores filogenéticas são adequadas para representar histórias evolutivas nas quais os principais eventos são especiações e descendentes com modificação.

Não obstante, as árvores filogenéticas são menos apropriadas para modelar mecanismos de evolução reticulada, entre estes, transferência horizontal de genes, hibridização, recombinação genética (*crossing-over*) ou rearranjo (SNEATH, 1975; ZHU et al., 2019; WEN et al., 2016). Ademais, mecanismos, como separação de linhagens incompletas ou padrões complicados de duplicação e perda de genes, podem levar a incompatibilidades que não podem ser representadas em uma árvore (HUSON; SCORNAVACCA, 2011). A partir destes problemas, o uso de rede filogenéticas tornou-se uma alternativa favorável nesta vertente.

2.6 Redes Filogenéticas

Redes filogenéticas provem uma estrutura mais natural para analisar evoluções reticuladas (WEN et al., 2016; ZHU et al., 2019). Uma rede filogenética pode ser definida como qualquer grafo que representa relações evolucionárias entre espécies ou organismos (HUSON; SCORNAVACCA, 2011; HUSON; RUPP; SCORNAVACCA, 2010). Estas podem ser divididas em redes filogenéticas enraizadas e não enraizadas. Como descrito pelo autor Huson em seus trabalhos (HUSON; LINZ, 2018; HUSON;

SCORNAVACCA, 2011), embora as redes filogenéticas enraizadas possam, em teoria, ser usadas para descrever explicitamente a evolução na presença de eventos reticulados, seu cálculo é difícil, e consequentemente, há falta de métodos apropriados para computá-las.

Dessa forma, o uso de redes filogenéticas não enraizadas é predominante. Dentre estas, contém redes de haplótipos, a qual é utilizada em estudos evolucionários, possibilitando a análise de variações genética entre organismos. Na rede de haplótipos, cada vértice representa um haplótipo, e as arestas conectam os que são semelhantes (MATSCHINER, 2015), onde estas conexões são definidas pelo método utilizado na análise. O método *median-joining* é o que recebe maior destaque na literatura para a construção de redes de haplótipos (LEIGH; BRYANT, 2015; KONG; SÁNCHEZ-PACHECO; MURPHY, 2016).

2.7 *Median-joining*

O método *median-joining* foi proposto por Bandelt et al. (1999). O principal *software* que implementa este método é o *Network* (ENGINEERING, 2017a), desenvolvido pela *Fluxus Engineering*¹. O método *median-joining* para construção da rede de haplótipos baseia-se no critério de redes com mínima abrangência (*minimum spanning network*) e utiliza *median vectors* (mv) para conectar haplótipos através do método de máxima parcimônia² (SOBER, 1983; FARRIS, 1970). Assim, de forma a diminuir o peso total da rede, o *median-joining* adiciona vetores medianos a rede (BANDELT; FORSTER; RÖHL, 1999), os quais não existem no conjunto de sequências original. Bandelt et al. (1999) indica que estes vetores correspondem há sequências que podem ser biologicamente interpretadas como possíveis sequências existentes e não amostradas ou como sequências de um ancestral extinto.

¹<http://www.fluxus-engineering.com/>

²Este método busca escolher a árvore ou rede que explique a evolução entre as sequências com o menor número possível de mutações.

No entanto, a desvantagem do método está na própria adição desses vetores, uma vez que estes são sequências geradas, as quais podem não representar uma característica autêntica do organismo que está sob análise (KONG; SÁNCHEZ-PACHECO; MURPHY, 2016). Consequentemente, estes vetores influenciam na construção da rede, isto é, prejudicam a clareza da visualização, visto que há diferentes elementos que não estão diretamente associados com a análise (TRONTELJ; MACHINO; SKET, 2005; ANDRESIUK et al., 2013; KONG, 2015). Ressalta-se que esta visualização é cada vez mais dificultada conforme cresce o número de sequências utilizadas na análise.

Não obstante, observa-se nos trabalhos os quais o *median-joining* é utilizado (HUANG et al., 2017; NISAR et al., 2019; HAPUARACHCHI et al., 2010; KOVALEV; MUKHACHEVA, 2017; ROSVOLD et al., 2012), que o número de sequências utilizadas não ultrapassavam poucas dezenas ou centenas. Este cenário não simula experimentos biológicos mais robustos, com milhares de sequências, os quais são factíveis, devidos as novas tecnologias de sequenciamento que geram até milhões de sequência (PATEL; JAIN, 2012). A partir disso, uma possibilidade para contornar problemas de desempenho que possivelmente limitaram o número de sequências utilizados nestes trabalhos, além dos problemas de visualização, pode-se utilizar outro método baseado em distância, como o *One-step* (CAMPO et al., 2014).

2.7.1 *Network*

O *software Network* (ENGINEERING, 2017a) é usado para construir redes e árvores filogenéticas, inferir tipos de ancestrais, tipos potenciais, ramificações evolutivas. Este *software* inclui um editor de dados, em que é possível visualizar e modificar as sequências selecionadas como entrada, ou adicionar novas, e uma janela gráfica, sendo possível visualizar a rede gerada com aquela entrada.

Nesta ferramenta é possível utilizar dois métodos para a construção da árvore:

median-joining ou *reduced median*. Cada vértice da rede representa um haplótipo, e pode ser dividido de acordo com sua frequência, como descrito em (MATSCHINER, 2015), caso seja haja haplótipos iguais nos arquivos dados como entrada.

Para a construção da rede são necessárias várias etapas, como pode ser visto em (ENGINEERING, 2017b), sendo estas: Preparação dos dados/Alinhamento de DNA, Cálculo da rede, Opções MP (algumas opções, como alterar ou deletar conexões entre sequências, entre outras), Desenhar rede. Cada etapa produz uma saída e essa é utilizada como entrada na próxima etapa.

Esse programa suporta várias formatos de entrada, incluindo entre eles o formato RDF, binário, FASTA, entre outros. A desvantagem da ferramenta é que várias opções e a utilização de certos formatos de arquivo, como o FASTA, estão disponíveis apenas em sua versão comercial. Ademais, este *software* trabalha no máximo com 30000 sequências, o qual é um limitante, devido aos sequenciadores de nova geração, que geram milhões de sequência.

2.8 *One-step*

O *One-step* (CAMPO et al., 2014) (um passo de distância) é uma estratégia para construção de redes de biossequências, em que cada biossequência é representada como um vértice no grafo. Esta estratégia se baseia na distância hamming, a qual calcula o número de posições divergentes entre duas cadeias de caracteres de mesmo tamanho (*string*). Portanto, a distância entre duas biossequências é dada pelo número de nucleotídeos distintos, em uma mesma posição. Dessa forma, a abordagem utilizada conecta na rede construída as biossequências que possuem valor 1 de distância hamming.

Assim, este método possibilita uma melhor visualização e análise do conjunto de sequências, o qual não constrói vetores medianos, em comparação ao *median-joining* (BANDELT; FORSTER; RÖHL, 1999; BANDELT et al., 1995). O *One-step* destaca

pequenas modificações nas biossequências, sendo possível associar estas com um determinado estado ou condição do indivíduo estudado. Como relatado por Campo et al. (2014), redes *One-step* fornecem uma estrutura natural para exibir a relação entre populações heterogêneas, analisando pequenas variações nas biossequências. As conexões entre os haplótipos que diferem em uma mutação que é vantajoso pois provê um critério biológico simples para identificar a estrutura genética de subpopulações virais, conectando variantes estreitamente relacionadas sem qualquer suposição sobre sua origem ancestral (CAMPO et al., 2014). Este método é aplicado em estudos evolucionários, por exemplo no âmbito da virologia, como descrito em (PALMER et al., 2016; SALEEM et al., 2018; GANOVA-RAEVA et al., 2019), considerando diferentes indivíduos ou estágios de um tratamento.

2.9 Computação Paralela

A computação paralela é uma forma de computação em que vários cálculos são realizados ao mesmo tempo, operando sob o princípio de que grandes problemas podem ser divididos em problemas menores, que então serão resolvidos concorrentemente. Uma das abordagens existentes é a *multithread* (LEWIS; BERG, 1996), em que se utiliza um ou mais *threads* (subprocessos) nos núcleos existentes na CPU para executarem pedaços do código em paralelo. Porém, esta estratégia é limitada pelo número de núcleos e a sua velocidade, uma vez que cada núcleo da CPU possuem vários outros processos do sistema operacional a serem executados e uma fila de prioridades. Dessa forma, para melhorar o desempenho do algoritmo deste trabalho propõe-se uma abordagem que se utiliza de GPU ao invés de CPU para a paralelização do algoritmo.

Vários trabalhos utilizam-se de GPU para melhorar o desempenho dos algoritmos (LEE; EIGENMANN, 2010), repassando os cálculos intensos, como diversas comparações em *loops*, para serem executados nos milhares de núcleos da placa gráfica do sistema. Para tal, a parte paralelizável do algoritmo deve ser programado em

linguagem CUDA³. Esta linguagem possui suporte para a paralelização de funções, executando-as N em N *threads* paralelas (uma *thread* por núcleo da GPU). Assim, esta abordagem objetiva melhorar o tempo de execução do algoritmo proposto no presente trabalho, em que são realizadas milhões de comparações entre biossequências, sendo que cada uma é realizada em um núcleo da GPU.

³Esta é uma linguagem de programação destinada a computação paralela em GPUs, criada pela Nvidia (<https://developer.nvidia.com/cuda-zone>)

Capítulo 3

Trabalhos Relacionados

Nas Subseções 3.1 e 3.2 são descritos trabalhos relacionados com o presente trabalho, tanto no âmbito de estudos evolucionários a partir de redes de haplótipos, quanto na construção de redes *One-step*. Destaca-se que para a confecção dos trabalhos relacionados, não foi encontrado nenhum método ou ferramenta que usufrui de GPU para os cálculos intensos de comparações entre várias biossequências, tanto na abordagem *One-step* quanto no âmbito de estudos evolucionários.

3.1 Estudos sobre *Evolutionary Pathway*

Huang discute em (HUANG et al., 2017) sobre o enterovírus humano A71 (EV-A71), o qual é um dos principais agentes causadores de doença na mão, pé e boca (HFMD), podendo também causar manifestações neurológicas graves. No entanto, o mecanismo pelo qual EV-A71 evolui dentro do corpo humano não é claro. Desta forma, os autores realizaram um estudo com o objetivo de traçar o caminho evolutivo do mesmo, para tentar entender quais são as vias de disseminação do EV-A71 e descobrir como este evolui de forma a atingir o sistema nervoso do paciente. Assim, coletou-se amostras de vários tecidos de uma criança com doença fatal, com o intuito de analisá-los.

Foram efetuados a reconstrução dos haplótipos dos tecidos e suas respectivas frequências. Estes haplótipos foram comparados, com o objetivo de identificar possíveis variantes (variações entre as sequências) entre eles, e entender a influência dessas variantes nas vias de disseminação do EV-A71, e sua evolução para o resto do organismo. Para a construção da rede, foi utilizado o método *median-joining* (BANDELT; FORSTER; RÖHL, 1999).

Como resultado, os autores conseguiram encontrar um possível caminho evolucionário, analisando os haplótipos dominantes (maior frequência) e variantes dos haplótipos com menor frequência. Foi concluído que estes possuíam uma grande influência na disseminação do EV-A71 para outros tecidos do organismo, atingido diretamente o sistema nervoso central do hospedeiro.

Outros trabalhos presentes na literatura também utilizam o *median-joining* em estudos evolutivos, como pode ser visto no trabalhos dos autores Xue et al. (XUE et al., 2018), Wood et al. (WOOD; GRAVE; DANIELS, 2018), Chong et al. (CHONG et al., 2018), entre outros. No entanto, este método, por adicionar vetores medianos a rede, dificulta a visualização das relações entre haplótipos. Andresiuk et. al (2013) em seu trabalho sobre a doença equinococose cística, relata que a rede construída pelo método Median-joining, para identificar genótipos e relações filogenéticas, foi extremamente confusa devido a várias reticulações e vetores medianos existentes. Assim, deve-se ressaltar que a visualização da rede é cada vez mais dificultada conforme aumenta o número de sequências envolvidas, pois consequentemente são criados um maior número de vetores medianos na rede.

3.2 *One-step Network*

Campo et al. (2014) utilizou redes *One-step* em seu trabalho para estudar as pequenas modificações entre diversas variantes do vírus da Hepatite C (HCV) e sua influência em pacientes infectados. O HCV muda continuamente durante a infecção crônica,

mostrando uma dinâmica complexa de subpopulações virais nos hospedeiros. Contudo, avanços recentes nos métodos de sequenciamento de nova geração (NGS) permitiram a análise em um maior número de variantes, sem precedentes, de sequências do vírus HCV em pacientes infectados, apresentando uma nova oportunidade para entender a evolução deste no hospedeiro, sua resistência a drogas e escape imunológico. Portanto, para este fim, é calculado a distância *hamming* par-a-par, comparando-se todas as variantes de sequências do vírus HCV. Posteriormente constrói-se uma rede utilizando a abordagem *One-step*, onde cada sequência é um vértice da rede, para a análise das modificações entre as variantes.

Da mesma forma, como realizado por Campo et al. (2014), Palmer et al. (2016) e Saalem et al. (2018) também discutem em seus trabalhos a construção de redes *One-step* com o objetivo de analisar a evolução das variantes do vírus HCV. No entanto, não há um método computacional consolidado para a construção de redes *One-step*. Como visto no trabalho de Palmer et al. (PALMER et al., 2016), é utilizado pelo menos dois *softwares* para realizar a computação necessária e exibição dos dados, as quais são o Mega6 e MATLAB R2014b. No entanto, a utilização do MATLAB R2014b não é adequada nestas análises, onde pode haver milhares de sequências, uma vez que este *software* possui seu desempenho degradado (MUTNEJA; SINGH, 2017; KENSICHER et al., 2017; POND; SUTTON, 2015), principalmente ao implementar laços de repetição (*loops*) (LI; CHEN; WANG, 2017; KAPOURANI, 2015). Não obstante, o *software* Mega6 foi utilizado apenas para o cálculo de distância entre sequências, pois neste não é possível a construção de redes *One-step* (TAMURA et al., 2013; MEGA, 2019).

Além disso, a utilização de mais de um *software* para a construção da rede gera uma maior dependência do fator humano para realização dos cálculos. Também deve-se ressaltar que ao se trabalhar com N sequências, é necessário um método eficiente para realizar a comparação das sequências, uma vez que o valor de N pode escalar

a casa de milhares. Desta forma, o método computacional proposto no presente trabalho constrói redes *One-step* sem a necessidade de intervenção humana no meio do processo, além de usufruir da arquitetura GPU para otimização dos cálculos de distância.

Capítulo 4

Desenvolvimento do Trabalho

O método proposto foi dividido em duas partes: cálculos de distância e construção da rede. A primeira parte é responsável por calcular os parâmetros necessários na construção da rede, como a distância entre as sequências, analisando quais são iguais ou devem estar conectadas. Na segunda parte do método, a partir dos cálculos realizados anteriormente, constrói-se a rede *One-step*.

4.1 Cálculos de distâncias

O cálculo de distância entre biossequências baseia-se na distância *hamming*, em que esta é o número de posições divergentes entre duas cadeias de caracteres de mesmo tamanho (*string*). Portanto, a distância entre duas biossequências é dado pelo número de nucleotídeos distintos, em uma mesma posição. Como exemplo, na Figura 4.1, observa-se duas sequências de DNA com distância *hamming* de valor 1.

Figura 4.1: Exemplo de sequências de DNA.



Seq 1 - A T G **G** C C C A A T A C G T
Seq 2 - A T G **C** C C C A A T A C G T

O diagrama mostra duas sequências de DNA alinhadas horizontalmente. A primeira sequência, Seq 1, é composta pelos caracteres A, T, G, G, C, C, C, A, A, T, A, C, G, T. A segunda sequência, Seq 2, é composta pelos caracteres A, T, G, C, C, C, C, A, A, T, A, C, G, T. Os caracteres A, T, C, G são coloridos em azul, enquanto os caracteres A, A, T, A, C, G, T são coloridos em verde. Um retângulo vermelho realça a diferença entre o quarto caractere das duas sequências: 'G' na Seq 1 e 'C' na Seq 2.

Fonte: Figura Elaborada pelo autor.

Esta parte do método utiliza o processamento da GPU, em conjunto com a CPU, com o objetivo de obter melhor desempenho. Como relatado pela Nvidia (NVIDIA, 2018), a arquitetura da GPU trabalha com blocos de processamento, onde cada bloco pode executar um número limite de *threads*, por exemplo: 256, 512, 1024 ou 2048 *threads*. Este número limite é definido de acordo com versão da placa gráfica utilizada (NVIDIA, 2019a).

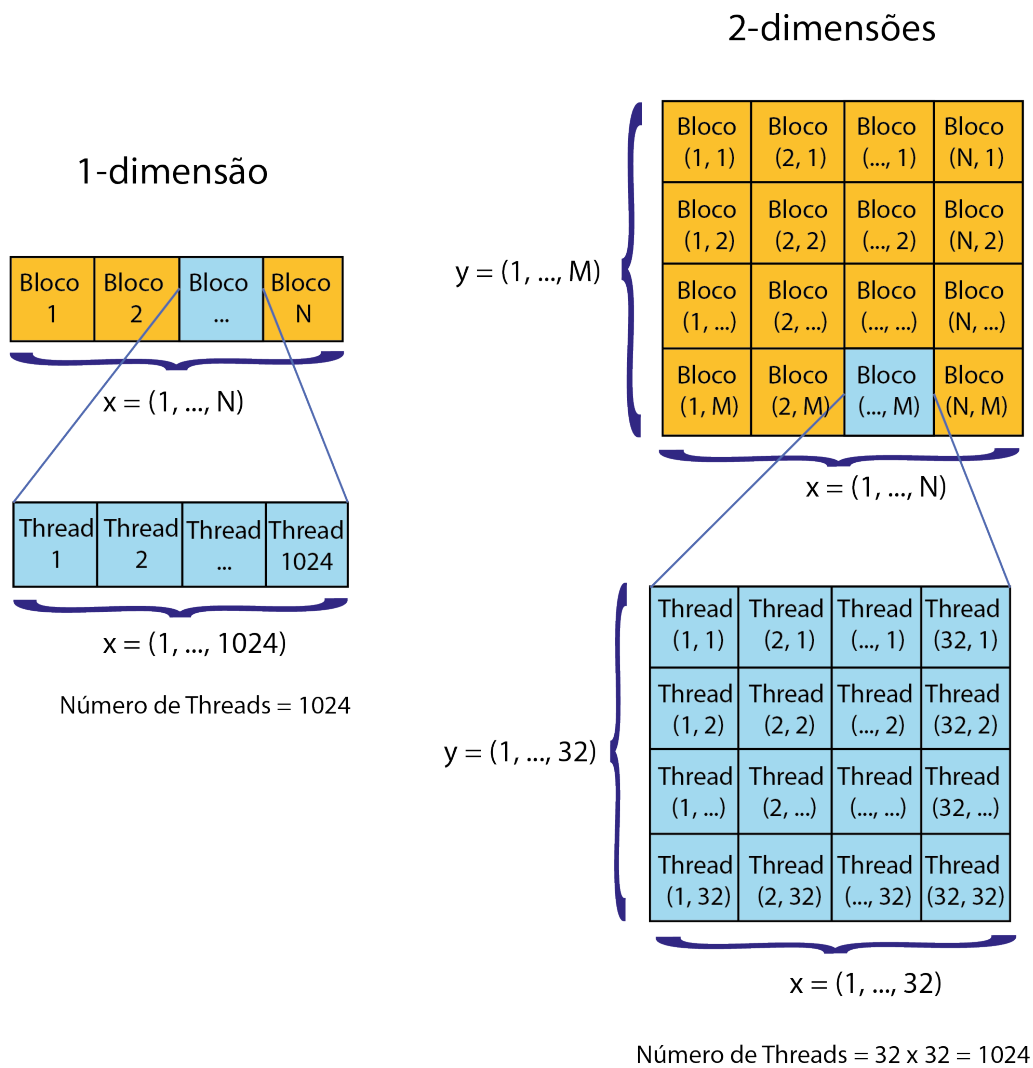
Nesta arquitetura, pode-se abstrair blocos em até três dimensões (x,y,z), onde o número limite de blocos para cada dimensão é indicado pela Nvidia (NVIDIA, 2019a). Adicionalmente, também pode-se abstrair as *threads* internas de cada bloco em até três dimensões, porém, o número de total de *threads* dentro do bloco não pode ultrapassar o limite suportado pela placa gráfica utilizada. Nas Figuras 4.2 e 4.3 observa-se exemplos de abstrações até três dimensões utilizando 1024 *threads* por bloco.

Contudo, apesar do considerável número de blocos que pode ser utilizado em uma execução, é possível obter um índice único para cada *thread*, de acordo com as Equações 4.1, 4.2 e 4.3. Os índices y e z são usados apenas caso seja utilizado a abstração de duas ou três dimensões, os quais identificam cada *thread* com um índice único em uma matriz ou paralelepípedo.

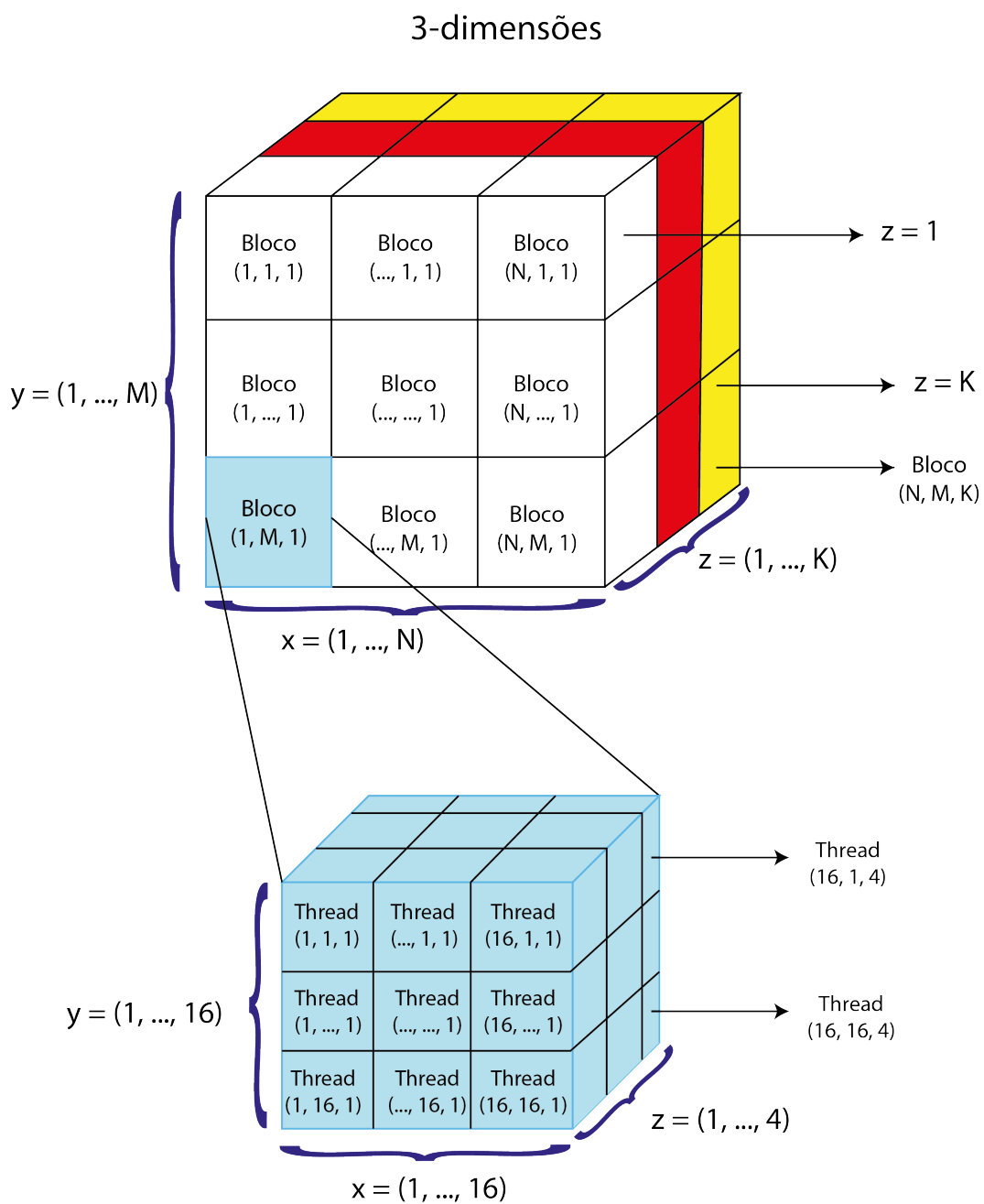
A variável *threadIdx* contém o índice da *thread* dentro do bloco, para as dimensões x (*threadIdx.x*), y (*threadIdx.y*) e z (*threadIdx.z*). No entanto, as variáveis *blockIdx* e *blockDim* representam respectivamente o Id único do bloco dentro da *Grid* e o número de *threads*, também para as dimensões x (*blockIdx.x* e *blockDim.x*), y (*blockIdx.y* e *blockDim.y*) e z (*blockIdx.z* e *blockDim.z*). Logo, com estas variáveis consegue-se identificar uma *thread* de forma única, ou seja, cada *thread* possui uma identificação única dentre todas as *threads*. Ressalta-se que estas dimensões são apenas abstrações, não interferindo no desempenho da placa gráfica ao executar o algoritmo.

$$indice_x = blockIdx.x * blockDim.x + threadIdx.x \quad (4.1)$$

Figura 4.2: Exemplo de abstração da GPU com uma e duas dimensões de blocos e threads.



Fonte: Figura Elaborada pelo autor.

Figura 4.3: Exemplo de abstração da GPU com três dimensões de blocos e *threads*.

Número de Threads = $16 \times 16 \times 4 = 1024$

Fonte: Figura Elaborada pelo autor.

$$indice_y = blockIdx.y * blockDim.y + threadIdx.y \quad (4.2)$$

$$indice_z = blockIdx.z * blockDim.z + threadIdx.z \quad (4.3)$$

Portanto, no presente trabalho utilizou-se uma dimensão (x) com 1024 *threads* por bloco, sendo este número o máximo suportado pela placa gráfica utilizada. Consequentemente, foi necessário apenas a Equação 4.1 para identificar unicamente uma *thread* no conjunto de execução. O número de blocos utilizados foi calculado de acordo com o número de sequências dadas como entrada, com a Equação 4.4. Nesta equação, a variável N , que representa o número de *threads* por bloco, para este trabalho possui valor de 1024.

$$numBlocos = \frac{A * B}{N} + 1 \quad (4.4)$$

Neste algoritmo é recebido um ou mais arquivos (.fas) como entrada. Dessa forma, é calculado a distância entre as sequências presentes em um mesmo arquivo, e posteriormente cada arquivo é comparado com as sequências presentes nos outros, comparando dois arquivos por vez. Portanto, para calcular o número de blocos que serão necessários na comparação dos dois arquivos, A e B, utiliza-se o número de sequências existentes no arquivo A (variável A) e no arquivo B (variável B). A soma do +1 na Equação 4.4 é necessária, pois o número de blocos é um valor inteiro, e caso a divisão não seja exata (Ex: 2,31), é necessário alocar um bloco a mais (Ex: 3) para o algoritmo conseguir processar todas as comparações. Assim, cria-se uma *thread* para cada cálculo de distância par-a-par, executando milhares de comparações em paralelos.

Um diagrama de como o algoritmo funciona na parte de cálculo de distâncias pode ser observado na Figura 4.4. Primeiramente ocorre a leitura dos arquivos de sequências para a memória RAM. De forma a realizar os cálculos utilizando a GPU, faz-se neces-

sário transferir as sequências lidas para a memória dedicada da placa gráfica. Assim, após a transferência, calcula-se as distâncias *hamming* entre as sequências. Posteriormente, o resultado obtido é salvo em disco, onde este é lido pela segunda parte do método, a qual realiza a construção da rede.

A etapa de cálculo de distâncias foi toda desenvolvida em um ambiente de programação denominado Microsoft Visual Studio, ou Visual Studio Community (MICROSOFT, 2019), o qual utilizou a linguagem C# e a extensão Cuda Toolkit¹. O uso da linguagem de programação C# ocorreu pelo fato de que esta possui uma série de recursos para a manipulação de cadeias de caracteres (*strings*), facilitando a leitura dos arquivos de sequências. Além disso, o Microsoft Visual Studio para a linguagem C# fornece uma coleção de objetos que facilita a modelagem da interface gráfica da ferramenta proposta. A integração desta com a linguagem de programação Cuda foi utilizada pelo fato que Cuda possui instruções para o processamento de dados paralelos em GPUs.

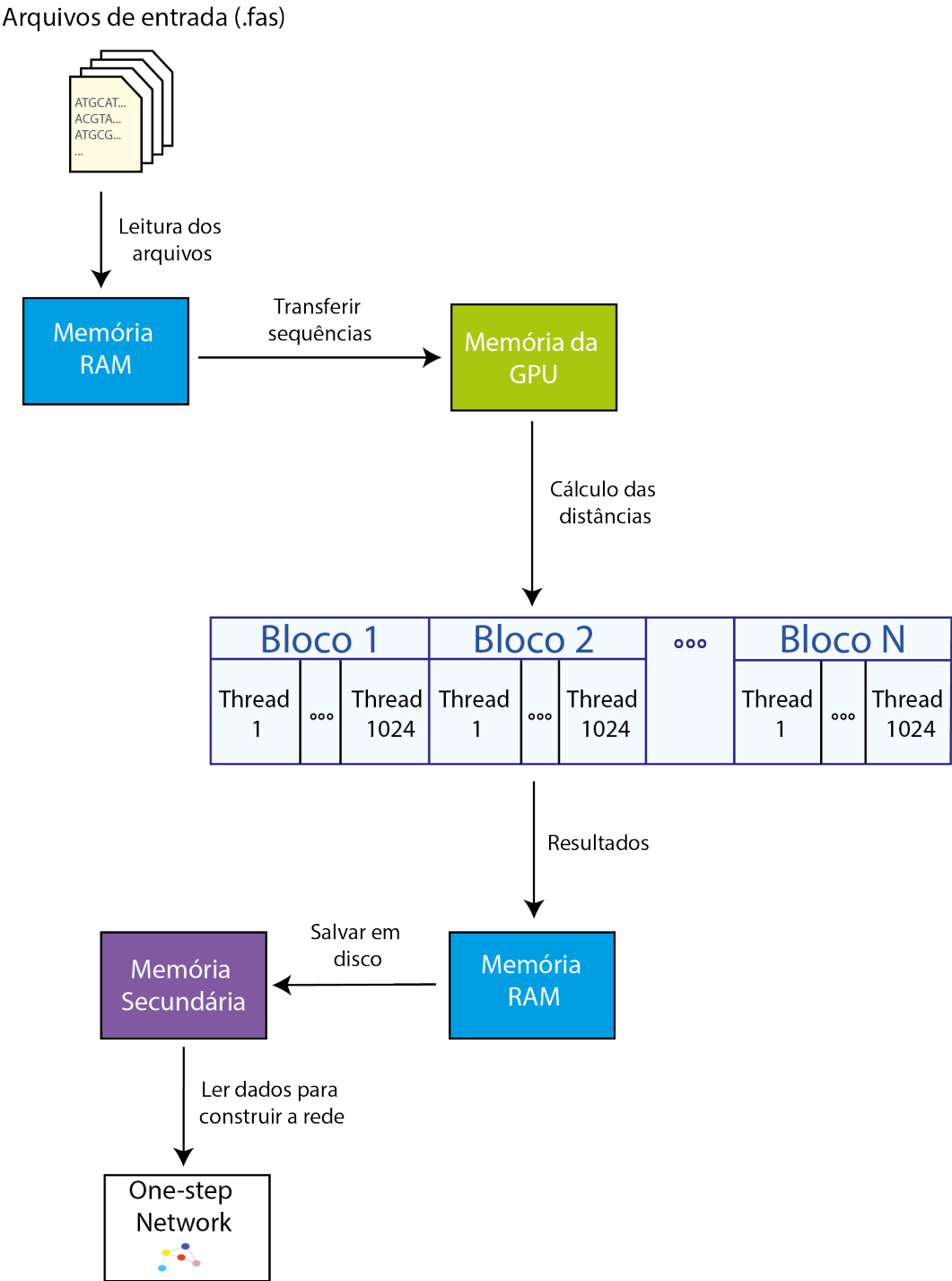
4.2 Construção da rede

Na segunda parte do método, é construída a rede *One-step*, de acordo com os dados previamente calculados. Como pode-se observar na Figura 4.5, inicialmente lê-se as distâncias dos haplótipos, as quais estão salvas em disco. Assim, constrói-se os vértices e arestas, formando, ao final, a rede *One-step*.

Nesta rede, cada haplótipo representa um vértice da rede, e as arestas as conexões *One-step*. Como mencionado na Subseção 4.1, este método recebe um ou mais arquivos de sequências como entrada. Dessa forma, na construção da rede, cada arquivo representa uma rede *One-step*, onde estas redes são conectadas através dos vértices que possuem menor distância com os vértices da outra rede. Por exemplo, ao se comparar todas as sequências da rede A com as sequências da rede B na primeira etapa, a

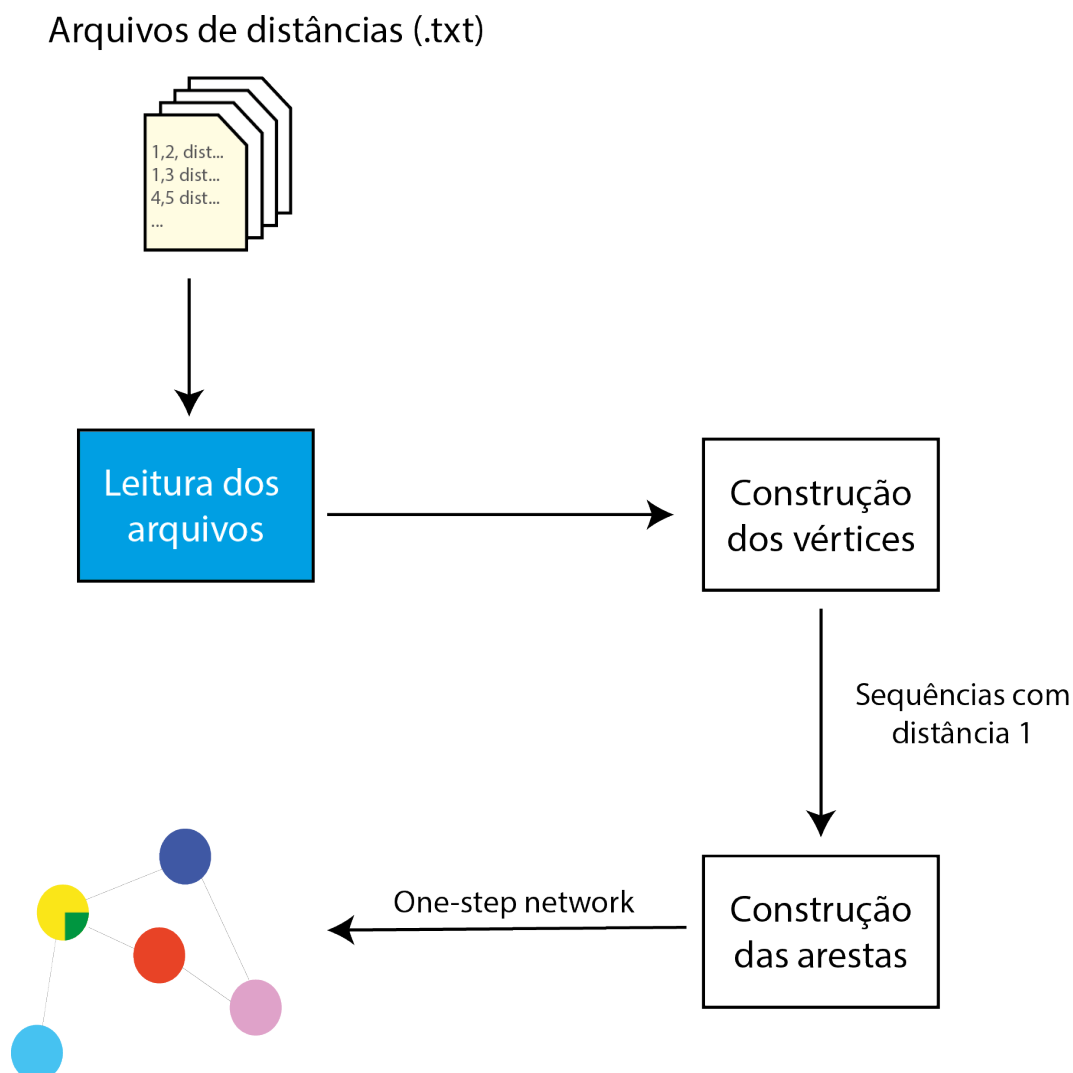
¹<https://developer.nvidia.com/cuda-toolkit>

Figura 4.4: Diagrama da etapa de cálculo de distâncias.



Fonte: Figura Elaborada pelo autor.

Figura 4.5: Diagrama da etapa de construção da rede.



Fonte: Figura Elaborada pelo autor.

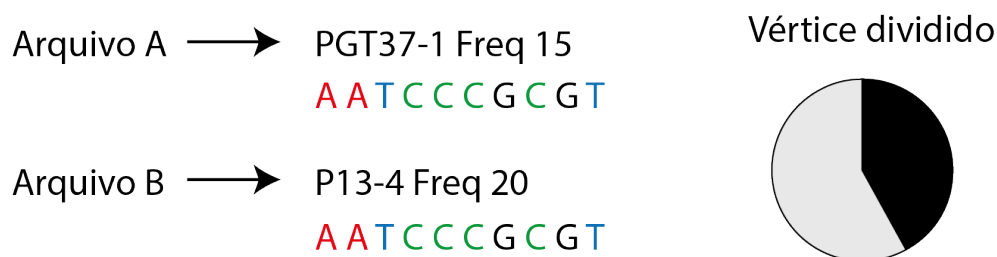
menor distância encontrada for 33, serão conectadas todos os haplótipos (vértices) da rede A com as da rede B que possuem distância de valor 33.

Além disso, caso haja haplótipos com sequências iguais em arquivos diferentes², será criado apenas um nó para aquela sequência, onde este será dividido, como em um gráfico de pizza, de acordo com as frequências das sequências envolvidas. Na Figura 4.6, encontra-se um exemplo de sequências iguais com frequências diferentes.

Neste exemplo, há dois haplótipos com sequências iguais pertencentes a arquivos

²Uma mesma sequência existente em um ou mais arquivos.

Figura 4.6: Exemplo de haplótipos iguais em arquivos diferentes.



Fonte: Figura Elaborada pelo autor.

diferentes. O primeiro possui frequência 15, portanto, esta sequência apareceu 15 vezes no sequenciamento do arquivo A. Da mesma forma, o segundo possui frequência 20, indicando que esta sequência apareceu 20 vezes no sequenciamento do arquivo B. O cálculo da porcentagem do arquivo A na divisão interna do vértice é feito como visto na Equação 4.5.

$$P = \frac{F_{tp}}{F_{total}} = \frac{15}{35} = 0,42. \quad (4.5)$$

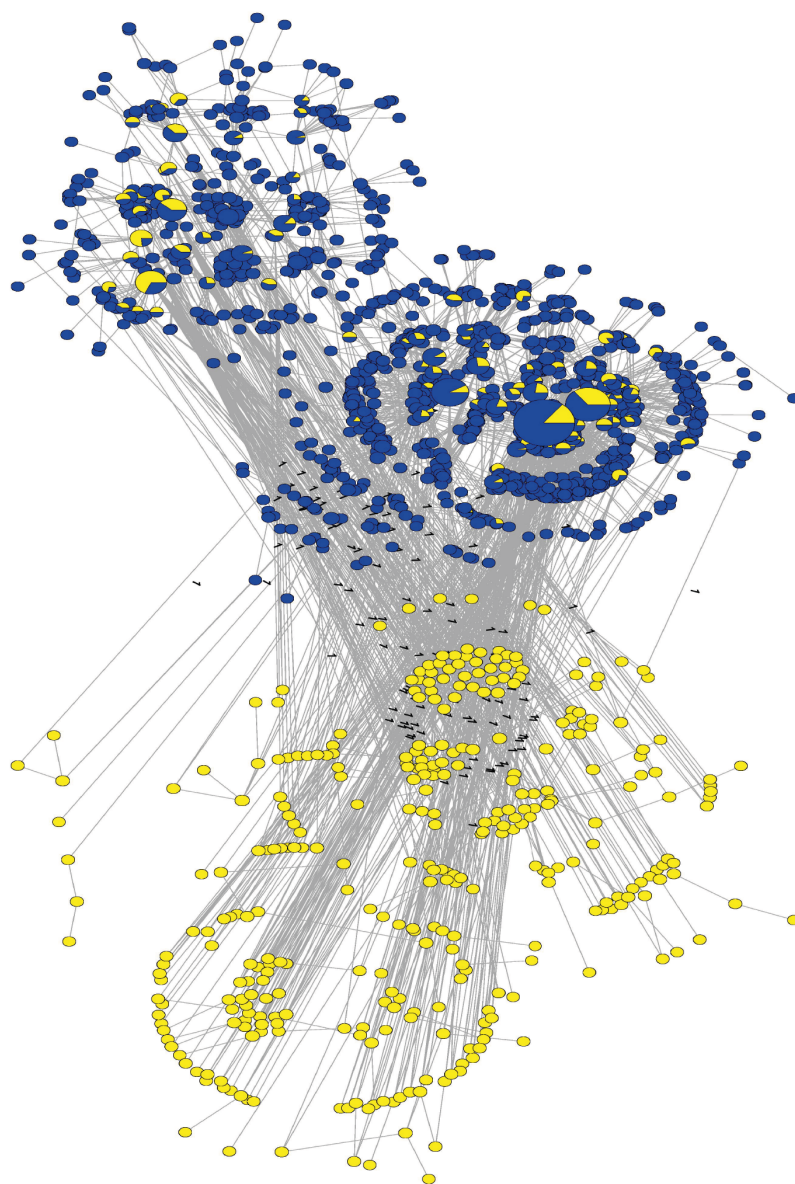
Nesta equação, a variável P é a porcentagem do arquivo A na divisão do vértice, F_{tp} é a frequência do do arquivo A, e F_{total} é a frequência total. Assim, F_{total} é a soma das frequências de todos os haplótipos que esta sequência ocorre, no caso a soma tem valor 35. Então, a partir do cálculo aferido anteriormente, verifica-se que o arquivo A terá uma porcentagem de 42% na subdivisão do vértice. Consequentemente, o arquivo B ocupará o resto da porcentagem, constituindo 58% do nó. Deve-se ressaltar que este vértice pode se conectar tanto com os haplótipos do arquivo A, quanto do B, desde que a distância entre as sequências conectadas seja 1.

Na Figura 4.7, pode-se observar a rede construída utilizando, como exemplo, dois arquivos fornecidos pelo *Centers for Disease Control and Prevention* (CDC, HHS/USA) no formato *FASTA* (*FAST-All*). Destaca-se que, quanto maior o grau do nó, maior seu tamanho.

A etapa de construção da rede foi toda desenvolvida na Unity3d³. A utilização

³<https://unity3d.com/pt/unity>

Figura 4.7: Rede *One-step* construída utilizando dois arquivos.



Fonte: Figura Elaborada pelo autor.

da Unity3d se dá pelo motivo de tal sistema ser um motor gráfico, o que facilita a criação de interfaces gráficas, e, da mesma forma, a implementação de um método para a concepção de grafos. Tal ferramenta fornece uma interface de uso agradável, e inclui funcionalidades avançadas para extrair completamente a capacidade das placas gráficas recentes (como Nvidia (NVIDIA, 2018)), obtendo um melhor desempenho para a aplicação (LV et al., 2013).

Este motor gráfico foi utilizado em estudos da área da Bioinformática, como apresentado por Perry et al. (PERRY et al., 2014) e Perry et al. (PERRY et al., 2013). Ademais, este *software* possibilita a escolha de programação em duas linguagens: C# e Javascript. No presente trabalho a linguagem de programação utilizada na Unity3d foi C#.

Capítulo 5

Resultados

Neste capítulo é discutido a metodologia de testes utilizada, e resultados obtidos a partir da execução do método que foi proposto no presente trabalho, comparando o tempo de execução da abordagem com apenas CPU, com a abordagem com GPU e o método *median-joining*. Além disso, apresenta-se a plataforma utilizada nos testes que foram executados.

5.1 Plataforma de testes

Os casos de testes do presente trabalho foram executados em uma máquina com processador Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz 2.60 GHz, memória RAM de 32,0 GB, sistema operacional Windows 10 Home 64 bits e placa gráfica GPU Nvidia GTX 960M 2,0 G de memória dedicada com 1024 núcleos Cuda.

5.2 Testes e Resultados

5.2.1 Abordagem com CPU vs Abordagem com GPU - Primeiro caso

Na metodologia de testes utilizada, foram executados 15 casos de testes, variando-se o número de sequências de haplótipos dados como entrada. Desta forma, cada caso de teste foi executado 10 vezes, com o objetivo de obter tempos de execução mais consistentes. Inicialmente, o número de sequências dados como entrada, tanto na abordagem com CPU, como na massivamente paralela, foi 5000 sequências. Assim, variou-se este número de sequências em 5000 para cada caso, onde a ultima configuração possuiu 75000 sequências. Ambas as abordagens obtiveram os mesmos resultados em termos de visualização.

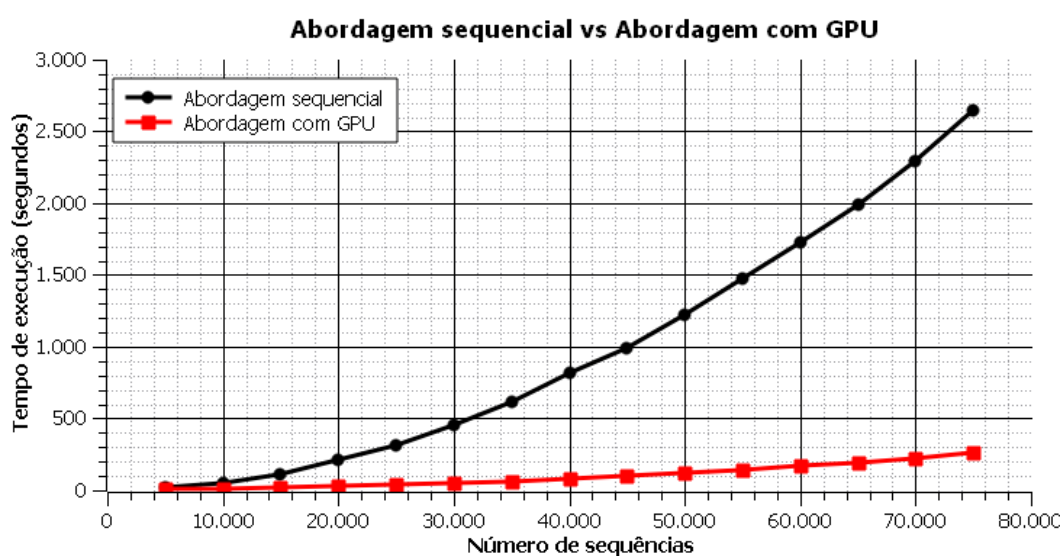
Os resultados das execuções do método *One-step* podem ser vistos na Tabela 5.1. Como pode-se observar, para todos os casos de testes executados, a abordagem massivamente paralela, com GPU, obteve-se melhor desempenho.

Tabela 5.1: Tempos de execução para a abordagem *One-step*.

Número de Sequências	One-step CPU		One-step GPU	
	Média (s)	Desvio Padrão (s)	Média (s)	Desvio Padrão (s)
5000	16,98	0,202	7,34	0,120
10000	51,73	0,183	10,86	0,031
15000	111,58	1,171	16,35	0,065
20000	208,87	4,289	25,91	1,109
25000	313,77	4,777	36,02	0,361
30000	451,96	3,890	48,35	0,394
35000	618,34	2,777	62,38	0,306
40000	813,30	3,285	77,85	0,216
45000	990,97	4,508	96,53	0,626
50000	1220,90	6,949	117,38	0,369
55000	1472,28	6,842	140,23	0,344
60000	1722,75	28,297	166,69	0,567
65000	1988,61	6,758	196,27	0,443
70000	2296,31	5,421	226,50	0,474
75000	2645,57	9,608	258,51	0,778

Constata-se que ao se comparar o caso de teste com 75000 sequências, o tempo de execução na estratégia com CPU é 2645 segundos, aproximadamente 44 minutos. No entanto, na estratégia massivamente paralela, o tempo de execução foi 258 segundos, aproximadamente 4 minutos e 18 segundos, ou seja, em torno de 10 vezes mais rápido que a com CPU. Assim, realça-se a importância de GPU em métodos com cálculos onerosos, com grandes quantidades de sequências a serem analisadas. A diferença dos tempos de execução pode ser melhor visualizada na Figura 5.1.

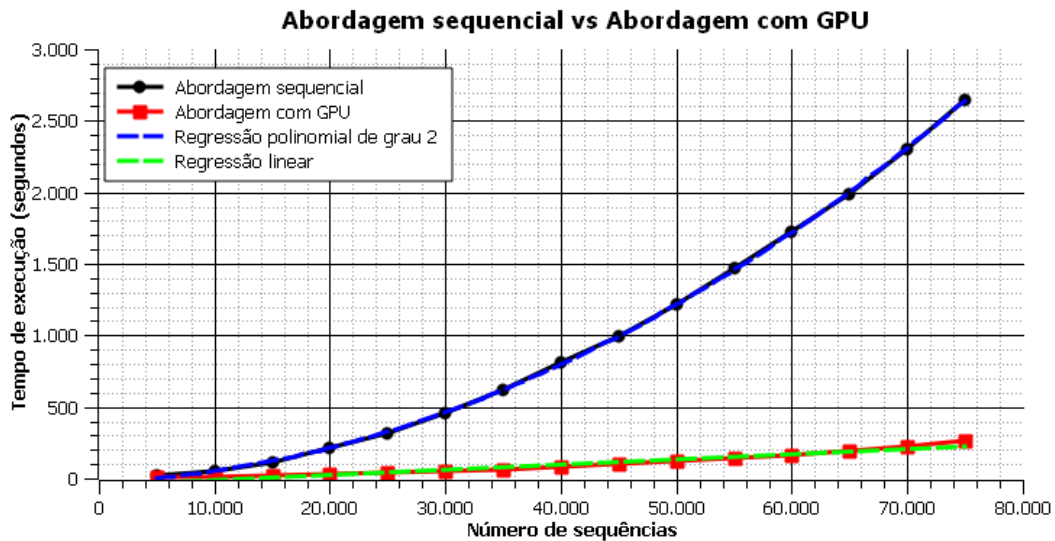
Figura 5.1: Gráfico da abordagem com CPU vs abordagem com GPU.



Fonte: Elaborado pelo autor.

Ademais, aplicou-se a regressão linear e polinomial de grau 2 nos resultados da abordagem com GPU e abordagem com CPU, respectivamente. Estas regressões podem ser visualizadas na Figura 5.2. Destaca-se que a abordagem com GPU obteve um coeficiente de Pearson (R^2) de 0.945, e assim a disposição de seus dados possui 94,5% de proximidade em relação a uma função linear. Em adendo, a abordagem com CPU obteve um coeficiente de Pearson no valor de 0.9998, alcançando uma proximidade de 99% ao se comparar com uma curva de um polinômio de grau 2. Assim, a notação assintótica para a abordagem com CPU é $O(n^2)$, enquanto na abordagem com GPU é $O(n)$.

Figura 5.2: Gráfico de Regressão da abordagem com CPU vs abordagem com GPU.



Fonte: Elaborado pelo autor.

Nota-se que o maior desvio-padrão encontrado foi na abordagem com CPU utilizando 60000 seqüências. Nas Tabelas 5.2 e 5.3 pode-se observar todas as 10 execuções para este caso de teste. Destaca-se que nos conjuntos de execução 1 a 5, a oscilação do tempo de execução foi pequena, com desvio-padrão de 1,93. Tal situação também ocorre nas execuções de 6 a 10 para 60000 seqüências, com desvio-padrão de 2,97. No entanto, ocasionou-se um degrau entre as execuções de 1 a 5 e 6 a 10, assim provocando um desvio padrão um pouco maior comparados aos outros, tendo em vista as 10 execuções. Este fato pode ter ocorrido devido a uma intervenção do escalonador de processos do sistema operacional, ou concorrência de outros processos, dado que o método é determinístico e possui a mesma entrada, executa e resulta do mesmo modo para os 10 testes.

Tabela 5.2: Tempos de execução 1-5 para o caso com 60000 seqüências.

Nº de seqüências	Execução 1	Execução 2	Execução 3	Execução 4	Execução 5
60000	1730,51	1736,22	1734,54	1732,98	1734,69

Assim, utilizando as médias dos tempos de execuções, da abordagem com CPU e massivamente paralela, foi calculado o *speedup* do método *One-step*. A equação do

Tabela 5.3: Tempos de execução 6-10 para o caso com 60000 sequências.

Nº de sequências	Execução 6	Execução 7	Execução 8	Execução 9	Execução 10
60000	1672,46	1679,28	1676,92	1681,4	1677,02

speedup pode ser vista na Equação 5.1 (LIAN et al., 2016). Nesta, a variável CPU representa o tempo de execução da abordagem com CPU para o caso x, e a variável GPU o tempo de execução da abordagem com GPU para o caso x.

$$Speedup(x) = \frac{CPU}{GPU} \quad (5.1)$$

Portanto, o *speedup* calculado para cada caso de teste pode ser observado na Tabela 5.4. Destaca-se que, conforme o número de sequências cresce, o *speedup* aumenta, estabilizando em aproximadamente 10 vezes com 40000 sequências.

Tabela 5.4: *Speedup* de todos os casos de testes.

Nº de sequências	Speedup
5000	2,31
10000	4,76
15000	6,82
20000	8,06
25000	8,71
30000	9,35
35000	9,91
40000	10,45
45000	10,26
50000	10,40
55000	10,50
60000	10,33
65000	10,13
70000	10,14
75000	10,23

Contudo, com 5000 sequências foi calculado o menor valor de *speedup*. Apesar disso, neste caso a abordagem com GPU é 2,31 vezes mais rápido que a abordagem com CPU. Por conseguinte, com 55000 sequências, a utilização de GPU diminuiu o

tempo de execução em 10,5 vezes, comparada com a abordagem com CPU. Assim, realça-se a importância do uso de um método eficiente para a construção de redes de haplótipos.

5.2.2 Abordagem com CPU vs Abordagem com GPU - Segundo caso

No segundo caso de execuções do método One-step CPU vs GPU, foram realizados 10 casos de teste, variando-se o número de sequências dado como entrada, de 100 mil em 100 mil, no intervalo de 100 mil a 1 milhão. Cada caso de teste foi executado 5 vezes. Não foram executados testes com um maior número de sequências, pois com estes casos de testes executados faz-se suficiente para validar o desempenho do método proposto utilizando GPU, mostrando a eficiência do mesmo na casa de milhão. Os resultados da execuções pode ser observados na Tabela 5.5.

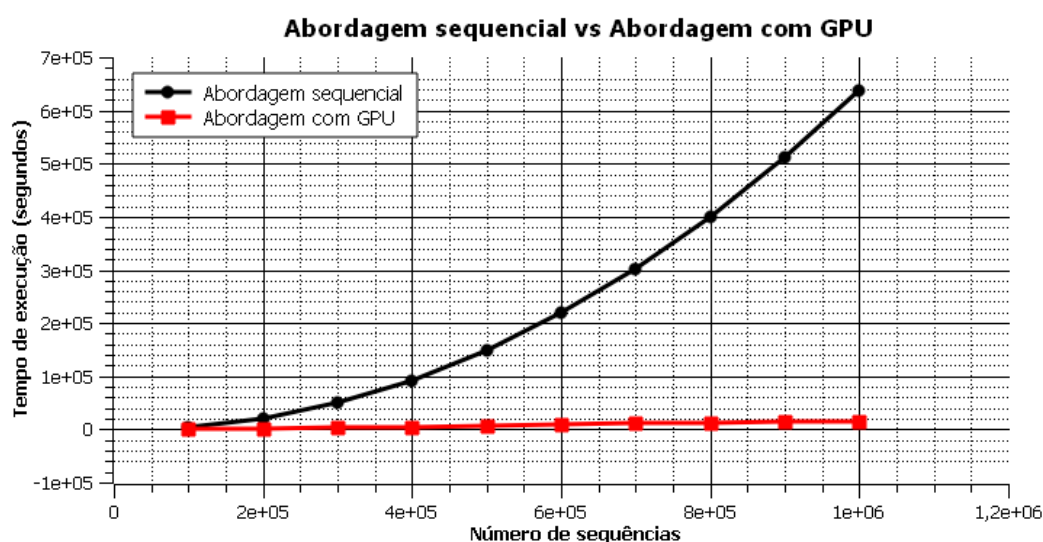
Tabela 5.5: Tempos de execução para a abordagem *One-step* de 100 mil a 1 milhão de sequências.

Número de Sequências	One-step CPU		One-step GPU	
	Média (s)	Desvio Padrão (s)	Média (s)	Desvio Padrão (s)
100000	4937,76	3,551	273,93	1,629
200000	19154,66	3,452	1066,54	4,680
300000	49745,98	3,679	2395,34	4,616
400000	91460,75	5,421	4305,83	4,259
500000	148237,22	6,442	6706,78	5,698
600000	218500,07	6,179	9629,98	5,121
700000	302511,87	7,367	11674,04	5,923
800000	400272,60	9,954	12801,30	8,176
900000	511783,33	10,328	13493,99	9,763
1000000	637041,42	10,124	13818,44	8,571

Dessa forma, analisando o primeiro caso de teste com 100 mil sequências, observa-se o tempo de execução com CPU foi de 4937 segundos, aproximadamente 1 hora e 22 minutos. Porém, na abordagem com GPU o tempo de execução foi de 273 segundos, aproximadamente 4 minutos e 30 segundos. Ademais, destaca-se que ao se comparar

o caso de teste com 1 milhão de sequência, o tempo de execução com CPU foi de 637041 segundos, ou seja, aproximadamente 7,37 dias para uma execução. Contudo, na execução com GPU, o tempo de execução foi de 13818 segundos, aproximadamente 3 horas e 50 minutos. Assim, a partir destes resultados, observa-se a importância da utilização de GPU em métodos que são necessários um grande número de iterações, como é o caso do método One-step, ao se trabalhar com milhares de sequências. A diferença entre os tempos de execução entre as duas abordagens pode ser visualizado no Gráfico 5.3.

Figura 5.3: Gráfico da abordagem com CPU vs abordagem com GPU de 100 mil a 1 milhão de sequências.

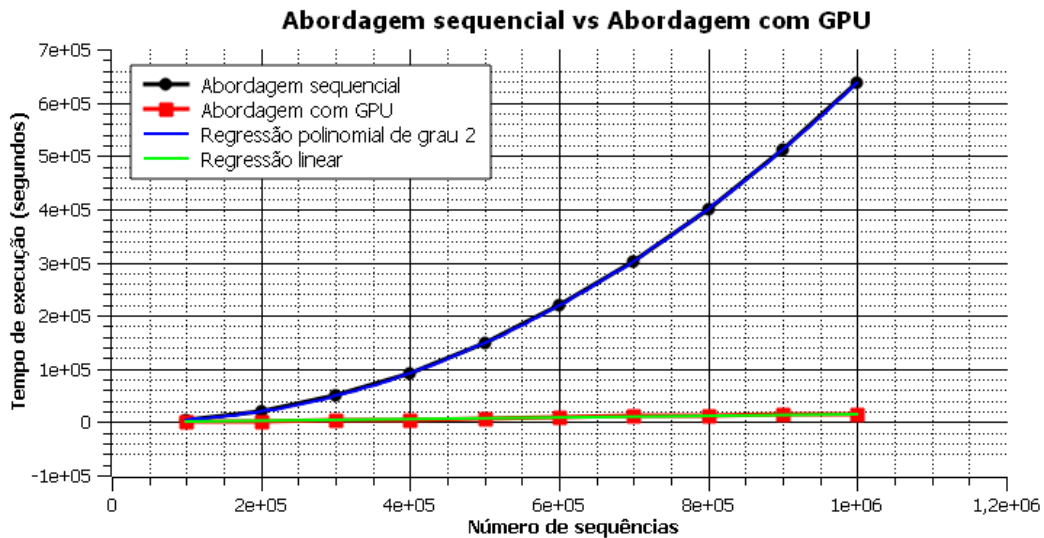


Fonte: Elaborado pelo autor.

Assim, aplicou-se a regressão linear e polinomial de grau 2 nos dados obtidos durante as execuções dos casos de testes da abordagem com GPU e CPU, respectivamente. As regressões comparadas com os resultados obtidos por cada abordagem podem ser visualizadas na Figura 5.4. Realça-se que a abordagem com GPU obteve um coeficiente de Pearson de 0.966, sendo assim a disposição de seus dados possuem 94,5% de proximidade em relação a uma função linear. Em adendo, a abordagem com CPU obteve um coeficiente de Pearson no valor de 0,9999, alcançando uma proximidade de 99,99% ao se comparar com uma curva de um polinômio de grau 2. Assim,

a notação assintótica para a abordagem com CPU é $O(n^2)$, enquanto na abordagem com GPU é $O(n)$.

Figura 5.4: Gráfico de regressão da abordagem com CPU vs abordagem com GPU de 100 mil a 1 milhão de sequências.



Fonte: Elaborado pelo autor.

A partir das médias dos tempos de execução obtidos, calculou-se o *speedup* dos casos de teste de 100 mil a 1 milhão de sequências. A equação do *speedup* pode ser vista na Equação 5.1. Os valores de *speedup* dos casos com 100 mil sequências até 1 milhão podem ser analisados na Tabela 5.6.

Tabela 5.6: *Speedup* de todos os casos de testes de 100 mil a 1 milhão.

Nº de sequências	Speedup
100000	18,02
200000	17,95
300000	20,76
400000	21,24
500000	22,10
600000	22,68
700000	25,91
800000	31,26
900000	37,92
1000000	46,10

O menor *speedup* foi calculado para o caso com 200 mil sequência, sendo neste

caso a abordagem com GPU 17,95 vezes mais rápida que a abordagem com CPU. Entretanto, o maior *speedup* foi obtido no caso o qual utilizou-se 1 milhão de sequências, sendo neste com o uso de GPU foi 46,10 vezes mais rápida a execução do que utilizando CPU. Dessa forma, ressalta-se a importância do presente trabalho, uma vez que sequenciadores de nova geração geram milhares de sequências para análise, a utilização de GPU é importante, pois possui uma arquitetura massivamente paralela, otimizando o tempo de execução do método.

5.2.3 Método *median-joining*

Nos testes performados com o método *median-joining*, foram executados quatro cenários, nos quais variou-se a divergência entre as sequências dentro do conjunto. Esta divergência é determinada pelo número de nucleotídeos diferentes entre cada par de sequência. Em cada cenário executou-se cinco testes, variando-se o número de sequências dadas como entradas, no qual cada teste foi executado 10 vezes afim de obter tempos de execução mais consistentes para análise. As sequências de haplótipos utilizadas foram fornecidas pelo CDC e, anteriormente a execução dos testes, foram passadas por um pré-processamento.

O pré-processamento funciona da seguinte forma:

- No primeiro cenário, não houve alterações nas sequências durante o pré-processamento.
- No segundo, alterou-se um nucleotídeo aleatório em cada sequência, afim de aumentar suas divergências.
- No terceiro, alterou-se dois nucleotídeos de cada sequência de haplótipos, de forma aleatória.
- Por fim, no quarto cenário, alterou-se três nucleotídeos em cada sequência, de forma aleatória.

Nas Tabelas 5.7, 5.8, 5.9 e 5.10 podem ser observados as médias dos tempos de execução do método *median-joining* no primeiro, segundo, terceiro e quarto cenário, respectivamente (Valor NC = Não Completado).

Tabela 5.7: Média dos tempos de execução do método *median-joining*.

Número de Sequências	Média (s)	Desvio Padrão (s)
1000	206,16	2,401
2000	223,46	2,292
3000	244,31	5,173
4000	264,72	5,452
5000	293,55	4,551

Tabela 5.8: Média dos tempos de execução do método *median-joining* - Sequências com alteração randômica de um nucleotídeo.

Número de Sequências	Média (s)	Desvio Padrão (s)
1000	615,27	6,530
2000	1611,94	33,816
3000	2985,86	56,666
4000	NC	NC
5000	NC	NC

Tabela 5.9: Média dos tempos de execução do método *median-joining* - Sequências com alteração randômica de dois nucleotídeo.

Número de Sequências	Média (s)	Desvio Padrão (s)
1000	961,75	21,350
2000	NC	NC
3000	NC	NC
4000	NC	NC
5000	NC	NC

Tabela 5.10: Média dos tempos de execução do método *median-joining* - Sequências com alteração randômica de três nucleotídeo.

Número de Sequências	Média (s)	Desvio Padrão (s)
1000	1578,61	28,293
2000	NC	NC
3000	NC	NC
4000	NC	NC
5000	NC	NC

Nota-se que no primeiro cenário, onde não houve alteração nas sequências utilizadas, o método *median-joining* conseguiu executar todos os casos de teste. Este comportamento ocorreu pois estas sequências fornecidas pelo CDC são haplótipos muito próximos uns dos outros, ou seja, as sequências possuem distâncias com valores baixos entre si, ao compará-las par-a-par. Porém, conforme aumenta-se a divergência entre cada sequência, como acontece no segundo, terceiro e quarto cenário, em alguns casos o método não conseguiu concluir seu processamento (NC - Não completado). Como exemplo, no segundo cenário, visto na Tabela 5.8, o método *median-joining* conseguiu completar o processamento com até 3000 sequências. Contudo, no terceiro e quarto cenário (Tabela 5.9 e 5.10), o método *median-joining* conseguiu completar o processamento apenas nas execuções utilizando 1000 sequências. Assim, destaca-se que o método *median-joining* possui dificuldade ao ser executado com sequências que possuem uma maior distância entre elas, o qual é um problema, pois ao se trabalhar com milhares de haplótipos, a divergência entre as sequências acaba sendo variada.

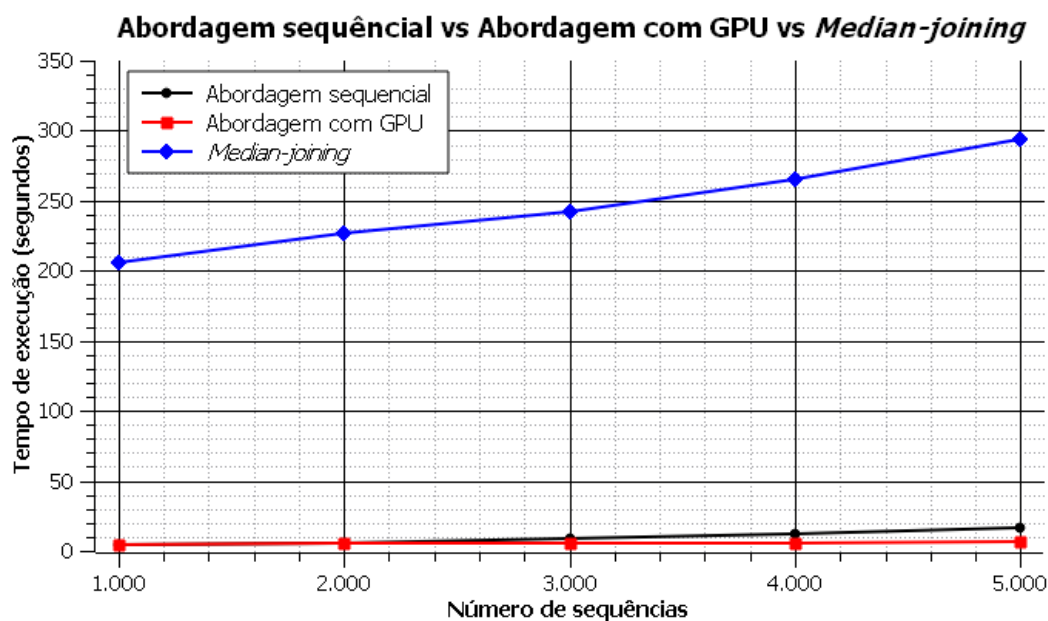
Assim, na Tabela 5.11 observa-se os tempos de execuções da abordagem com CPU e com GPU, com o uso dos mesmos conjuntos de sequências do primeiro, segundo, terceiro e quarto cenário do método *median-joining*. Constata-se que, mesmo que as sequências sejam mais distantes, os tempos de execução entre os cenários estão próximos, sendo que em alguns casos, ao aumentar a divergência, o tempo de execução diminuiu, como por exemplo pode ser visto ao comparar o caso de 5000 sequências do segundo, terceiro e quarto cenário.

Tabela 5.11: Média dos tempos de execução, em segundos, da abordagem com CPU vs abordagem com GPU em todos os cenários.

Número de Sequências	Primeiro Cenário		Segundo Cenário		Terceiro Cenário		Quarto Cenário	
	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU
1000	4,31	4,55	2,91	3,08	2,86	3,03	2,82	3,00
2000	6,01	5,08	5,12	4,16	5,13	4,04	5,05	4,02
3000	8,71	5,60	8,71	5,59	8,56	5,20	8,34	5,12
4000	12,18	6,04	13,35	7,18	13,09	6,45	12,65	6,36
5000	16,28	6,20	19,51	9,32	18,93	8,01	17,96	7,70

Na Figura 5.5 é possível visualizar a diferença dos tempos de execução, no primeiro cenário, entre as abordagens sequencial, GPU e do método *median-joining*. Nota-se que, mesmo sem ocorrer alterações de nucleotídeos em cada sequência no pré-processamento, o método *median-joining* obteve um pior desempenho que as outras abordagens.

Figura 5.5: Gráfico da abordagem sequencial vs abordagem com GPU vs *Median-joining*.



Fonte: Elaborado pelo autor.

Portanto, utilizando os dados do primeiro cenário, calculou-se o *speedup* das abordagens com CPU e GPU, comparando com o método *median-joining*. A equação do *speedup* pode ser vista na Equação 5.2. Nesta, a variável MJ representa o tempo de

execução do método *median-joining* para o caso x , e a variável ABD o tempo de execução da abordagem com CPU ou com GPU para o caso x .

$$Speedup(x) = \frac{MJ}{ABD} \quad (5.2)$$

Dessa forma, o *speedup* calculado é mostrado na Tabela 5.12. Analisando os dados, destaca-se que o método *median-joining* obteve um desempenho de até 47,72 vezes pior do que abordagem com CPU, e 47,27 vezes pior que a abordagem com GPU. Assim, ressalta-se a importância de um método que possua, além de uma boa visualização, um bom desempenho, devido ao volume de dados gerado por sequenciadores de nova geração.

Tabela 5.12: *Speedup* abordagem com CPU e GPU vs método *Median-joining*

Número de Sequências	Speedup	
	<i>Median-joining</i> vs CPU	<i>Median-joining</i> vs GPU
1000	47,72	45,31
2000	37,13	43,98
3000	28,02	43,59
4000	21,72	43,82
5000	18,02	47,27

Capítulo 6

Conclusão

Como conclusão, destaca-se a importância do método *One-step* para análise de redes de haplótipos, pois este fornece uma estrutura natural, agrupando os haplótipos que possuem ligações evolutivas mais próximas (PALMER et al., 2016), sendo possível visualizar pequenas variações entre as sequências. Não obstante, não há um método computacional consolidado na literatura para a construção deste tipo de rede. Ademais, a utilização de GPU possibilita a comparação de milhares de sequências paralelamente, o que não é possível quando sistema usufrui apenas dos recursos da CPU, devido ao número limitado de núcleos disponíveis.

O método proposto também possui vantagens sobre o método *median-joining*, sendo este o que mais se destaca na literatura. Na visualização, o método *One-step* não cria vetores medianos como ocorre com o método *median-joining*. Dessa forma, constrói-se a rede com o conjunto de dados originais, sem qualquer suposição sobre suas origens ancestrais. Em relação ao desempenho, verificou-se que o método *One-step* com GPU foi até 47 vezes mais rápido que o método *median-joining*, ao se trabalhar com 5000 haplótipos.

Assim, ressalta-se que a utilização de placas gráficas otimiza o desempenho do método, o qual é importante, devido aos sequenciamentos de nova geração que geram milhares ou milhões de sequências. Por fim, este método auxiliará profissionais da

área biológica em estudos evolucionários, com um método robusto para análise das relações entre as sequências envolvidas no cálculo da rede.

6.1 Contribuições Obtidas

São destacados nesta seção as contribuições no presente trabalho. A partir das execuções realizadas e análises feitas, provou-se que é possível construir uma rede com até milhões de sequências em um tempo factível, ao se utilizar um método eficiente. O método *One-step* possui vantagens de desempenho, e visualização, sendo possível visualizar sequências que estão estreitamente relacionadas sem qualquer suposição de sua origem ancestral. Assim, profissionais da área biológica não ficaram limitados a analisar uma rede com um pequeno número de sequências, o qual era um empecilho, sendo necessário quebrar grandes conjuntos de sequências em vários menores para ser possível analisa-los.

6.2 Trabalhos Futuros

Nos trabalhos futuros, pretende-se construir redes *One-step* em um ambiente 3d, não apenas 2d como consta no presente trabalho. Ao se construir redes *One-step* com milhares ou milhões de sequências, em um ambiente apenas 2d a sobreposição de nós e arestas é inevitável, podendo tornar a rede confusa devido a quantidade de dados em análise. Dessa forma, com o objetivo de evoluir ainda mais este projeto, será modelado um algoritmo para construção de redes *One-step* 3d.

Referências

- ANDRESIUK, M. V.; GORDO, F. P.; SAARMA, M.; ELISSONDO, M. C.; TARABORELLI, A.; CASALONGUE, C.; DENEGRI, G.; SAARMA, U. *Echinococcus granulosus* genotype g1 dominated in cattle and sheep during 2003–2006 in buenos aires province, an endemic area for cystic echinococcosis in argentina. *Acta tropica*, Elsevier, v. 127, n. 2, p. 136–142, 2013.
- BANDELT, H.-J.; FORSTER, P.; RÖHL, A. Median-joining networks for inferring intraspecific phylogenies. *Molecular biology and evolution*, v. 16, n. 1, p. 37–48, 1999.
- BANDELT, H.-J.; FORSTER, P.; SYKES, B. C.; RICHARDS, M. B. Mitochondrial portraits of human populations using median networks. *Genetics*, Genetics Soc America, v. 141, n. 2, p. 743–753, 1995.
- BAUM, D. Reading a phylogenetic tree: the meaning of monophyletic groups. *Nature Education*, v. 1, n. 1, p. 190, 2008.
- BEAR, R.; RINTOUL, D.; SNYDER, B.; SMITH-CALDAS, M.; HERREN, C.; HORNE, E. Principles of biology. The Authors, 2016.
- BIGGS, N.; LLOYD, E. K.; WILSON, R. J. *Graph Theory, 1736-1936*. [S.l.]: Oxford University Press, 1976.
- BUJÁN, N.; BALBOA, S.; ROMALDE, J. L.; TORANZO, A. E.; MAGARIÑOS, B. Population genetic and evolution analysis of controversial genus *edwardsiella* by multilocus sequence typing. *Molecular phylogenetics and evolution*, Elsevier, v. 127, p. 513–521, 2018.
- CAMPO, D. S.; DIMITROVA, Z.; YAMASAKI, L.; SKUMS, P.; LAU, D. T.; VAUGHAN, G.; FORBI, J. C.; TEO, C.-G.; KHUDYAKOV, Y. Next-generation sequencing reveals large connected networks of intra-host hcv variants. *BMC genomics*, BioMed Central, v. 15, n. 5, p. S4, 2014.
- CHONG, E. T. J.; NEOH, J. W. F.; LAU, T. Y.; LIM, Y. A.-L.; CHUA, K. H.; LEE, P.-C. Genetic and haplotype analyses targeting cytochrome b gene of plasmodium knowlesi isolates of malaysian borneo and peninsular malaysia. *Acta tropica*, Elsevier, v. 181, p. 35–39, 2018.

- CLEMENT, M.; POSADA, D.; CRANDALL, K. A. Tcs: a computer program to estimate gene genealogies. *Molecular ecology*, v. 9, n. 10, p. 1657–1659, 2000.
- DELWICHE, C. F.; PALMER, J. D. Rampant horizontal transfer and duplication of rubisco genes in eubacteria and plastids. *Molecular Biology and Evolution*, v. 13, n. 6, p. 873–882, 1996.
- DEYASI, K.; BANERJEE, A.; DEB, B. Phylogeny of metabolic networks: A spectral graph theoretical approach. *Journal of biosciences*, Springer, v. 40, n. 4, p. 799–808, 2015.
- DURBIN, R. Efficient haplotype matching and storage using the positional burrows–wheeler transform (pbwt). *Bioinformatics*, Oxford University Press, v. 30, n. 9, p. 1266–1272, 2014.
- EDGAR, R. C.; BATZOGLOU, S. Multiple sequence alignment. *Current opinion in structural biology*, Elsevier, v. 16, n. 3, p. 368–373, 2006.
- EIDHAMMER, I.; JONASSEN, I.; TAYLOR, W. R. Structure comparison and structure patterns. *Journal of Computational Biology*, Mary Ann Liebert, Inc., v. 7, n. 5, p. 685–716, 2000.
- ENGINEERING, F. *Phylogenetic Network Software*. 2017. Disponível em: <<http://www.fluxus-engineering.com/sharenet.htm>>.
- ENGINEERING, F. *Phylogenetic Network Software User Guide*. 2017. Disponível em: <http://www.fluxus-engineering.com/Network4600_user_guide.pdf>.
- FARRIS, J. S. Methods for computing wagner trees. *Systematic Biology*, Society of Systematic Zoology, v. 19, n. 1, p. 83–92, 1970.
- FAZ, M.; MARTÍNEZ, J. S.; GÓMEZ, L. B.; QUIJANO-HERNÁNDEZ, I.; FAJARDO, R.; ÁNGEL-CARAZA, J. D. Origin and genetic diversity of canine parvovirus 2c circulating in mexico. *Archives of virology*, Springer, v. 164, n. 2, p. 371–379, 2019.
- FERGUSON, T. S. Who solved the secretary problem? *Statistical science*, JSTOR, p. 282–289, 1989.
- FIGUEIREDO, G. M. Existence of a positive solution for a kirchhoff problem type with critical growth via truncation argument. *Journal of Mathematical Analysis and Applications*, Elsevier, v. 401, n. 2, p. 706–713, 2013.
- GANDOMI, A. H.; ALAVI, A. H. Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, Elsevier, v. 17, n. 12, p. 4831–4845, 2012.

- GANOVA-RAEVA, L.; DIMITROVA, Z.; ALEXIEV, I.; PUNKOVA, L.; SUE, A.; XIA, G.-I.; GANCHEVA, A.; DIMITROVA, R.; KOSTADINOVA, A.; GOLKOCHIEVA-MARKOVA, E.; KHUDYAKOV, Y. Hcv transmission in high-risk communities in bulgaria. *PloS one*, Public Library of Science, v. 14, n. 3, p. e0212350, 2019.
- GOODWIN, S.; MCPHERSON, J. D.; MCCOMBIE, W. R. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, Nature Publishing Group, v. 17, n. 6, p. 333, 2016.
- GOULD, S. J. *Ontogeny and phylogeny*. [S.l.]: Harvard University Press, 1977.
- GOUY, M.; GUINDON, S.; GASCUEL, O. Seaview version 4: a multiplatform graphical user interface for sequence alignment and phylogenetic tree building. *Molecular biology and evolution*, Oxford University Press, v. 27, n. 2, p. 221–224, 2009.
- GRIFFITHS, R. C.; MARJORAM, P. Ancestral inference from samples of dna sequences with recombination. *Journal of Computational Biology*, v. 3, n. 4, p. 479–502, 1996.
- HAIDER, M. S. H.; DEEBA, F.; KHAN, W. H.; NAQVI, I. H.; ALI, S.; AHMED, A.; BROOR, S.; ALSENAIDY, H. A.; ALSENAIDY, A. M.; DOHARE, R.; PARVEEN, S. Global distribution of na1 genotype of respiratory syncytial virus and its evolutionary dynamics assessed from the past 11 years. *Infection, Genetics and Parveen, Shama*, Elsevier, v. 60, p. 140–150, 2018.
- HAPUARACHCHI, H.; BANDARA, K.; SUMANADASA, S.; HAPUGODA, M.; LAI, Y.-L.; LEE, K.-S.; TAN, L.-K.; LIN, R. T.; NG, L. F.; BUCHT, G.; ABEYEWICKREME, W.; NG, L.-C. Re-emergence of chikungunya virus in south-east asia: virological evidence from sri lanka and singapore. *Journal of General Virology*, Microbiology Society, v. 91, n. 4, p. 1067–1076, 2010.
- HARTUV, E.; SHAMIR, R. A clustering algorithm based on graph connectivity. *Information processing letters*, Elsevier, v. 76, n. 4-6, p. 175–181, 2000.
- HUANG, S.-W.; HUANG, Y.-H.; TSAI, H.-P.; KUO, P.-H.; WANG, S.-M.; LIU, C.-C.; WANG, J.-R. A selective bottleneck shapes the evolutionary mutant spectra of enterovirus a71 during viral dissemination in humans. *Journal of virology*, Am Soc Microbiol, v. 91, n. 23, p. e01062–17, 2017.
- HUANG, X. On global sequence alignment. *Bioinformatics*, Oxford University Press, v. 10, n. 3, p. 227–235, 1994.
- HUSON, D. H.; BRYANT, D. Application of phylogenetic networks in evolutionary studies. *Molecular biology and evolution*, Oxford University Press, v. 23, n. 2, p. 254–267, 2005.

HUSON, D. H.; LINZ, S. Autumn algorithm—computation of hybridization networks for realistic phylogenetic trees. *IEEE/ACM transactions on computational biology and bioinformatics*, IEEE, v. 15, n. 2, p. 398–410, 2018.

HUSON, D. H.; RUPP, R.; SCORNAVACCA, C. *Phylogenetic networks: concepts, algorithms and applications*. [S.l.]: Cambridge University Press, 2010.

HUSON, D. H.; SCORNAVACCA, C. A survey of combinatorial methods for phylogenetic networks. *Genome biology and evolution*, Oxford University Press, v. 3, p. 23–35, 2011.

KAPOURANI, A. C. Similarity and recommender systems. 2015.

KENSICHER, T.; LECLERC, J.; BIEDIGER, D.; SHAH, D. J.; SEIMENIS, I.; BECKER, A. T.; TSEKOS, N. V. Towards mri-guided and actuated tetherless milli-robots: Preoperative planning and modeling of control. In: IEEE. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2017. p. 6440–6447.

KONG, S. *On the Use of Median-Joining Networks: A Philosophical and Empirical Evaluation of Its Suitability in Evolutionary Biology*. Tese (Doutorado) — Department of Ecology and Evolutionary Biology University of Toronto, 2015.

KONG, S.; SÁNCHEZ-PACHECO, S. J.; MURPHY, R. W. On the use of median-joining networks in evolutionary biology. *Cladistics*, Wiley Online Library, v. 32, n. 6, p. 691–699, 2016.

KOVALEV, S.; MUKHACHEVA, T. Reconsidering the classification of tick-borne encephalitis virus within the siberian subtype gives new insights into its evolutionary history. *Infection, Genetics and Evolution*, Elsevier, v. 55, p. 159–165, 2017.

LADNER, J. T.; WILEY, M. R.; MATE, S.; DUDAS, G.; PRIETO, K.; LOVETT, S.; NAGLE, E. R.; BEITZEL, B.; GILBERT, M. L.; FAKOLI, L.; DICLARO, J. W.; SCHOEPP, R. J.; FAIR, J.; KUHN, J. H.; HENSLEY, L. E.; PARK, D. J.; SABETI, P. C.; RAMBAUT, A.; SANCHEZ-LOCKHART, M.; BOLAY, F. K.; KUGELMAN, J. R.; PALACIOS, G. Evolution and spread of ebola virus in liberia, 2014–2015. *Cell host & microbe*, Elsevier, v. 18, n. 6, p. 659–669, 2015.

LANGDON, W. B.; LAM, B. Y. H.; MODAT, M.; PETKE, J.; HARMAN, M. Genetic improvement of gpu software. *Genetic Programming and Evolvable Machines*, Springer, v. 18, n. 1, p. 5–44, 2017.

LEE, S.; EIGENMANN, R. Openmpc: Extended openmp programming and tuning for gpus. In: IEEE COMPUTER SOCIETY. *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*. [S.l.], 2010. p. 1–11.

LEIGH, J. W.; BRYANT, D. popart: full-feature software for haplotype network construction. *Methods in Ecology and Evolution*, Wiley Online Library, v. 6, n. 9, p. 1110–1116, 2015.

LEMOS, M.; ARAGAO, M. V. S. P.; CASANOVA, M. A. *Padrões em Biossequências*. [S.l.]: PUC, 2003.

LEWIS, B.; BERG, D. J. *Threads primer: a guide to multithreaded programming*. [S.l.]: SunSoft Press Mountain View, CA, 1996.

LI, L.; CHEN, X.; WANG, W. A cuda-based parallel adaptive dynamic programming algorithm. In: IEEE. *2017 36th Chinese Control Conference (CCC)*. [S.l.], 2017. p. 3510–3515.

LIAN, X.; ZHANG, H.; HSIEH, C.-J.; HUANG, Y.; LIU, J. A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2016. p. 3054–3062.

LV, Z.; TEK, A.; SILVA, F. D.; EMPEREUR-MOT, C.; CHAVENT, M.; BAADEN, M. Game on, science-how video game technology may help biologists tackle visualization challenges. *PloS one*, Public Library of Science, v. 8, n. 3, p. e57990, 2013.

MATSCHINER, M. Fitchi: haplotype genealogy graphs based on the fitch algorithm. *Bioinformatics*, Oxford University Press, v. 32, n. 8, p. 1250–1252, 2015.

MEGA: Molecular evolutionary genetic analysis. 2019. Disponível em: <<https://www.megasoftware.net/>>.

MICROSOFT. *Microsoft Visual Studio*. 2019. Disponível em: <<https://www.visualstudio.com/pt-br>>.

MITCHELL, A.; CHANG, H.-Y.; DAUGHERTY, L.; FRASER, M.; HUNTER, S.; LOPEZ, R.; MCANULLA, C.; MCMENAMIN, C.; NUKA, G.; PESSEAT, S.; SANGRADOR-VEGAS, A.; SCHEREMETJEW, M.; RATO, C.; YONG, S.-Y.; BATEMAN, A.; PUNTA, M.; ATTWOOD, T. K.; SIGRIST, C. J. A.; REDASCHI, N.; RIVOIRE, C.; XENARIOS, I.; KAHN, D.; GUYOT, D.; BORK, P.; LETUNIC, I.; GOUGH, J.; OATES, M.; HAFT, D.; HUANG, H.; NATALE, D. A.; WU, C. H.; ORENGO, C.; SILLITOE, I.; MI, H.; D., T. P.; FINN, R. D. The interpro protein families database: the classification resource after 15 years. *Nucleic acids research*, Oxford University Press, v. 43, n. D1, p. D213–D221, 2014.

MUTNEJA, V.; SINGH, S. Graphical processing unit accelerated face resolution enhancement using pixels-homogeneity and relative-ratios. *Int J Cur Res Revl Vol*, v. 9, n. 13, p. 1, 2017.

NATURE. *Scitable by nature education. A collaborative learning space of science*. 2017. Disponível em: <<https://www.nature.com/scitable/definition/haplotype-haplotypes-142>>.

NISAR, A.; WAHEED, A.; KHAN, S.; FENG, X.; SHAH, A. H. Population structure, genetic diversity and phylogenetic analysis of different rural and commercial chickens of pakistan using complete sequence of mtdna d-loop. *Mitochondrial DNA Part A*, Taylor & Francis, v. 30, n. 2, p. 273–280, 2019.

NVIDIA. *Nvidia*. 2018. Acessado: 01-05-2018. Disponível em: <<http://www.nvidia.com.br/page/home.html>>.

NVIDIA. *Cuda C Programming Guide*. 2019. Acessado: 14-05-2019. Disponível em: <<https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>>.

NVIDIA. *O que é computação acelerada por placas de vídeo?* 2019. Acessado: 08-05-2019. Disponível em: <<https://www.nvidia.com.br/object/what-is-gpu-computing-br.html>>.

PALMER, B. A.; SCHMIDT-MARTIN, D.; DIMITROVA, Z.; SKUMS, P.; CROSBIE, O.; KENNY-WALSH, E.; FANNING, L. J. Network analysis of the chronic hepatitis c virome defines hypervariable region 1 evolutionary phenotypes in the context of humoral immune responses. *Journal of virology*, Am Soc Microbiol, v. 90, n. 7, p. 3318–3329, 2016.

PATEL, R. K.; JAIN, M. Ngs qc toolkit: a toolkit for quality control of next generation sequencing data. *PloS one*, Public Library of Science, v. 7, n. 2, p. e30619, 2012.

PERRY, D.; ARAGON, C.; CRUZ, S.; PETERS, M. A.; CHOWNING, J. T. Human centered game design for bioinformatics and cyberinfrastructure learning. In: ACM. *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery*. [S.l.], 2013. p. 74.

PERRY, D.; ROBINSON, J.; CRUZ, S.; ARAGON, C.; CHOWNING, J. T.; PETERS, M. Game design for bioinformatics and cyberinfrastructure learning: a parallel computing case study. *Concurrency and Computation: Practice and Experience*, Wiley Online Library, v. 26, n. 13, p. 2303–2315, 2014.

POND, J. E.; SUTTON, G. W. Precise evaluation of the helmholtz equation for optical propagation. *Optics letters*, Optical Society of America, v. 40, n. 1, p. 97–99, 2015.

READ, R. C. *Graph theory and computing*. [S.l.]: Academic Press, 2014.

RIDDER, D. de; RIDDER, J. de; REINDERS, M. J. Pattern recognition in bioinformatics. *Briefings in bioinformatics*, Oxford University Press, v. 14, n. 5, p. 633–647, 2013.

RIESEBERG, L. H. Hybrid origins of plant species. *Annual review of Ecology and Systematics*, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 28, n. 1, p. 359–389, 1997.

- ROSVOLD, J.; RØED, K. H.; HUFTHAMMER, A. K.; ANDERSEN, R.; STENØIEN, H. K. Reconstructing the history of a fragmented and heavily exploited red deer population using ancient and contemporary dna. *BMC evolutionary biology*, BioMed Central, v. 12, n. 1, p. 191, 2012.
- SALEEM, S.; ALI, A.; KHUBAIB, B.; AKRAM, M.; FATIMA, Z.; IDREES, M. Genetic diversity of hepatitis c virus in pakistan using next generation sequencing. *Journal of Clinical Virology*, Elsevier, v. 108, p. 26–31, 2018.
- SALZBURGER, W.; EWING, G. B.; HAESELER, A. V. The performance of phylogenetic algorithms in estimating haplotype genealogies with migration. *Molecular ecology*, Wiley Online Library, v. 20, n. 9, p. 1952–1963, 2011.
- SHEHAB, M. A.; GHADAWI, A. A.; ALAWNEH, L.; AL-AYYOUB, M.; JARARWEH, Y. A hybrid cpu-gpu implementation to accelerate multiple pairwise protein sequence alignment. In: IEEE. *2017 8th International Conference on Information and Communication Systems (ICICS)*. [S.l.], 2017. p. 12–17.
- SHENDURE, J.; JI, H. Next-generation dna sequencing. *Nature biotechnology*, Nature Publishing Group, v. 26, n. 10, p. 1135, 2008.
- SNEATH, P. H. Cladistic representation of reticulate evolution. *Systematic Zoology*, JSTOR, v. 24, n. 3, p. 360–368, 1975.
- SOBER, E. Parsimony in systematics: philosophical issues. *Annual Review of Ecology and Systematics*, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 14, n. 1, p. 335–357, 1983.
- SYVANEN, M. Cross-species gene transfer; implications for a new theory of evolution. *Journal of theoretical Biology*, Elsevier, v. 112, n. 2, p. 333–343, 1985.
- TAMURA, K.; STECHER, G.; PETERSON, D.; FILIPSKI, A.; KUMAR, S. Mega6: molecular evolutionary genetics analysis version 6.0. *Molecular biology and evolution*, Oxford University Press, v. 30, n. 12, p. 2725–2729, 2013.
- TEACHER, A.; GRIFFITHS, D. Hapstar: automated haplotype network layout and visualization. *Molecular Ecology Resources*, Wiley Online Library, v. 11, n. 1, p. 151–153, 2011.
- TRONTELJ, P.; MACHINO, Y.; SKET, B. Phylogenetic and phylogeographic relationships in the crayfish genus *austropotamobius* inferred from mitochondrial coi gene sequences. *Molecular Phylogenetics and Evolution*, Elsevier, v. 34, n. 1, p. 212–226, 2005.
- VESTERSTROM, J.; THOMSEN, R. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: IEEE. *Evolutionary Computation, 2004. CEC2004. Congress on*. [S.l.], 2004. v. 2, p. 1980–1987.

- VULTOS, T. D.; MESTRE, O.; RAUZIER, J.; GOLEC, M.; RASTOGI, N.; RASOLOFO, V.; TONJUM, T.; SOLA, C.; MATIC, I.; GICQUEL, B. Evolution and diversity of clonal bacteria: the paradigm of mycobacterium tuberculosis. *PloS one*, Public Library of Science, v. 3, n. 2, p. e1538, 2008.
- WANG, W.; WEI, Z.; LAM, T.-W.; WANG, J. Next generation sequencing has lower sequence coverage and poorer snp-detection capability in the regulatory regions. *Scientific reports*, Nature Publishing Group, v. 1, 2011.
- WATSON, J. D.; CRICK, F. H. Molecular structure of nucleic acids. *Nature*, v. 171, n. 4356, p. 737–738, 1953.
- WEN, D.; YU, Y.; HAHN, M. W.; NAKHLEH, L. Reticulate evolutionary history and extensive introgression in mosquito species revealed by phylogenetic network analysis. *Molecular Ecology*, Wiley Online Library, v. 25, n. 11, p. 2361–2372, 2016.
- WEST, D. B. *Introduction to graph theory*. [S.l.]: Prentice hall Upper Saddle River, 2001. v. 2.
- WOOD, L. E.; GRAVE, S. de; DANIELS, S. R. A comparative evolutionary study reveals radically different scales of genetic structuring within two atyid shrimp species (crustacea: Decapoda: Atyidae). *Zoological Journal of the Linnean Society*, 2018.
- XUE, J.; ZHANG, H.; NING, X.; BU, W.; YU, X. Evolutionary history of a beautiful damselfly, *matrona basilaris*, revealed by phylogeographic analyses: The first study of an odonate species in mainland china. *Heredity*, Nature Publishing Group, p. 1, 2018.
- YOUNG, J. *Friedrich Nietzsche: A philosophical biography*. [S.l.]: Cambridge University Press, 2010.
- ZHANG, K.; QIN, Z.; CHEN, T.; LIU, J. S.; WATERMAN, M. S.; SUN, F. Hapblock: haplotype block partitioning and tag snp selection software using a set of dynamic programming algorithms. *Bioinformatics*, Oxford University Press, v. 21, n. 1, p. 131–134, 2004.
- ZHU, J.; LIU, X.; OGILVIE, H. A.; NAKHLEH, L. K. A divide-and-conquer method for scalable phylogenetic network inference from multi-locus data. *bioRxiv*, Cold Spring Harbor Laboratory, p. 587725, 2019.