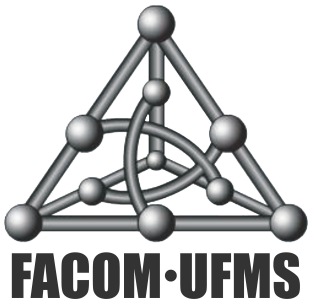
Filogenia viva baseada em distâncias

Tese de Doutorado

Graziela Santos de Araújo

Orientação: Prof. Dr. Nalvo Franco de Almeida Junior

Área de concentração: Bioinformática



Faculdade de Computação
Universidade Federal de Mato Grosso do Sul

Campo Grande, 23 de abril de 2019.

Agradecimentos

À minha família pelo apoio, dedicação, incentivo e compreensão nessa jornada.

Ao meu orientador, Prof. Nalvo Franco de Almeida Junior, pela oportunidade de aprender mais sobre bioinformática, pelo incentivo constante, pelo apoio incondicional, pela dedicação, amizade, paciência e ensinamentos que tornaram possível a conclusão deste trabalho.

Ao professor Guilherme Pimentel Telles, pelo apoio, contribuições e críticas, que foram de uma importância imensurável para a conclusão deste trabalho, sem deixar de mencionar as dicas pontuais e extremamente importantes nas otimizações dos programas construídos.

Ao professor Marcelo Henriques de Carvalho pelas conversas de corredor, por dedicar um pouco do seu tempo para discutir os problemas de otimização que foram essenciais para a solução de alguns algoritmos neste trabalho.

À professora Maria Emília M. T. Walter pelas contribuições, apoio e acolhida nas reuniões de pesquisa ocorridas no Laboratório de Biologia Molecular da UnB.

À direção da FACOM, bem como aos seus professores e funcionários, por toda ajuda e apoio, que direta ou indiretamente, contribuíram para minha formação e consequentemente para o desenvolvimento deste trabalho.

Aos membros da banca examinadora pela oportunidade de receber seus comentários e críticas e, consequentemente, melhorar meu trabalho.

Aos amigos que próximos ou mesmo distantes sempre me apoiaram para que o estímulo não acabasse.

Resumo

O problema da filogenia viva baseada em distâncias generaliza o conhecido problema da filogenia baseada em distâncias, uma vez que admite ancestrais vivos entre os objetos taxonômicos. É possível, em casos de espécies de evolução rápida, que elas coexistam e sejam ancestrais ou descendentes ao mesmo tempo, tais como vírus e objetos não-biológicos, como documentos, imagens e registros de bancos de dados. Para n objetos, a entrada do problema é uma matriz de ordem n, em que a posição i, j representa a distância evolutiva entre os objetos i e j. A saída é uma árvore sem raiz, com pesos nas arestas, onde os objetos podem aparecer como folhas ou como nós internos, e as distâncias entre pares de objetos na árvore são iguais às respectivas distâncias na matriz. Quando a matriz é aditiva, é possível encontrar tal filogenia em tempo polinomial. Neste trabalho provamos que o problema de decisão associado ao problema da filogenia viva baseada em distâncias é NP-completo quando a matriz não é aditiva e apresentamos duas heurísticas para resolver o problema no caso de matriz não-aditiva. Discutimos o problema da filogenia viva em uma abordagem na qual se busca atribuir pesos às arestas de uma topologia dada, respeitando as distâncias na matriz de entrada.

Palavras-chave: Filogenia, Evolução, Matriz Aditiva, Topologia, Heurísticas

Abstract

The Distance-Based Live Phylogeny Problem generalizes the well-known Distance-Based Phylogeny Problem by admitting live ancestors among the taxonomic objects. This problem suites in cases of fast-evolving species that co-exist and are ancestors/descendants at the same time, like viruses, and non-biological objects like documents, images and database records. For n objects, the input is an $n \times n$-matrix where position i, j represents the evolutionary distance between the objects i, j. Output is an unrooted, weighted tree where the objects may be represented either as leaves or as internal nodes, and the distances between pairs of objects in the tree are equal to the distances in the corresponding positions in the matrix. When the matrix is additive, it is possible to find such a tree in polynomial time. In this work we prove that the decision problem associated with live phylogeny problem is NP-complete when the matrix is not additive. We present two heuristics for solve the problem in the case of a nonadditive matrix. We discuss the live phylogeny problem where the problem is to assign weights to the edges of a given topology, agreeing with the distances in the input matrix.

Keywords: Phylogeny, Evolution, Additive Matrix, Topology, Heuristics

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

Os organismos existentes na Terra passam ao longo do tempo por um processo de transformação denominado *evolução*. O problema da filogenia é um dos principais problemas de Biologia, em especial de Biologia Molecular e busca explicar a história evolutiva das espécies, bem como o relacionamento existente entre as espécies atuais, com o objetivo de determinar ancestrais em comum entre elas.

Para tentar explicar a história evolutiva construímos uma árvore, denominada *árvore filogenética*, ou simplesmente *filogenia*. As folhas da árvore representam os organismos e os nós internos representam os supostos ancestrais. As arestas da árvore representam a relação evolutiva entre os organismos e ancestrais. As filogenias são construídas a partir de dados de populações, espécies, gêneros ou outros grupos de indivíduos, inclusive sequências de proteínas ou de ácidos nucléicos.

É extremamente comum, hoje em dia, o uso de sequências moleculares como fonte de dados pois, visto que a evolução ocorre nas moléculas, teremos um resultado com maior fidelidade; além do fato de haver uma grande quantidade de informação disponível em bases de dados biológicas pelo mundo, devido ao rápido aprimoramento e à diminuição de custo do processo de sequenciamento de genomas. Podemos citar como exemplos de bases de dados o GenBank [6], EMBL [54] e Swiss-Prot [4] .

A construção de filogenias é uma componente essencial em pesquisas modernas nas áreas da Medicina e da Biologia, para descobrir novas drogas, entender rapidamente as mutações de patógenos, a dispersão de espécies e a evolução dos genomas, dentre

outras aplicações. As filogenias são construídas com base em dados resultantes das comparações entre as espécies. Vamos nos referir às espécies e outras taxonomias como objetos taxonômicos, táxons ou simplesmente objetos.

Basicamente existem duas classes de problemas de filogenia, de acordo com o tipo de dado de entrada: filogenia baseada em distâncias e filogenia baseada em característica. No primeiro caso a filogenia é construída a partir das distâncias evolutivas entre objetos, enquanto que a segunda baseia-se em características que os objetos possuem. O foco do nosso trabalho é o problema da filogenia baseada em distâncias. As distâncias são uma estimativa da distância evolutiva entre os objetos.

O problema da filogenia baseada em distâncias consiste na construção de uma árvore filogenética T não enraizada, a partir de uma matriz simétrica M de distâncias entre n objetos, em que T possui exatamente n folhas representando os objetos e a distância d_{ij} entre qualquer par i, j na árvore é igual à distância M_{ij}. Tal árvore parte da premissa de que os elementos ancestrais são sempre considerados como elementos hipotéticos, que possivelmente não existem mais no presente momento.

A motivação para este trabalho, no entanto, descarta a premissa considerada no parágrafo anterior, ou seja, considera que objetos possam existir no presente momento e ao mesmo tempo serem também ancestrais de outros objetos. Com esse objetivo, Telles e colegas [93] propuseram uma nova classe de problemas de filogenia, denominada *Filogenia viva*, na qual admite-se que os objetos taxonômicos sejam representados como nós internos da árvore. A ideia é que a filogenia viva ofereça hipóteses biológicas alternativas, possivelmente explicando melhor o relacionamento entre objetos em uma população onde ancestrais e descendentes coexistam.

Podemos citar como aplicações da filogenia viva no mundo real a análise de populações de vírus ou outros organismos de evolução rápida [14, 46, 73]. As árvores filogenéticas vivas também podem ser usadas na análise de objetos não biológicos, tais como documentos, imagens e entradas de bancos de dados relacionais grandes, melhorando as técnicas de mineração para repositório de dados [20, 70].

Assim, o problema da filogenia viva baseada em distâncias, foco deste trabalho, tem como entrada também uma matriz simétrica M, contendo as distâncias entre n objetos. No entanto, a saída é uma árvore T com *no máximo* n folhas, representando os objetos de tal forma que os nós internos possam representar objetos atuais ou

ancestrais hipotéticos. A árvore construída é também não enraizada, com peso nas arestas, de forma que, as distâncias entre os objetos, sejam eles internos ou folhas, são iguais às distâncias dadas na matriz de distâncias de entrada.

Nosso trabalho consistiu primeiramente em caracterizar melhor o problema, analisando o caso quando as distâncias são aditivas e quando não são aditivas. Essa primeira caracterização foi apresentada por Araújo e colegas [1]. Em uma segunda fase, verificamos a complexidade do problema quando os dados não são aditivos, mostrando que o problema de decisão associado é NP-completo e apresentando uma heurística baseada em promoção de folhas. Esses resultados foram apresentados originalmente por Araújo e colegas [2]. Outra heurística, desta vez baseada na conhecida técnica de Neighbor-Joining e denominada Live Neighbor-Joining (LNJ), foi proposta e apresentada por Telles e colegas [94]. Além disso, analisamos o caso da filogenia viva em uma abordagem em que uma topologia também é dada como entrada e o problema consiste em atribuir pesos às arestas da árvore, de forma a respeitar as distâncias dadas na matriz.

O problema da filogenia viva baseada em características, que também foi introduzido iniciamente em [93], está fora do escopo deste trabalho e é abordado por Güths e colegas [49, 48, 38].

Este texto segue a seguinte estrutura. O Capítulo 2 apresenta o problema da filogenia tradicional baseada em distâncias, as abordagens existentes para resolvê-lo e algumas heurísticas utilizadas na sua solução. Nesse mesmo Capítulo 2, ainda descrevemos o problema da filogenia viva, abordando o problema quando as distâncias são aditivas e provamos que o problema é NP-completo quando as distâncias são não aditivas. Já no Capítulo 3 apresentamos uma heurística construída para o caso não aditivo, que é baseada na promoção de folhas a nós internos vivos. No Capítulo 4 é apresentada uma outra heurística, que segue o mesmo raciocínio da heurística Neighbor-Joining. Uma nova abordagem do problema da filogenia é apresentada no Capítulo 5, na qual dados uma topologia de árvore e uma matriz de distâncias, busca-se atribuir pesos às arestas da árvore, respeitando as distâncias dadas na matriz. Tal abordagem é apresentada levando-se em conta as filogenias tradicional e viva. Finalmente, no Capítulo 6, são apresentados alguns resultados finais e propostas de trabalhos futuros.

Capítulo 2

Filogenia baseada em distâncias

Neste capítulo vamos abordar o problema da filogenia baseada em distâncias, considerando o problema tradicional e o problema da filogenia viva. O problema tradicional considera que os elementos ancestrais são sempre hipotéticos, que possivelmente não existem mais no presente momento. Já o problema da filogenia viva admite que os objetos taxonômicos sejam representados como nós internos da árvore.

Na seção 2.1 apresentamos os principais conceitos relacionados à construção de filogenia tradicional e, na seção 2.2, os conceitos relacionados à filogenia viva.

2.1 Filogenia tradicional baseada em distâncias

O problema da filogenia tradicional baseada em distâncias consiste na construção de árvores filogenéticas utilizando dados numéricos resultantes de comparações entre n objetos. A entrada é uma matriz quadrada M de ordem n, cujo elemento M_{ij} é um número real não-negativo, chamado de **distância** entre os objetos i e j. As matrizes de distâncias entre objetos podem ser utilizadas para inferir árvores ultramétricas e árvores aditivas, que serão definidas nas Seções 2.1.2 e 2.1.3, respectivamente.

De um modo geral, segundo Catanzaro [15], não há como validar empiricamente uma filogenia candidata para um conjunto de objetos, uma vez que ninguém é capaz de observar o processo evolutivo real ao longo de milhares ou milhões de anos [43]. Por essa razão, a literatura propõe diferentes critérios para selecionar uma filogenia entre alternativas plausíveis [37]. Normalmente, esses critérios podem ser expressos

em termos de funções objetivo e as filogenias que as otimizam são referidas como ótimas [43]. Cada critério adota um conjunto de suposições, tentando descrever o processo evolucionário real e determinando a lacuna entre a filogenia ótima e a verdadeira, ou seja, a filogenia que se obteria sob essas premissas se todos os dados dos objetos estivessem disponíveis [37]. Se a filogenia ótima se aproximar da verdadeira filogenia à medida que o volume de dados dos objetos analisados aumenta, então o critério correspondente é dito estatisticamente consistente [43].

O objetivo desta seção é, no âmbito da filogenia tradicional, apresentar o critério mais comumente utilizado para construir tais árvores, além de descrever o problema e os algoritmos para a construção de árvores ultramétricas e aditivas, bem como analisar o problema da construção das filogenias quando os dados não atendem a condições de uma árvore ultramétrica ou de uma árvore aditiva.

Na seção 2.1.1 apresentamos o principal critério utilizado na construção de filogenias baseadas em distâncias, o critério da Evolução Mínima. Conceitos relacionados à matriz ultramétrica e construção de árvore ultramétrica são apresentados na seção 2.1.2, enquanto que, conceitos de matriz aditiva e construção de árvore aditiva são apresentados na seção 2.1.3. Na seção 2.1.4 são apresentadas heurísticas para resolver o problema da filogenia quando as matrizes não são aditivas e, na seção 2.1.5, são descritos outros métodos para construção de filogenias baseadas em distâncias.

2.1.1 Critério da Evolução Mínima

Os critérios baseados em distâncias visam encontrar uma filogenia que melhor se ajusta a uma determinada matriz de distâncias evolutivas entre pares de objetos. Diferentes definições de ajuste dão origem a diferentes critérios baseados em distâncias [37]. Um dos mais importantes critérios baseados em distâncias é o critério da Evolução Mínima (*Minimum Evolution* - ME), que afirma que a filogenia ótima para um conjunto de objetos é aquela cuja soma total dos pesos das arestas, de alguma maneira estimada a partir das distâncias evolutivas correspondentes, é mínima [42, 79].

Filogenias satisfazendo o critério ME são determinadas pela resolução de um Problema de Evolução Mínima (*Minimum Evolution Problem* - MEP) cuja versão depende da maneira pela qual a estimativa do peso da aresta é realizada. O critério ME tem o benefício de ser geralmente estatisticamente consistente e de exigir um

pequeno esforço computacional para estimar os pesos de aresta de uma filogenia em relação a outros critérios estatisticamente consistentes, como o da probabilidade máxima ou da inferência bayesiana. O método da probabilidade máxima (*Maximum Likelihood* - ML) estima a probabilidade de uma dada topologia de árvore ter produzido os dados observados assumindo um dado modelo de evolução, e registra a topologia que produz a maior probabilidade como a hipótese mais apropriada da história evolutiva. Métodos bayesianos tomam esta abordagem estatística um passo adiante, eles já incluem estimativas da probabilidade de que um determinado modelo tenha ocorrido, incluindo a topologia da árvore, com base em suposições anteriores [103, 15].

Em poucas palavras, o MEP consiste em, dada uma matriz de distâncias evolutivas M entre n objetos, contruir uma filogenia com peso (soma dos pesos de todas as arestas) mínimo, de forma que a diferença entre a soma das distâncias dos objetos na árvore e na matriz seja zero. No entanto, encontrar a filogenia que satisfaz o critério ME é NP-difícil [22].

O MEP pode ser dividido em dois subproblemas: o de determinar a estrutura da filogenia ótima; e o de encontrar os pesos de arestas que melhor se ajustam à matriz de distâncias M. De fato, o subproblema de estimativa de peso de aresta influencia diretamente a escolha do tipo de função que será usada para estimar a diferença entre as distâncias na árvore e na matriz, bem como a escolha da estrutura de função para se atribuir pesos às arestas e, portanto, a versão do MEP [15].

A literatura propõe duas famílias principais de modelos de estimativa de peso de aresta: os modelos de mínimos quadrados e os modelos de programação linear. Os primeiros serão mais discutidos na Seção 5.2 e os últimos não são discutidos neste trabalho e podem ser encontrados em [7, 99].

Apresentamos aqui os conceitos e algoritmos relacionados ao primeiro subproblema, o de construir a topologia da árvore filogenética, no qual estará implicitamente inserido o problema de atribuir pesos às arestas desta árvore durante sua construção. Um outro subproblema que também será abordado neste trabalho, especificamente no Capítulo 5, é o de atribuir pesos às arestas de uma dada topologia já conhecida.

Considerando o critério da evolução mínima, dois tipos de árvores filogenéticas podem ser construídas: árvores ultramétricas e árvores aditivas. A construção de

árvores ultramétricas é uma tarefa bastante interessante, pois essas árvores demonstram o tempo evolutivo decorrido de uma espécie para outra, além de mostrar quais espécies são mais próximas. Um nó interno nessa árvore representa um evento divergente, ou seja, um ponto no tempo quando as histórias evolutivas de pelo menos duas espécies divergiram.

Os dados utilizados para construir árvores ultramétricas são geralmente baseados em mutações aceitas ocorridas numa proteína, ou seja, mutações ocorridas na sequência de aminoácidos que codificam a proteína. Isto se deve ao fato de que o número de mutações aceitas em qualquer intervalo de tempo é proporcional ao tamanho daquele intervalo [47].

Os primeiros métodos para estimar o número de mutações aceitas entre duas espécies eram realizados em laboratório e utilizavam reações químicas e físicas, obtendo dados como a temperatura de fusão das hibridizações de DNA, entre outros [47]. Alguns métodos estimam o número de mutações aceitas baseando-se diretamente nas sequências de DNA ou nas sequências de aminoácidos. Para duas espécies, o número de mutações aceitas entre elas é calculado examinando as diferenças nas sequências de DNA ou nas sequências de aminoácidos codificadas para proteínas [103].

As matrizes de distâncias utilizadas para inferir árvores ultramétricas são chamadas matrizes ultramétricas e são definidas na Seção 2.1.2. Entretanto, nem sempre é possível construir uma árvore ultramétrica. Isso acontece porque os dados reais não são ultramétricos e mesmo quando são, não necessariamente refletem verdadeiramente o tempo decorrido desde a divergência [47].

Já que não podemos construir árvores ultramétricas sempre, podemos tentar construir árvores aditivas, que não indicam relações de ancestralidade ou direção de evolução das espécies, mas mostram a proximidade evolutiva entre elas. Essas árvores fornecem menos informação que as ultramétricas. As matrizes de distâncias utilizadas para inferir árvores aditivas são definidas na Seção 2.1.3.

2.1.2 Árvores e matrizes ultramétricas

Nesta seção apresentamos o conceito de árvore ultramétrica, condições para que uma matriz de distâncias possibilite a construção de uma árvore ultramétrica e também um algoritmo para tal construção.

Dada uma matriz simétrica M de distâncias para n objetos, uma **árvore ultramétrica** para M, segundo Gusfield [47], é uma árvore enraizada, com n folhas, sendo cada folha correspondente a uma linha da matriz M, portanto um objeto. Um nó interno da árvore é rotulado com uma entrada da matriz M e tem pelo menos dois filhos. Os rótulos dos nós internos são estritamente decrescentes ao longo de qualquer caminho da raiz até uma folha. E para quaisquer duas folhas i e j na árvore, M_{ij} é o rótulo do ancestral comum mais próximo de i e j. Os conceitos apresentados acima, que definem uma árvore ultramétrica, podem ser visualizados na árvore da Figura 2.1.

(a)

	A	*B*	*C*	*D*	*E*
A	*0*	*8*	*8*	*5*	*3*
B		*0*	*3*	*8*	*8*
C			*0*	*8*	*8*
D				*0*	*5*
E					*0*

(b)

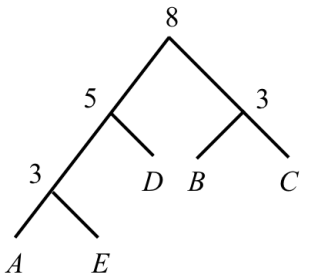


Figura 2.1: a) Exemplo de uma matriz simétrica M. b) Árvore ultramétrica para a matriz M.

Uma matriz simétrica M de números reais define uma distância ultramétrica se, e somente se, para quaisquer três índices i, j e k, o máximo entre M_{ij}, M_{ik} e M_{jk} não é único [47].

Quando M define uma distância ultramétrica, dizemos que M é uma **matriz ultramétrica**. O resultado abaixo caracteriza uma árvore ultramétrica e foi adaptado de [47] .

Teorema 2.1. *Uma matriz simétrica M admite uma árvore ultramétrica se, e somente se, M é uma matriz ultramétrica.*

Demonstração. Suponha que M admite uma árvore ultramétrica. Sejam i, j, k três objetos quaisquer. A Figura 2.2 mostra uma subárvore contendo as folhas i, j e k quaisquer. A árvore original pode conter outros nós. Como a árvore é ultramétrica, então o número escrito em u deve ser estritamente maior que o número em v. Por definição, M_{ij} é o número escrito em v e $M_{ik} = M_{jk}$. Os três valores satisfazem a condição de que o máximo não é único.

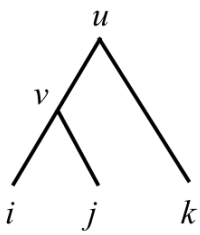


Figura 2.2: Exemplo de subárvore genérica contendo as folhas i, j e k.

Vamos provar agora que, se M é uma matriz ultramétrica, então existe uma árvore ultramétrica para M. Vamos construir uma árvore ultramétrica T a partir de M, nos concentrando inicialmente num único nó, por exemplo, a folha i. Se há d entradas distintas na linha i de M, então qualquer árvore ultramétrica T para M contém um caminho da raiz à folha i com exatamente d nós, incluindo a raiz e a folha i. Cada nó neste caminho é rotulado por uma das d entradas distintas na linha i, e estes rótulos devem aparecer em ordem decrescente no caminho.

Qualquer nó interno v neste caminho, rotulado M_{ij}, é o ancestral comum mais próximo da folha i e da folha j. Isto fixa onde a folha j deve aparecer em T, em relação ao caminho à folha i.

Dessa forma, o caminho à folha i particiona as $n-1$ folhas remanescentes em $d-1$ classes. Chamamos esta partição de $\mathcal{D}$. Duas folhas j e k estão juntas na mesma classe de $\mathcal{D}$ se, e somente se, $M_{ij} = M_{ik}$. Cada classe em $\mathcal{D}$ é definida por um nó distinto no caminho da raiz até i. O nó que define a classe contendo j, por exemplo, é o nó rotulado com M_{ij}.

Dada a partição $\mathcal{D}$ definida pelo caminho a i, basta resolver o problema da árvore ultramétrica recursivamente em cada uma das $d-1$ classes em $\mathcal{D}$ e então conectar estas árvores para formar a árvore ultramétrica para a matriz M. □

A prova do Teorema 2.1 nos fornece um algoritmo para construir uma árvore ultramétrica. Esse algoritmo foi proposto por Gusfield [47]. Além disso, de acordo com Gusfield, se D é uma matriz ultramétrica, então uma árvore ultramétrica para D pode ser construída em tempo $O(n^2)$.

Em sentido biológico, uma árvore ultramétrica tem uma taxa constante de mutação assumida ao longo das arestas. Esta propriedade é chamada de *clock* molecular, pois

podemos, a princípio, medir os tempos reais de eventos evolutivos de tais árvores. Todas as árvores ultramétricas têm uma raiz, e uma aresta da árvore é diretamente proporcional ao tempo. Na árvore da Figura 2.3 isto é representado nas arestas verticais, com os dias atuais na parte inferior e o último ancestral comum no topo. Observe que a distância evolutiva de um ancestral comum para todos os seus descendentes é a mesma.

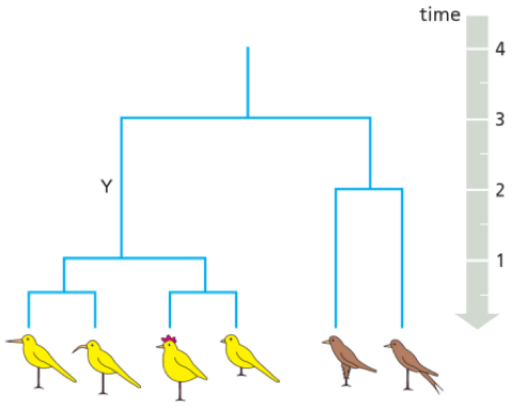


Figura 2.3: Exemplo de árvore ultramétrica. Adaptada de [103].

Como podemos ver, uma árvore ultramétrica mostra informações importantes, uma vez que podemos identificar melhor quem é ancestral de quem, qual nó veio antes, em que ponto ocorreu uma divergência e fez com que um ancestral produzisse duas espécies diferentes. Porém, as distâncias utilizadas na prática não atendem à definição de distância ultramétrica e, com isso, temos que recorrer a outra abordagem para construção de filogenias, que é a construção de árvores a partir de distâncias aditivas. Na verdade, a propriedade aditiva é uma condição mais fraca do que a ultramétrica, e será apresentada na seção seguinte.

2.1.3 Árvores e matrizes aditivas

Nesta seção apresentamos o conceito de matriz aditiva, condições necessárias e suficientes para que uma matriz de distâncias seja aditiva, além de um algoritmo para a construção de uma árvore compatível com uma matriz aditiva.

No processo de construção de árvores filogenéticas, quando trabalhamos com distâncias, o conceito de espaço métrico é necessário. Isto é importante para verificar

a qualidade dos dados e melhorar a confiabilidade da análise filogenética, segundo Dress [28].

Um espaço métrico é um conjunto de objetos O tal que para todo par $i, j \in O$, associamos um número real d_{ij} com as seguintes propriedades:

$$
\begin{aligned}
&d_{ij} > 0 \text{ para } i \neq j, && \text{(I)}\\
&d_{ij} = 0 \text{ para } i = j, && \text{(II)}\\
&d_{ij} = d_{ji} \text{ para todo } i \text{ e } j, && \text{(III)}\\
&d_{ij} \leq d_{ik} + d_{kj} \text{ para todo } i, j \text{ e } k \text{ (desigualdade triangular).} && \text{(IV)}
\end{aligned}
$$

Queremos que nossa matriz de entrada M seja tal que os objetos constituam um espaço métrico. Logo, da propriedade (III), vemos que a matriz é simétrica. A definição mais precisa da árvore que deverá ser construída com base em M vem a seguir [84, 87].

Definição 2.1. *Dado um espaço métrico de n objetos taxonômicos O, uma árvore para O é uma árvore T não enraizada com pesos nas arestas que satisfaz:*

- *T tem n folhas e cada folha corresponde a um objeto da entrada;*
- *cada nó interno tem grau 3;*
- *para quaisquer dois objetos i, j, o custo d_{ij} do caminho $(i \ldots j)$ entre eles deve ser igual a M_{ij}; e*
- *para quaisquer dois nós internos adjacentes u, v em T, $d_{uv} > 0$.*

Se tal árvore T puder ser construída, dizemos que M é uma matriz aditiva e T é uma árvore aditiva [84]. Além disso, dizemos que M e T são compatíveis.

Alguns resultados essenciais para a construção de uma árvore aditiva são apresentados a seguir e foram adaptados de [87].

Lema 2.1. *Para três objetos quaisquer existe uma única árvore aditiva T.*

Demonstração. Sejam x, y, z objetos quaisquer. Uma árvore aditiva pode ser construída como segue: crie uma aresta (x, y). Para inserir z, crie um nó c entre x e y e

uma aresta (z, c) (Figura 2.4). Basta agora notar que os pesos d_{xc}, d_{yc} e d_{zc} devem satisfazer o sistema

$$\begin{cases} d_{xc} + d_{yc} = M_{xy} \\ d_{yc} + d_{zc} = M_{yz} \\ d_{xc} + d_{zc} = M_{xz} \end{cases},$$

cuja solução é única:

$$d_{xc} = \frac{M_{xy}+M_{xz}-M_{yz}}{2}, \quad d_{yc} = \frac{M_{xy}+M_{yz}-M_{xz}}{2} \quad e \quad d_{zc} = \frac{M_{xz}+M_{yz}-M_{xy}}{2}.$$

□

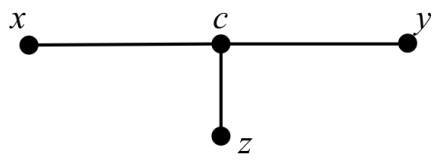


Figura 2.4: Inserção de uma folha z entre folhas x e y.

O Lema 2.2, a seguir, chamado *condição dos quatro pontos*, caracteriza uma matriz aditiva.

Lema 2.2. *(Condição dos quatro pontos - C4P) Uma matriz de distâncias M com no mínimo quatro objetos é aditiva se, e somente se, M for um espaço métrico e, quaisquer quatro objetos podem ser rotulados como i, j, k e l de tal modo que:*

$$M_{ij} + M_{kl} = M_{ik} + M_{jl} \geq M_{il} + M_{jk}.$$

Demonstração. Suponha que M seja aditiva, ou seja, admita uma árvore aditiva T. Sejam x, y, z, w folhas quaisquer de T. Seja T' a árvore obtida eliminando-se todos os vértices de T menos os pertencentes aos caminhos entre as folhas x, y, z, w. Como todos os nós internos de T têm grau 3, temos que a topologia de T' deve ser da forma como mostrada na árvore da Figura 2.5(a) e, sem perda de generalidade, podemos rotular as folhas como mostrado na árvore da Figura 2.5(b).

Como $|(u, v)| > 0$ (Definição 2.1), temos que $|(x, w)| + |(y, z)| = |(x, z)| + |(y, w)| > |(x, y)| + |(z, w)|$, ou seja, $d_{xw} + d_{yz} = d_{xz} + d_{yw} > d_{xy} + d_{zw}$.

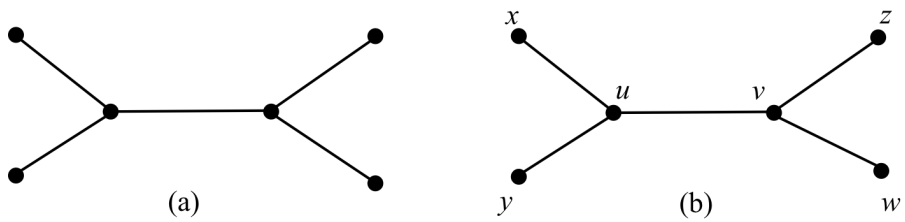


Figura 2.5: (a) Topologia de uma árvore com 4 folhas. (b) Uma possível rotulação dos nós para uma árvore com quatro folhas.

Suponha agora que seja válida a condição dos quatro pontos. Vamos demonstrar por indução que uma árvore aditiva T pode ser construída. Seja n o número de objetos taxonômicos.

- Se $n = 3$, então uma árvore pode ser construída, de acordo com o Lema 2.1.
- Suponha que exista uma árvore para $n = k$ objetos. Para adicionar um objeto z, tome duas folhas x e y quaisquer e resolva o sistema descrito no Lema 2.1, criando um nó u no caminho entre x e y e uma aresta (u, z). Há duas possibilidades para a inserção de u:
  - u não coincide com nenhum outro nó: basta demonstrar que a distância a qualquer outra folha diferente de x e y é correta. Seja w uma folha qualquer. Evidentemente os caminhos $(w, \ldots, x)$ e $(w, \ldots, y)$ devem, respectivamente, ser da forma $(w, \ldots, v, \ldots, x)$ e $(w, \ldots, v, \ldots, y)$, $v \in (x, \ldots, y)$ (se não existisse v nessas condições, x ou y não poderia ser folha). Assim, $u \in (x, \ldots, v)$ ou $u \in (y, \ldots, v)$. Como, pelo Lema 2.1, a posição de u é única e, por hipótese, vale a condição dos quatro pontos, devemos necessariamente ter $d_{uv} > 0$ e $d_{zw} = M_{zw}$.
  - u coincide com um nó v: Como u tem grau três, há três subárvores partindo de u. Seja w uma folha da subárvore que não contém x e y. Resolva o sistema descrito no Lema 2.1 para x e w, criando um nó u' em $(x, \ldots, w)$. Se u' não coincide com um nó de $(x, \ldots, w)$, crie uma aresta (u', z). Caso contrário, repita o processo até que z possa ser inserido. Para demonstrar que z deve ser inserido em alguma aresta, suponha que em duas iterações consecutivas a criação de um nó u para a inserção de z coincida com um nó v. Isso, no entanto, implica em $d_{uv} = 0$, absurdo, pois vale a condição dos quatro pontos. Como existe um número finito

de nós e em cada iteração u coincide com um nó diferente, temos que em alguma iteração z deverá ser inserido.

□

Em 1977, Waterman e colegas [99] propuseram um algoritmo cujo tempo de execução é $O(n^2)$ para construção de uma árvore compatível com uma matriz aditiva, o qual é apresentado no Algoritmo 2.1, sendo uma adaptação do descrito em [84].

Note que a inclusão de um novo nó interno, a cada passo do algoritmo, só é possível porque sempre existe uma rotulação que atende a C4P. Se nenhuma rotulação é verdadeira em algum passo, logo a matriz não é aditiva. No entanto, para testarmos a condição aditiva de M, não precisamos do algoritmo. Basta, para isso, testarmos a validade da condição dos quatro pontos para M.

No algoritmo, para todo objeto que adicionamos à árvore, podemos ter que verificar todos os outros objetos já colocados, gastando tempo constante por verificação. Isto significa que, no pior caso, o algoritmo executa em tempo $O(n^2)$.

A matriz da Figura 2.6 é aditiva, já que obedece as condições do Lema 2.2.

	A	B	C	D	E
A	0	5	11	7	10
B		0	12	8	11
C			0	8	9
D				0	7
E					0

Figura 2.6: Exemplo de matriz aditiva.

Vamos construir uma árvore aditiva para os objetos mostrados na Figura 2.6, usando o algoritmo acima. Inicialmente criamos a aresta (A,B) de tamanho 5, como mostrado na Figura 2.7(a). Em seguida, adicionamos o objeto C. Aplicando as Equações 2.1, 2.2 e 2.3, encontramos que um novo nó x_1 deveria ser criado a uma distância 2 de A e 3 de B. Além disso, adicionamos uma nova aresta (x_1,C) de tamanho 9, conforme visto na Figura 2.7(b). Considerando agora o nó D, ao aplicarmos as equações usando A e B, que já estão na árvore, encontramos que D deveria ficar na mesma subárvore, como C. Então, o novo nó interno coincidiu com o nó já

Algoritmo 2.1 CONSTROI-ARVORE-ADITIVA(M)

Entrada: Uma matriz de distâncias M entre n objetos.
Saída: Uma árvore $\mathcal{T}$ não enraizada.

Passo I: Inicialização

1: Escolha um par de objetos qualquer i e j e construa a primeira aresta da árvore, cujo peso é dado por M_{ij}.
2: Escolha um terceiro objeto qualquer k. Divida a única aresta da árvore, criando um nó interno, que chamaremos de c. Crie uma nova aresta partindo de c, onde k será colocado como folha. Devemos descobrir onde exatamente c dividirá a aresta que liga i a j. As distâncias das arestas são dadas por:

$$d_{ic} = \frac{M_{ij} + M_{ik} - M_{jk}}{2} \tag{2.1}$$

$$d_{jc} = \frac{M_{ij} + M_{jk} - M_{ik}}{2} \tag{2.2}$$

$$d_{kc} = \frac{M_{ik} + M_{jk} - M_{ij}}{2}. \tag{2.3}$$

3: $n = n - 3$

Passo II: Iteração (Adicionar o objeto $k + 1$)

4: **while** $n > 0$ **do**
5: Escolha um par de folhas u e v já adicionadas à árvore e aplique as Equações 2.1, 2.2 e 2.3 para obter d_{ux}, d_{vx} e d_{k+1x}, sendo x o novo nó interno a ser criado.
6: **if** x não coincide com qualquer outro nó na árvore **then**
7: Adicione à $\mathcal{T}$ o nó x e arestas $(u, x), (v, x), (k + 1, x)$ com pesos d_{ux}, d_{vx} e d_{k+1x}.
8: **else**
9: **repeat**
10: {A posição de x cai em um nó já existente, por exemplo u. Como u é um nó interno, sabemos que há uma subárvore cuja raiz é u. }
11: Escolha qualquer objeto pertencente a esta subárvore com raiz u, digamos r, e aplique as Equações 2.1, 2.2 e 2.3 sobre os objetos i (ou j), r e o objeto $k + 1$.
12: **until** a posição correta de x ser encontrada.
13: **end if**
$n = n - 1$
14: **end while**
15: **return** $\mathcal{T}$

existente x_1. Então aplicamos as equações novamente agora considerando B, C e D. Com isso, encontramos a posição correta do nó x_2, que está a uma distância 3 de x_1 e 6 de C. Prosseguindo com estes passos, obtemos a árvore final mostrada na Figura 2.7(d). Note que as informações contidas em uma árvore aditiva não nos

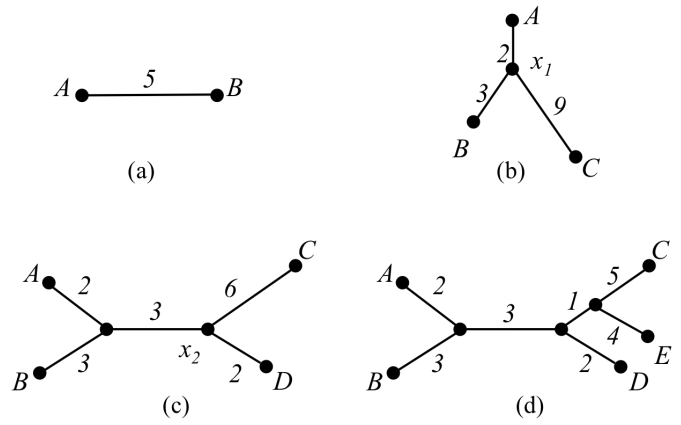


Figura 2.7: Exemplo de execução do algoritmo para a matriz da Figura 2.6.

permite inferir predição de ancestralidade, pois a árvore construída não possui raiz. Com isso, não conseguimos ver que nó é o ancestral de todos ou que nó vem antes de outro nó.

Mesmo a condição aditiva sendo mais fraca do que a condição ultramétrica, matrizes reais raramente são aditivas. Dessa forma, o problema passa a ser o de encontrar uma árvore cujas distâncias entre pares de objetos sejam mais próximas o possível das distâncias na matriz, ou seja, queremos minimizar a diferença entre as distâncias da matriz e da árvore. Este problema é NP-difícil [22] e duas heurísticas para ele serão apresentadas na Seção 2.1.4, bem como uma justificativa para a escolha delas.

2.1.4 Heurísticas para matrizes não aditivas

Os métodos baseados em distâncias comumente utilizados são *Neighbor-Joining* (NJ) e *Unweighted Pair-Group Method using Arithmetic Averages* (UPGMA). Ambos produzem uma única árvore com peso nas arestas. Outros métodos que calculam peso nas arestas a partir das distâncias evolutivas são o método dos mínimos quadrados (*least-squares method*) e o método Fitch-Margoliash; o primeiro será tratado no Capítulo 5 e o último na Seção 2.1.5.

O método NJ pertence ao grupo dos métodos de evolução mínima, e produz uma árvore aditiva quando os dados são aditivos; é derivado da suposição de que a árvore mais adequada para representar os dados será aquela que propõe a menor quantidade de evolução, mensurada como a soma total das arestas da árvore.

O método UPGMA supõe que os objetos evoluíram a uma taxa constante e igual (a hipótese do *clock* molecular), e produz árvores ultramétricas com raiz, com todos os objetos a uma mesma distância do último ancestral em comum. A seguir vamos explorar um pouco de cada método.

Neighbor-Joining

Um dos métodos mais justificados estatisticamente para aproximar uma árvore de uma matriz de distâncias é a abordagem *least squares*, segundo Shamir [45]. Na Equação 2.4, para cada par de objetos, a distância medida M_{ij} entre elas e o peso w_{ij}, intuitivamente quantificam a precisão desta medida.

O objetivo é encontrar uma árvore T cujas folhas são os n objetos dados, e que prediga as distâncias d_{ij} entre eles, tal que a expressão seguinte é minimizada.

$$SSQ(T) = \sum_{i \neq j} w_{ij}(M_{ij} - d_{ij})^2. \tag{2.4}$$

SSQ é uma medida de discrepância entre as distâncias observadas M_{ij} e as distâncias d_{ij} preditas por T . Os pesos w_{ij}, bem como a abordagem *least-square*, serão melhor abordados no Capítulo 5.

O método *Neighbor-Joining* (NJ) é um dos métodos que utiliza a abordagem acima e combina velocidade computacional com singularidade de resultados [23]. A idéia do método é unir dois objetos que, além de estarem próximos entre si, também estejam, juntos, distantes dos demais. A distância de um objeto i ao restante da árvore, denotada por u_i, é estimada pela fórmula

$$u_i = \sum_{k \neq i} \frac{M_{ik}}{(n-2)}. \tag{2.5}$$

De acordo com Nei e Saitou [82], a árvore inicial é uma árvore estrela, conforme

exemplo mostrado na Figura 2.8(a), produzida sobre a hipótese de que ainda não há nenhum par de objetos agrupados. A Figura 2.8(b), mostra o primeiro par de objetos escolhido para serem unidos. Qualquer par poderia ocupar as posições de 1 e 2 na árvore, dentre as $\frac{n(n-1)}{2}$ possibilidades. O método trabalha objetivando minimizar a soma dos tamanhos de todas as arestas da árvore. Então, escolhemos o par (i, j) cujo valor $M_{ij} - u_i - u_j$ é o menor. Esta forma de escolha do par é uma versão de NJ proposta por Studier e Keppler [90]. Uma vez escolhido o par

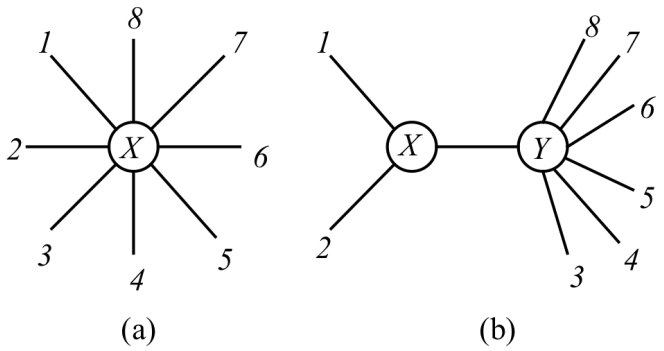


Figura 2.8: (a) Árvore estrela com nenhuma estrutura hierárquica e (b) Árvore em que os objetos 1 e 2 estão agrupados.

de objetos a ser agrupado, como é o caso do par (1-2) na Figura 2.8(b), calculamos os tamanhos das novas arestas pelo método Fitch-Margoliash, de acordo com Nei e Saitou [82], descritos na Equação 2.6. Se os objetos 1 e 2 são os escolhidos, um novo objeto é criado, resultante da junção de 1 e 2, representado por (1-2) e os objetos 1 e 2 isolados são removidos da matriz. A distância entre um novo objeto combinado, por exemplo (1-2), e outro objeto j é calculada pela Equação 2.7. O número de objetos é então reduzido de um e o procedimento é novamente aplicado para encontrar novos vizinhos. O ciclo se repete até o número de objetos tornar-se igual a dois. Se a árvore é aditiva, o método dá o tamanho correto das arestas [82].

A heurística Neighbor-Joining (Algoritmo 2.2) segue a abordagem proposta por Studier e Keppler [90] e foi melhor descrita por Felsenstein [37].

A formulacão original, proposta por Saitou e Nei [82], difere da formulação acima, também baseada na ideia de minimizar a soma dos pesos das arestas na topologia final, avaliando S, a soma mostrada na Equação 2.9, para cada par de objetos, e escolhendo o par de objetos (i, j) com o menor valor para serem agrupados como

Algoritmo 2.2 NEIGHBOR-JOINING(M)

Entrada: Uma matriz de distâncias M entre n objetos.
Saída: Uma árvore $\mathcal{T}$ não enraizada.

Passo I: Inicialização

Crie n nós rotulados $\{1, 2, \ldots, n\}$ e nenhuma aresta

Passo II: Iteração

while $n > 3$ **do**
  for $\forall$ objeto i **do**
    $u_i = \sum_{k \neq i} \frac{M_{ik}}{(n-2)}$
  end for
  Encontre $\{i, j\}$ para os quais $M_{ij} - u_i - u_j$ é mínimo.
  Una i e j em um novo objeto ij, que corresponde a um novo nó em T. Calcule o tamanho das arestas de i e j para o novo nó, da seguinte forma:

$$d_{i(ij)} = \frac{1}{2}M_{ij} + \frac{1}{2}(u_i - u_j), d_{j(ij)} = \frac{1}{2}M_{ij} + \frac{1}{2}(u_j - u_i) \tag{2.6}$$

  Compute as distâncias entre ij e cada outro objeto k:

$$M_{(ij)k} = \frac{M_{ik} + M_{jk} - M_{ij}}{2} \tag{2.7}$$

  Exclua os objetos i e j da matriz e os substitua por ij.
  $n = n - 1$
end while

Passo III: Conectando os nós restantes ($n = 2$ ou $n = 3$)

if $n = 3$ **then**
  Seja $M = \{i, j, k\}$
  Adicione à $\mathcal{T}$ o nó x e arestas (i, x),(j, x),(k, x) com pesos

$$d_{ix} = \frac{M_{ij} + M_{ik} - M_{jk}}{2}, d_{jx} = \frac{M_{ij} + M_{jk} - M_{ik}}{2}, d_{kx} = \frac{M_{ik} + M_{jk} - M_{ij}}{2} \tag{2.8}$$

else
  Seja $M = \{i, j\}$
  Adicione à $\mathcal{T}$ a aresta (i, j) com peso M_{ij}
end if
return $\mathcal{T}$

filhos de um nó ancestral x, conforme pode ser visto na Figura 2.9.

$$S_{ij} = \frac{1}{2(n-2)} \sum_{\substack{1 \le k \le n \\ k \neq i,j}} (M_{ik} + M_{jk}) + \frac{M_{ij}}{2} + \frac{1}{n-2} \sum_{\substack{1 \le k < \ell \le n \\ k,\ell \neq i,j}} M_{k\ell}. \tag{2.9}$$

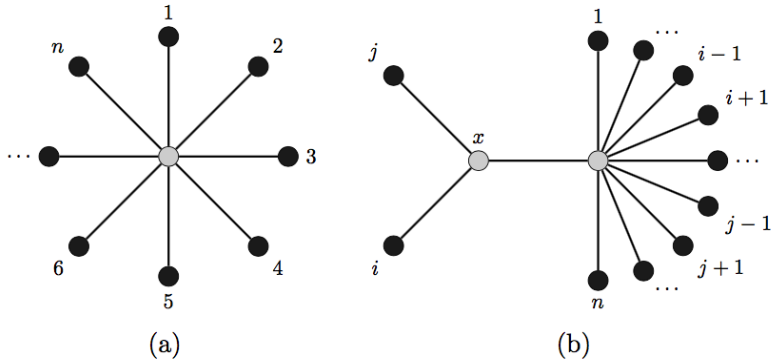


Figura 2.9: (a) Árvore estrela inicial. (b) Árvore após a escolha dos nós i, j a serem agrupados.

Neighbor-Joining possui tempo de execução $O(n^3)$. Para grandes conjuntos de dados, outras heurísticas baseadas em NJ melhoram o tempo de execução, às vezes sacrificando a precisão, por exemplo as descritas em [30, 31, 51, 56, 59, 85, 90, 100].

Segundo Peer [23], outras versões alternativas do algoritmo NJ foram propostas, incluindo BIONJ [40], NJ generalizado [72], NJ ponderado ou *weighbor* [8], NJ *maximum-likelihood* [69] e multi-NJ [21].

Por que usar Neighbor-Joining?

A heurística Neighbor-Joining é bastante utilizada nos mais diversos aspectos considerados para construção de filogenias. Existem várias frentes de pesquisas no estudo de filogenias que utilizam Neighbor-Joining por possuir algumas características que o destacam das demais heurísticas.

Wang e colegas [96] fizeram um dos primeiros estudos de métodos para construção rápida de filogenias para dados relacionados a ordem de genes, usando abordagens baseadas em distâncias e baseadas em parcimônia. Neste trabalho, o método NJ foi utilizado para a construção das árvores baseadas em *breakpoints* e inversões. O

método NJ executa muito mais rápido que todos os outros algoritmos utilizados na experiência. Em relação à exatidão topológica das árvores produzidas, o método NJ foi o melhor método baseado em distâncias, obtendo precisão igual ou superior aos outros.

Nakhleh e colegas [67] estudaram a precisão, taxa de convergência e velocidade de vários métodos de construção de filogenias rápidos, entre eles o método Neighbor-Joining. Esses métodos foram submetidos a um numeroso conjunto de sequências longas. Embora a precisão do método NJ tenha sido afetada significantemente nesses conjuntos maiores, sua velocidade ainda foi melhor que todos os outros métodos avaliados. Em estudos realizados por eles, mostraram que NJ pode recuperar a árvore verdadeira com alta probabilidade quando as sequências dadas são de tamanho limitado por uma função que cresce exponencialmente em n, sendo n o número de espécies.

Cosner e colegas [19] apresentaram uma nova heurística para construir árvores evolutivas a partir de dados da ordem de genes. Eles apresentaram e discutiram os resultados dos experimentos realizados com dados artificiais (sintéticos) e dados reais, sobre três métodos, entre eles o Neighbor-Joining. Quando as taxas de evolução são suficientemente baixas, todos os métodos recuperam boas estimativas da árvore verdadeira. Enquanto NJ executa em tempo polinomial, os outros métodos não o fazem. Também notaram que o método NJ atua tão bem quanto a nova heurística proposta por eles, em termos de precisão topológica.

Tateno e colegas [91] estudaram, por meio de simulação de computador, a eficiência relativa de alguns métodos, entre eles o *Maximum Likelihood* (ML) e o Neighbor-Joining. Foram levados em consideração se a topologia produzida é correta e a estimação do tamanho das arestas para o caso de quatro sequências de DNA de 1000 nucleotídeos. O método NJ possui uma eficiência maior em obter a árvore correta em relação aos outros modelos, mesmo quando estes produzem árvores consistentes. Além disso, o método NJ pode dar uma topologia correta mesmo quando as medidas de distâncias usadas não são estatisticamente confiáveis.

Alguns autores preocupam-se com o fato que o método NJ gera somente uma árvore final e que esta árvore pode não ser a melhor em termos do critério de evolução mínima. Simulações de computador já mostraram que na maioria dos casos a árvore NJ tem a mesma topologia que a árvore ótima ME, a menos que o número de

sequências usadas seja muito grande [78, 81].

Atteson [3] analisou a *performance* de NJ, determinando que este método faz o melhor possível para determinar a topologia da árvore entre todos os métodos baseados em distâncias. Além disso, quando a matriz de distâncias entre as espécies sendo comparadas é aditiva, NJ recupera a árvore geradora com precisão ([3] [10] [89]).

Makarenkov e colegas [60] afirmam que NJ é o mais popular entre os métodos baseados em distâncias, pois é capaz de retornar a filogenia verdadeira quando a distância observada é suficientemente próxima da distância evolutiva verdadeira. Comparado com UPGMA, NJ é projetado para corrigir as taxas desiguais de evolução em diferentes ramos da árvore. Além disso, NJ tem uma baixa complexidade e, como outros métodos baseados em distâncias, funciona bem quando a divergência entre sequências é baixa.

As matrizes de distâncias calculadas a partir de dados muitas vezes não satisfazem a condição de matriz aditiva ([34] *apud* [52]), particularmente quando as populações evoluem de maneira que não é necessariamente no formato de árvore, por exemplo, por processos de hibridização ou mistura, nos quais certos grupos descendem de pares de grupos de origem que há muito tempo foram separados. Kopelman e colegas [52] em suas pesquisas estudaram o comportamento de NJ e formalizaram as definições de três propriedades frequentemente vistas em árvores construídas por NJ na presença de tal situação, denominada mistura.

Segundo Gascuel e colegas [44], apesar de NJ se comportar bem tanto com dados simulados como com dados reais [53, 82], os fundamentos matemáticos das fórmulas de NJ só ficaram claras após investigações matemáticas ao longo dos anos. As duas principais questões eram em relação à consistência (ou seja, se NJ reconstrói corretamente uma árvore quando as distâncias se encaixam perfeitamente nessa árvore) e ao critério que NJ otimiza. As provas de consistência apresentadas em [90] foram contestadas por Mirkin [64], Gascuel [41] e Atteson [3]. Esse último ainda apresentou um resultado mais forte, mostrando que NJ reconstrói a árvore correta quando a matriz de distância é perturbada por pequenos ruídos e que NJ é ideal em relação à amplitude de ruído tolerável. Em 2005, Bryant [10] forneceu uma prova simples e elegante da consistência de NJ e, assim, a primeira questão foi colocada de lado [44].

A segunda questão era a respeito do critério sendo otimizado. Vários autores questionaram sobre o critério utilizado por NJ e escreveram que NJ não otimiza explicitamente qualquer critério [44].

Em particular, o critério utilizado por NJ, além de ser fácil de computar, é linear nas distâncias, é de permutação equivariante (a ordem de entrada dos objetos não importa), e é consistente, isto é, encontra corretamente a árvore correspondente a uma árvore que é métrica, ou seja, que atende ao conceito de árvore aditiva. Bryant [10] mostrou que tal critério é de fato um critério de seleção única satisfazendo todas essas propriedades. Gascuel e Steel [44] fizeram uma excelente revisão, providenciando uma resposta matemática muito precisa para "o que NJ faz?", por meio de uma prova que NJ é um algoritmo guloso que "diminui o comprimento da árvore" como calculado pela fórmula de Pauplin [27, 71].

Uso de NJ em vários contextos

Descrevemos aqui a utilização de NJ em vários contextos, mostrando sua aplicabilidade.

Uma estratégia para agrupar indivíduos diagnosticados com transtorno depressivo maior (*Major depressive disorder - MDD*) foi apresentada no trabalho de Yu e colegas [101]. Utilizando dados provenientes de *Whole-Genome Sequencing* (WGS), este estudo propõe uma estratégia para aplicar os dados do WGS à medicina clínica, facilitando o diagnóstico por meio do agrupamento genético. Particularmente, WGS permite identificar variantes de um único nucleotídeo (*Single Nucleotide Variant* - SNVs), que são variantes genéticas privadas, e determinar todas as variantes genéticas dentro de cada pessoa. O método usado para testar a nova métrica de distância proposta a partir de SNVs foi o método NJ, o que lhes permitiu criar árvores de *clusters* de indivíduos, confirmando a importância dos SNVs no controle de novos casos ou grupos de controle.

O método NJ foi utilizado para gerar uma filogenia representando as relações evolutivas prováveis entre as populações amostradas de cicuta (*Carolina hemlock*). A cicuta é uma espécie de árvore da família dos pinheiros e pode ser encontrada dentro de uma área limitada das Montanhas Apalaches do Sul dos Estados Unidos. Um estudo feito sobre essa espécie, que se encontra em risco de extinção por estar sendo destruída por um inseto exótico, mostrou uma alta diferenciação na espécie, mesmo

existindo em populações pequenas e isoladas na região [74].

Outra área em que NJ é utilizado diz respeito a trabalhos relacionados à identificação de novas espécies de borboletas, especificamente na delimitação molecular de espécies e *barcodes*. *DNA barcode* é uma parte do genoma do organismo que representa regiões-alvo padronizadas gênicas e/ou intergênicas em qualquer espécie biológica. Uma árvore de genes usando dados de *DNA barcode* foi construída em um dos trabalhos de Prieto e Lorenc-Brudecka [75], em que foi possível identificar uma linhagem de borboleta como sendo uma nova espécie descrita como *Rhamma dawkinsi*, oriunda da Colômbia. No trabalho de Pyrcz e colegas [76], uma árvore filogenética foi construída a partir de *DNA barcode*, em que também foi possível identificar novas espécies de borboletas. NJ repetidamente demonstrou ter um bom desempenho na delimitação de espécies e aproximação de relações filogenéticas em ambos os trabalhos.

O ChemTreeMap é uma ferramenta interativa de bioinformática [58], publicada em 2016, projetada para explorar o espaço químico e minerar as relações entre estrutura química, propriedades moleculares e atividade biológica de sequências. As três tarefas implementadas na ferramenta para organização e visualização de uma biblioteca molecular compreendem: como representar uma molécula, como quantificar as similaridades entre diferentes moléculas e como representar graficamente essas similaridades. De acordo com [58], o método NJ foi escolhido para ser utilizado na ferramenta para construir a árvore de similaridade química entre as moléculas, pelo fato de ser amplamente utilizado na construção de árvores filogenéticas para sequências grandes e diversas e também por NJ não depender de qualquer ajuste de parâmetro, tornando a construção da árvore mais robusta. Além disso, está matematicamente provado [63] que, dada uma matriz de distâncias aditiva, a árvore de saída e os comprimentos de ramificação de NJ também estarão corretos.

Outro estudo feito com cavalos-marinhos, especificamente o *Hippocampus guttulatus* de baixa diversidade e focinho longo, concluiu que tal espécie é geneticamente subdividida em cinco linhagens semi-isoladas. Tais linhagens encontram-se ao longo das costas atlânticas no sudoeste da França, no Mar Mediterrâneo e no Mar Negro. NJ também foi utilizado neste trabalho para inferir os relacionamentos evolutivos entre as amostras coletadas [77].

Filogenias podem ser construídas a partir de dados não biológicos, como é o caso

apresentado no trabalho de Zhang e colegas [102], onde delineou-se um método computacional para um estudo evolutivo de artefatos digitais. O estudo foi concentrado na evolução de uma classe de serviços da Web chamada APIs da Web que, quando juntos, permitem uma variedade de aplicativos, conhecidos como *mashups*. O algoritmo escolhido para analisar a evolução das APIs e dos *mashups* foi NJ, por dois motivos: primeiro, é muito mais rápido processar uma grande quantidade de sequências (ou seja, milhares de *mashups*). Segundo, ele pode construir uma árvore mesmo quando o modelo de evolução é desconhecido ou a correção da matriz de distância é questionável [63, 102], o que tornava o resultado do trabalho deles menos sensível a possíveis violações às suposições relativas aos dados.

Como descrito nesta seção, o algoritmo NJ apresenta uma eficiência confiável, tanto em relação à topologia construída como em sua *performance*, para dados baseados em distâncias, além de poder ser utilizado com dados biológicos ou não. Por isso, nós o utilizamos como ponto de partida para estudos sobre filogenia viva baseada em distâncias, como será explicado na seção 2.2.

UPGMA

O método UPGMA é um dos mais antigos métodos de construção de filogenias e assume a hipótese de um *clock* molecular constante, produzindo uma árvore ultramétrica como saída. Foi idealizado por Robert Soka and Charles Michener em 1958, segundo Baum e Zvelebil [103]. O nome UPGMA é um acrônimo de *Unweighted Pair-Group Method using Arithmetic Averages*, uma descrição da técnica usada [86]. A suposição do *clock* molecular, de que todas as sequências são associadas a um mesmo ponto de tempo evolutivo, tem consequências importantes, principalmente nos dias atuais, uma vez que indica que o mesmo número de substituições teria ocorrido em cada sequência desde o último ancestral em comum. Então, a distância de qualquer nó para qualquer folha que é sua descendente será a mesma para todos os descendentes. As árvores produzidas são enraizadas e ultramétricas.

No processo de construção da árvore, as duas sequências, representando os objetos, com a menor distância entre elas são assumidas terem sido as últimas a divergir, e devem, então, terem surgido a partir do nó interno mais recente na árvore. Logo, suas arestas devem ser de tamanho igual e devem possuir o valor igual à metade da distância entre si. Este é o passo inicial do método, que deve recuperar todos os nós internos na árvore em cada passo.

As sequências são agrupadas em *clusters* à medida que a árvore é construída e cada *cluster* é definido como o conjunto de todos os descendentes do novo nó já adicionado. Inicialmente, cada uma das sequências é considerada como definindo seu próprio *cluster*. Em cada passo, os dois *clusters* com menor distância evolutiva são combinados em um novo *cluster*. A árvore está completa quando todas as sequências pertencem a um mesmo *cluster*, cujo nó é a raiz da árvore.

A distância entre dois *clusters* é definida como descrito a seguir. Considere a construção de uma árvore para N sequências e suponha que em algum passo tenhamos os *clusters* X contendo N_X sequências e Y contendo N_Y sequências. A distância evolutiva entre os dois *clusters* X e Y é definida como a média aritmética das distâncias entre as suas sequências constituintes, ou seja,

$$d_{XY} = \frac{1}{N_X N_Y} \sum_{i \in X, j \in Y} d_{ij} \tag{2.10}$$

na qual i e j correspondem às sequências de X e Y, respectivamente, e d_{ij} é a distância entre as sequências i e j. Quando dois *clusters* X e Y são combinados para formar um novo *cluster* Z há uma forma eficiente de calcular as distâncias de outros *clusters*, por exemplo, W, para o novo nó. As novas distâncias podem ser definidas usando as distâncias *cluster*-a-*cluster* sem a necessidade de usar as distâncias sequência-a-sequência, através da equação a seguir:

$$d_{ZW} = \frac{N_X d_{XW} + N_Y d_{YW}}{N_X + N_Y} \tag{2.11}$$

O método é bem simples de aplicar e pode ser usado para construir árvores para conjuntos grandes de sequências. Uma ilustração do método é mostrada na Figura 2.10. No passo inicial, as sequências A e D são as mais próximas e são combinadas formando um novo *cluster*, nó V de altura $1/2$ ($= d_{AD}/2$), conforme Figura 2.10(a). Após calcularmos as distâncias de V para as outras sequências, verificamos que o próximo par a ser combinado é formado por E e V, que combinados resultam no *cluster* W, em uma altura de 1 ($= d_{EV}/2$), conforme Figura 2.10(b). Seguindo os passos desta maneira, a árvore final é obtida, conforme mostrado na Figura 2.10(e).

A principal desvantagem do UPGMA é que, se os dados não evoluíram sob condições de um *clock* molecular, os resultados podem ser seriamente enganosos [37, 103]. Um

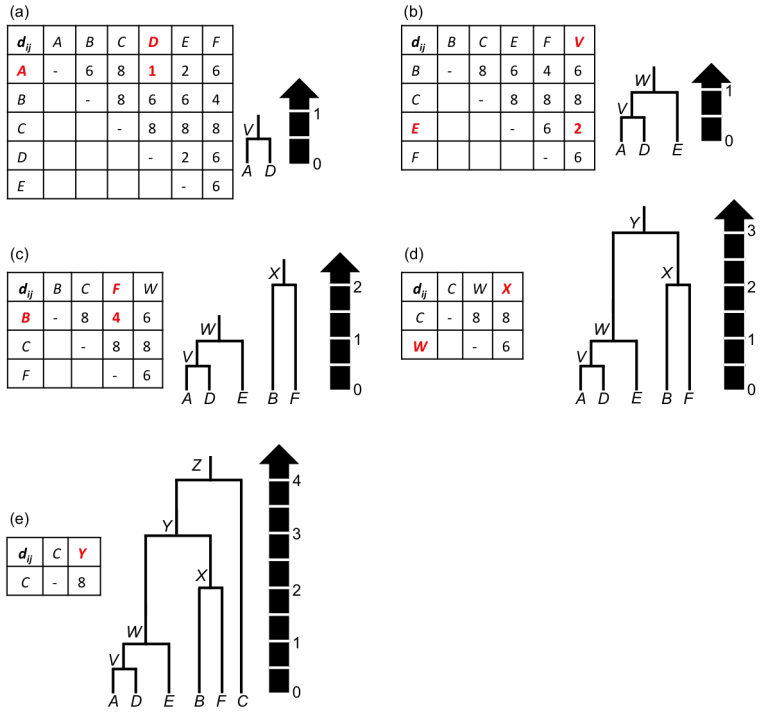


Figura 2.10: Ilustração da execução do método UPGMA. Figura adaptada de [103]. (a) Matriz de distâncias inicial mostrando que o par (A,D) são os mais próximos. O par é escolhido no primeiro passo e produz o nó interno V. A distância evolutiva é indicada pela seta preta. (b) A matriz de distâncias incluindo o nó hipotético V, de onde podemos deduzir agora que V e E são os mais próximos entre si, produzindo agora o nó interno W. (c) e (d) mostram os passos seguintes que produzem os nós internos X, Y e Z e a árvore resultante é mostrada em (e).

conjunto de dados pode, no entanto, ser testado de antemão para provável grau de compatibilidade com o método [103]. Para uma árvore ultramétrica ser apropriada para um conjunto de dados, para todos os conjuntos de três sequências A, B e C, as três distâncias d_{AB}, d_{AC} e d_{BC} deveriam ser todas iguais ou duas deveriam ser iguais e a terceira distância ser a menor. Esse é o caso do conjunto de dados da Figura 2.10.

2.1.5 Outros métodos baseados em distâncias

Antes do método NJ, outros métodos de distância aproximados foram propostos, sendo que eles foram definidos por seus algoritmos detalhados, não por um critério explícito. Tais métodos foram amplamente substituídos por NJ. Entre eles, podemos citar o método de distância Wagner, que será abordado no final desta seção.

Vamos agora olhar para um método que não faz a suposição de taxa de mutação constante, mas supõe que as distâncias sejam aditivas: é o chamado método de Fitch-Margoliash (FM).

Fitch-Margoliash

O método de Fitch-Margoliash não faz a suposição de um *clock* molecular, mas assume que as distâncias são aditivas [39]. Ele é baseado na análise de uma árvore de três folhas, como a mostrada na Figura 2.11. As distâncias d_{ij} entre as folhas A, B e C são dadas em termos dos tamanhos das arestas pelas fórmulas:

$$\begin{aligned} d_{AB} &= b_1 + b_2 \\ d_{AC} &= b_1 + b_3 \\ d_{BC} &= b_2 + b_3 \end{aligned} \tag{2.12}$$

Isto significa que a árvore está sendo tratada como aditiva. Podemos derivar diretamente as fórmulas dos tamanhos das arestas em termos de distâncias:

$$\begin{aligned} b_1 &= \tfrac{1}{2}(d_{AB} + d_{AC} - d_{BC}) \\ b_2 &= \tfrac{1}{2}(d_{AB} + d_{BC} - d_{AC}) \\ b_3 &= \tfrac{1}{2}(d_{AC} + d_{BC} - d_{AB}) \end{aligned} \tag{2.13}$$

que são estimativas dos tamanhos das arestas baseadas exclusivamente nas distâncias evolutivas [103].

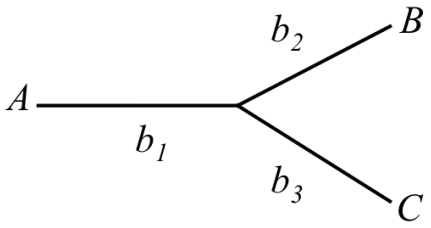


Figura 2.11: Uma árvore com três folhas da qual as Equações 2.13 do método Fitch-Margoliash são derivadas.

Árvores com mais de três folhas podem ser geradas de forma gradual, similar à usada pelo método UPGMA. Em cada passo três *clusters* são definidos, com todas as sequências pertencendo a um dos *clusters*. A distância entre *clusters* é definida (como no UPGMA) pela média aritmética das distâncias entre as sequências em diferentes *clusters* (Equação 2.10). No início de cada etapa, temos uma lista de

sequências que ainda não fazem parte da árvore que cresce e dos *clusters*, que representam cada parte da árvore que cresce. As distâncias entre todas as sequências e *clusters* são calculadas, e os dois mais relacionados (aqueles com a menor distância) são selecionados como os dois primeiros *clusters* de uma árvore de três folhas. Um terceiro *cluster* é definido para conter o restante das sequências, e as distâncias para as outras duas são calculadas. Usando as Equações 2.13, podemos determinar o tamanho da aresta do terceiro *cluster* para os outros dois *clusters*, assim como a localização do nó interno que os conecta (Figura 2.12). Estes dois *clusters* são então combinados em um único *cluster* com distância a outras sequências definidas por médias simples novamente. Existe agora uma sequência (*cluster*) a menos para ser incorporada na árvore. Repetindo tais passos, o método produz uma árvore única de maneira similar ao método UPGMA. A topologia obtida nos dois métodos é a mesma. As diferenças nas árvores são encontradas nos tamanhos das arestas e também no fato de que UPGMA produz uma árvore ultramétrica enquanto que Fitch-Margoliash produz uma árvore aditiva.

Um exemplo do método Fitch-Margoliash é mostrado na Figura 2.12, que ilustra também uma situação indesejável envolvendo a descoberta dos tamanhos de aresta entre nós internos, que é a atribuição de peso negativo à aresta.

O exemplo é adaptado de [103] e ilustra uma fraqueza do método em utilizar distâncias evolutivas diretamente para selecionar vizinhos mais próximos. Se houver taxas evolutivas muito diferentes ao longo de diferentes arestas da árvore, as duas sequências mais próximas, medidas pela distância evolutiva, podem não ser realmente vizinhas. Tal situação ocorre nos dados da Figura 2.12 para as sequências A e C. Neste caso isto leva a tamanho de aresta negativa. A história evolutiva verdadeira não pode originar uma árvore com tamanhos de arestas negativas, de forma que isto claramente é um erro do método [103]. Além disso, a árvore produzida não reproduz exatamente as distâncias entre os objetos. Há muitas ocasiões em que a variação da taxa não é tão grande a ponto de produzir esses efeitos, caso em que o método é capaz de reconstruir a árvore correta.

Analisando a Figura 2.12, em cada passo, uma árvore de três folhas equivalente à mostrada na Figura 2.11 é mostrada na árvore da esquerda. (a) No primeiro passo a menor distância é usada para identificar os dois *clusters* (A e C), que são combinados para criar o novo nó interno. Um *cluster* temporário W é definido contendo todos os demais *clusters* (exceto A e C) e as distâncias são calculadas de W para A e C,

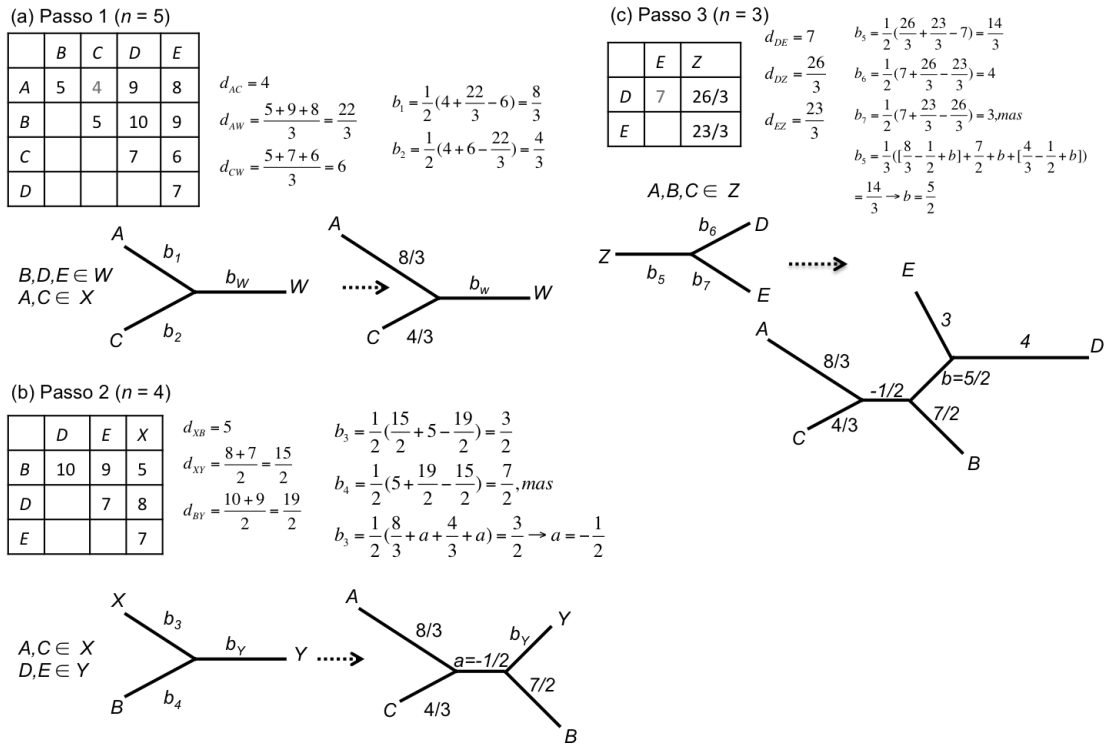


Figura 2.12: Um exemplo do método Fitch-Margoliash para reconstrução de uma árvore filogenética para cinco objetos ($n = 5$).

seguindo a Equação 2.10. Em seguida, o método usa as Equações 2.13 para calcular o peso da aresta de A e C para o novo nó interno que os conecta. (b) No segundo passo, A e C são combinados em um novo *cluster* X e as distâncias calculadas a partir dos outros *clusters*, pela Equação 2.10. Após identificar B e X como os próximos *clusters* a serem combinados e criar o *cluster* Z, um *cluster* temporário Y contém os demais objetos. Inicialmente, o valor obtido para b_3 é a distância do *cluster* X para o novo nó interno. Logo, expandindo-se o *cluster*, como mostrado na figura, calcula-se o peso da aresta rotulada com a. (c) Após combinar os objetos A, C e B em um novo *cluster* Z, restaram apenas três objetos e estes serão unidos em um novo nó interno, adicionando então D e E à árvore no passo final. Novamente, o valor obtido para b_5 corresponde à distância de todo o *cluster* Z ao novo nó interno. Desta forma, expandindo-se o *cluster*, calcula-se o peso b da aresta entre o nó interno do *cluster* Z e o nó interno ligado a D e E.

Outra aplicação do método Fitch-Margoliash é calcular tamanhos de arestas para uma dada topologia de árvore, o que é muito útil quando diferentes árvores são comparadas [103].

Método de distância Wagner

O método de distância de Farris [32] é um dos mais antigos. Está intimamente relacionado com o seu método anterior WISS (*Weighted Invariant Shared Steps*) e o seu algoritmo "método de Wagner" propõe uma construção aproximada de uma árvore mais parcimoniosa. Espécies são adicionadas a uma árvore, cada uma no melhor lugar possível, verificado pelo cálculo do aumento no comprimento da árvore causado por cada inserção possível daquela espécie [103].

Uma *árvore Wagner* para uma coleção S de objetos é uma árvore com as seguintes propriedades: $W1$, a coleção S é um subconjunto de todos os nós da árvore; $W2$, o tamanho da árvore Wagner, como definido abaixo, é menor ou igual ao tamanho de qualquer outra árvore que satisfaça a condição $W1$ [32].

O tamanho de uma árvore é definido como segue. Cada nó A da árvore é assumido ser descrito por um valor bem definido, ou estado de característica, $x(i, A)$, para cada elemento de um conjunto de características indexadas por i. A diferença fenética, que é baseada no grau de similaridade, entre quaisquer dois nós é definida como

$$D(A, B) = \sum_i |x(i, A) - x(i, B)|. \tag{2.14}$$

O tamanho de uma aresta da árvore é definido como a diferença fenética entre os dois nós que formam as extremidades da aresta. O tamanho da árvore é a soma dos tamanhos de todas as arestas da árvore. Uma árvore é dita ser mais parcimoniosa se tem tamanho mínimo de acordo com a medida definida pela Equação 2.14. Então, uma árvore Wagner é a árvore mais parcimoniosa para S e para o conjunto de características usado [32].

A função de distância definida na Equação 2.14 é uma métrica, geralmente referenciada como a métrica Manhattam, segundo Farris [32]. Sendo uma métrica, possui a propriedade da desigualdade triangular.

O algoritmo Wagner é uma técnica para calcular uma aproximação para a árvore Wagner para qualquer conjunto de objetos. A sequência de passos, apresentada em [32], é mostrada no Algoritmo 2.3. A intenção é semelhante à da evolução mínima, mas os detalhes são diferentes. Em vez de usar uma reconstrução de mínimos quadrados dos comprimentos de arestas, os comprimentos são calculados a partir

Algoritmo 2.3 ALGORITMO-WAGNER(S,D)

Entrada: (1) Um conjunto S de n objetos, (2) D uma matriz de distâncias fenéticas entre n objetos.

Saída: Uma árvore $\mathcal{T}$ não enraizada.

Passo I: Inicialização

1: Construa uma árvore com dois objetos A e B, conectados por uma aresta, sendo A e B o par de objetos com menor valor para a Equação 2.14.

2: $n = n - 2$

Passo II: Iteração

3: **while** $n > 0$ **do**

4: Selecione um objeto C ainda não adicionado à árvore.

5: Identifique a aresta da árvore com nós delimitadores denotados por G e H, que minimiza

$$D[C, (G, H)] = \frac{1}{2}[D(C, G) + D(C, H) - D(G, H)].$$

6: Construa um novo nó, denotado por F, e conecte-o por novas arestas aos nós C, G e H. Destrua a aresta conectando G a H.

7: **for** cada objeto Z em S ainda não adicionado à T **do**

8: Calcule $D(F, Z)$ usando a Equação 2.14.

9: **end for**

10: $n = n - 1$

11: **end while**

12: **return** $\mathcal{T}$

de distâncias entre pares de nós. As distâncias entre as espécies são dadas, mas aquelas entre uma folha e um nó interno, ou entre dois nós internos, também são calculadas aproximadamente. A aproximação usada assume a desigualdade triangular [103]. Assim como muitos outros métodos de matriz de distância, isso restringe o uso do método de distância Wagner para distâncias que satisfaçam a desigualdade triangular, o que muitas medidas de distância biológica não satisfazem.

Uma vez apresentados os principais conceitos envolvidos no problema geral da construção de filogenias tradicionais baseadas em distâncias, na próxima seção, vamos abordar o tópico objetivo do nosso trabalho que trata da construção de filogenias vivas baseadas em distâncias.

2.2 Filogenia viva baseada em distâncias

Este trabalho tem como foco principal abordar o problema da filogenia viva baseada em distâncias. O problema tem como entrada uma matriz de distâncias M, contendo as distâncias entre n objetos. A saída é uma árvore T, na qual os nós internos podem ser objetos atuais ou ancestrais hipotéticos. Este problema foi originalmente proposto por Telles e colegas [93]. A árvore construída é não enraizada, com peso nas arestas, de forma que as distâncias entre os objetos, que na filogenia viva podem ser nós internos ou folhas, são iguais às distâncias dadas na matriz de distâncias de entrada.

De acordo com os conceitos vistos na seção 2.1, a partir de uma matriz de distâncias entre objetos, podemos construir árvores ultramétricas e árvores aditivas. No entanto, o conceito de árvore ultramétrica não é adequado quando falamos em filogenia viva, pois de acordo com a definição, para quaisquer três objetos i, j e k, o máximo entre M_{ij}, M_{ik} e M_{jk} não pode ser único. Considerando tais objetos, teríamos os seguintes exemplos possíveis de topologia onde a condição não seria válida (Figura 2.13). Se analisarmos qualquer uma das topologias na Figura 2.13, por exemplo, aquela onde k é o nó interno vivo, teríamos $M_{ik} < M_{ij}$, $M_{jk} < M_{ij}$ e, portanto $\max\{M_{ij}, M_{ik}, M_{jk}\} = M_{ij}$ como valor único.

Já o conceito de árvore aditiva pode ser aplicado em filogenia viva e será apresentado na seção 2.2.1. Na seção 2.2.2 são apresentados conceitos e soluções para o problema quando a entrada é uma matriz não aditiva. Comentários finais sobre o problema, no caso não aditivo, são apresentados na seção 2.2.3.

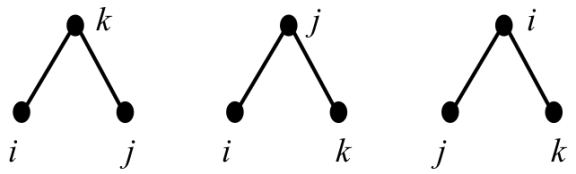


Figura 2.13: Exemplos de topologias que demonstram que a propriedade ultramétrica não é válida em filogenia viva.

2.2.1 Matrizes aditivas

A entrada para o problema da construção de filogenia viva a partir de distâncias aditivas é uma matriz M quadrada e simétrica de distâncias, em que n é o número de objetos, e M_{ij} é a distância entre os objetos i e j. A saída consiste em uma árvore sem raiz T, com pesos nas arestas. T é uma filogenia viva para M se T é compatível com M [93]. Uma árvore T é compatível com uma matriz M, denotada por $M \sim T$, se

- cada folha de T é rotulada com um objeto;
- cada objeto rotula exatamente um nó, interno ou não; e
- $d_{ij} = M_{ij}$, $1 \leq i, j \leq n$, com d_{ij} sendo a distância entre i e j em T, dada pela soma dos pesos das arestas no caminho entre i e j em T.

O problema da filogenia viva baseada em distâncias é, dada uma matriz aditiva M, construir uma filogenia viva T. Quando M é uma matriz aditiva, podemos construir tal árvore e um algoritmo polinomial foi proposto por Telles e colegas [93], cujo pseudocódigo é mostrado nos Algoritmos 2.4 e 2.5, sendo similar ao apresentado na Seção 2.1.3.

Após os algoritmos, mostramos um exemplo de execução do algoritmo para a matriz aditiva da Figura 2.14, gerando a árvore apresentada na Figura 2.15(e).

Analisando a Figura 2.15, inicialmente, o algoritmo escolhe dois objetos quaisquer, $x = 3$ e $y = 7$, por exemplo, e os une por uma aresta de peso $M_{3,7}$ (árvore (a)). Ao adicionar o próximo objeto $z = 1$ à árvore, temos que, $M_{3,1} = M_{3,7} + M_{7,1}$ e então uma nova aresta $(7, 1)$ é adicionada à árvore, obtendo a árvore mostrada em (b). O

Algoritmo 2.4 CONSTROI-FILOVIVA-ADITIVA(M)

Entrada: Uma matriz de distâncias aditiva M entre n objetos.
Saída: Uma árvore $\mathcal{T}$ viva não enraizada.

Passo I: Inicialização

1: Escolha um par de objetos i e j e conecte-os por uma aresta com peso M_{ij}.

Passo II: Adição de um terceiro objeto qualquer z

2: Sejam x e y os únicos objetos na árvore $\mathcal{T}$. Seja z um terceiro objeto qualquer.
3: **if** $M_{xy} = M_{xz} + M_{zy}$ **then**
4: Adicione z como nó interno vivo na aresta (x, y) de $\mathcal{T}$, tal que $d_{xz} = M_{xz}$ e $d_{zy} = M_{zy}$. (Caso 1)
5: **else if** $M_{xz} = M_{xy} + M_{yz}$ **then**
6: Adicione uma nova aresta (y, z) à $\mathcal{T}$, tal que $d_{yz} = M_{yz}$ e $d_{xz} = M_{xz}$. (Caso 2)
7: **else if** $M_{yz} = M_{yx} + M_{xz}$ **then**
8: Adicione uma nova aresta (z, x) à $\mathcal{T}$, tal que $d_{zx} = M_{zx}$ e $d_{yz} = M_{yz}$. (Caso 3)
9: **else**
10: {Caso 4: nenhum dos casos acima acontece}
11: Adicione um novo nó interno c na aresta (x, y) e o conecte a z, tal que

$$d_{xc} = \frac{M_{xy} + M_{xz} - M_{yz}}{2} > 0, d_{yc} = \frac{M_{xy} + M_{yz} - M_{xz}}{2} > 0,$$

$$d_{zc} = \frac{M_{xz} + M_{yz} - M_{xy}}{2} > 0$$

12: **end if**
13: $n = n - 3$

Passo III: Iteração - Adição de um novo objeto

14: $\mathcal{T}$ = ITERACAO-FILOVIVA-ADITIVA(M,n)
15: **return** $\mathcal{T}$

Algoritmo 2.5 ITERACAO-FILOVIVA-ADITIVA(M,n)

Entrada: (1) Uma matriz de distâncias aditiva M entre n objetos, (2) n - o número de objetos a serem adicionados à árvore.

Saída: Uma árvore $\mathcal{T}$ viva não enraizada.

Passo III: Adição de um novo objeto

Escolha um par de folhas x e y já adicionadas à árvore $\mathcal{T}$ {vamos denotar o único caminho conectando dois nós x e y em uma árvore por (x, y)-caminho}
while $n > 0$ **do**
  if $M_{xz} + M_{zy} = M_{xy}$ **then**
    {Caso i}
    Adicione z como nó interno vivo em (x, y)-caminho e faça $d_{xz} = M_{xz}$ e $d_{zy} = M_{zy}$.
  else if $M_{xz} = M_{xy} + M_{yz}$ **then**
    {Caso ii}
    Adicione uma nova aresta (y, z) à $\mathcal{T}$, tal que $d_{yz} = M_{yz}$ e $d_{xz} = M_{xz}$.
  else if $M_{yz} = M_{yx} + M_{xz}$ **then**
    {Caso iii}
    Adicione uma nova aresta (z, x) à $\mathcal{T}$, tal que $d_{zx} = M_{zx}$ e $d_{yz} = M_{yz}$.
  else
    {Caso iv - tente adicionar um novo nó z à $\mathcal{T}$ por uma aresta (z, c), estando c em (x, y)-caminho.}
    if $\nexists$ um nó c em $\mathcal{T}$ **then**
      {Caso iv-a}
      Crie um nó hipotético c, conecte c e z por uma aresta com peso $(M_{zx} + M_{zy} - M_{xy})/2$.
    else if $\exists$ um nó c em $\mathcal{T}$, com grau igual a 2 **then**
      {Caso iv-b: c é ancestral vivo.}
      Adicione z à $\mathcal{T}$, conecte z a c por uma aresta com peso $(M_{zx} + M_{zy} - M_{xy})/2$.
    else if $\exists$ um nó c em $\mathcal{T}$, mas c tem grau > 2 **then**
      {Caso iv-c: existe pelo menos uma folha $w \neq x, y$ conectada a c.}
      Encontre qualquer par de folhas r, s tal que c está em (r, s)-caminho e um dos casos anteriores i, ii, iii, iv-a e iv-b acontece, aplique o caso apropriado. Se tal par de folhas r, s não existe, então substitua x (ou y) por outra folha existente na árvore e aplique o caso apropriado. Se com nenhum dos pares um caso for satisfeito, então adicione z à árvore, conectando-o a c através de uma nova aresta (c, z).
    end if
  end if
  $n = n - 1$
end while
return $\mathcal{T}$

	1	2	3	4	5	6	7	8
1	0	95	100	52	82	99	82	24
2		0	31	99	69	20	13	71
3			0	104	74	35	18	76
4				0	86	103	86	28
5					0	73	56	58
6						0	17	75
7							0	58
8								0

Figura 2.14: Matriz aditiva para $n = 8$ objetos.

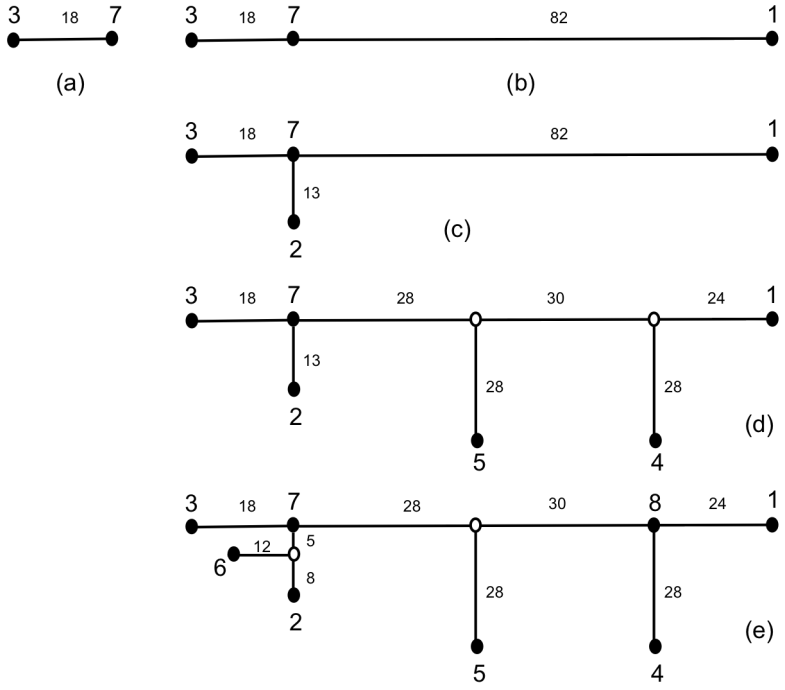


Figura 2.15: Exemplo de execução do Algoritmo 2.4 para a matriz aditiva da Figura 2.14.

próximo objeto a ser adicionado é $z = 2$, e o Caso iv deve ser aplicado. Escolhendo duas folhas quaisquer, $x = 1$ e $y = 3$, por exemplo, obtemos que um nó c deve existir no caminho (1,3) com distâncias iguais às calculadas pelas fórmulas do Caso 4. Mas este nó já existe, é o nó ocupado pelo objeto 7. Logo, Caso iv-b deve ser aplicado, onde existe um nó c com grau 2, vivo. Então, adicionamos a aresta $(7, 2)$, conforme visto na árvore (c). Ao adicionarmos o objeto $z = 4$, escolhemos $x = 1$ e $y = 2$, e novamente cairemos no Caso iv, especificamente, Caso iv-a, onde não existe um nó c no caminho $(1, 2)$ e podemos criá-lo, calculando as distâncias dos nós 1, 2 e 4 ao nó

c, da mesma forma como no Caso 4. Em seguida, vamos adicionar o objeto $z = 5$, escolhemos $x = 1$ e $y = 2$, e novamente Caso iv-a deve ser aplicado, pois não existe um nó c no caminho $(1, 2)$ e podemos criá-lo, calculando as distâncias dos nós 1, 2 e 5 ao nó c conforme apresentado no Caso 4, obtendo a árvore mostrada em (d). O próximo objeto a ser adicionado é $z = 6$. Escolhendo $x = 2$ e $y = 3$, os Casos i, ii, iii não são aplicáveis, então, o Caso iv-a pode ser aplicado, visto que não existe um nó c no caminho (2,3) com as distâncias calculadas conforme as equações mostradas no Caso 4. Basta criar as arestas conforme as equações e visualizamos este resultado na árvore (e). Finalmente, vamos adicionar o último objeto $z = 8$. Escolhendo $x = 1$ e $y = 2$, temos que $M_{1,8} + M_{8,2} = M_{1,2}$ (Caso i), então adicionamos o objeto $z = 8$ como nó interno vivo, conforme mostrado na árvore (e).

O algoritmo começa com dois objetos conectados por uma aresta e aplica, em cada passo, um dos casos descritos anteriormente. É fácil ver que o algoritmo tem tempo polinomial no número de objetos, uma vez que o teste para o caso correto pode ser feito em tempo constante, exceto para o Caso iv-c, em que é preciso encontrar um par de folhas satisfazendo algum dos outros casos. Como há $O(k^2)$ pares possíveis de folhas, em cada passo k, o tempo total do algoritmo é $O(n^3)$.

Como já mencionado, os dados na prática geralmente não atendem as condições para serem aditivos. Logo, o problema de construir a filogenia viva passa a ser o de minimizar a diferença entre as distâncias da matriz e as distâncias medidas na árvore, conforme será visto na próxima seção.

2.2.2 Matrizes não aditivas

No problema da filogenia viva baseada em distâncias, a história evolutiva dos objetos de entrada é reconstruída e tem como saída uma árvore filogenética T, não enraizada e com pesos nas arestas e onde todos os nós internos tem grau 3. Quando os dados são aditivos, a distância d_{ij} medida na árvore T entre cada par de objetos (i,j) é exatamente igual ao valor correspondente na matriz de entrada. Já quando a matriz não é aditiva, o problema consiste em encontrar uma filogenia viva que tem uma diferença mínima de M.

A partir de uma matriz M de distâncias entre n objetos, o problema da filogenia original é construir uma árvore T com exatamente n folhas tal que as distâncias em T melhor refletem as distâncias em M, de acordo com algum critério. No problema da

filogenia viva, estamos interessados em uma árvore que melhor reflita as distâncias em M, mas que tenha no máximo n folhas.

A seguir, vamos mostrar que o Problema da Filogenia Viva Baseada em Distâncias, quando os dados não são aditivos, é NP-difícil, mostrando que sua versão do problema de decisão é NP-completo (Teorema 2.2). Este resultado, que apresentamos originalmente em [2], melhora nosso entendimento do problema e sinaliza que, exceto para pequenos valores de n, ele não pode ser resolvido de maneira ótima e devemos recorrer a resultados próximos do ótimo.

Usaremos a medida Q, definida na Equação 2.15, para representar uma medida da diferença entre as distâncias em M e T:

Seja Q a medida da diferença entre as distâncias em M e T:

$$Q(M, d) = \sum_{i<j}^{n} (M_{ij} - d_{ij})^2. \tag{2.15}$$

Considere as seguintes versões dos problemas de decisão em filogenia.

Problema da Filogenia Baseada em Distâncias (PFBD)
Instância: Uma matriz $M_{n\times n}$ e um número real $K \geq 0$.
Pergunta: Existe uma árvore T com exatamente n folhas e tamanhos de caminhos d_{ij}, $1 \leq i < j \leq n$, tal que $Q(M, d) \leq K$?

Problema da Filogenia Viva Baseada em Distâncias (PFVBD)
Instância: Uma matriz $M'_{n\times n}$ e um número real $K' \geq 0$.
Pergunta: Existe uma árvore T' com no máximo n folhas e tamanhos de caminhos d'_{ij}, $1 \leq i < j \leq n$, tal que $Q(M', d') \leq K'$?

A prova de que o PFBD é NP-completo foi mostrada por Day [22].

Teorema 2.2. *PFVBD é NP-completo.*

Demonstração. Vamos reduzir o PFBD ao PFVBD como parte da prova de que PFVBD também é NP-completo. A redução é trivial: uma instância (M, K) de PFBD é transformada para uma instância (M', K') de PFVBD fazendo $M' = M$ e $K' = K$. É fácil observar que a redução é feita em tempo polinomial.

Se a resposta para uma instância do PFBD é `sim`, então existe uma árvore T com exatamente n folhas e pesos d_{ij} nas arestas para cada par de objetos i e j tal que $Q(M, d) \leq K$. É fácil ver que a árvore T também é uma resposta para PFVBD.

Reciprocamente, se a resposta para uma instância do PFVBD for `sim`, existe uma árvore T' com pesos d'_{ij} para cada par de objetos i e j tal que $Q(M', d') \leq K'$. Uma solução correspondente para o PFBD pode ser obtida de acordo com um dos casos a seguir:

- T' não tem nós internos vivos. Fazendo $T = T'$ temos uma árvore com exatamente n folhas, $K = K'$ e $Q(M, d') \leq K$, o que permite responder `sim` ao PFBD.
- T' tem nós internos vivos. Construa uma árvore T aplicando a seguinte operação enquanto existir um nó interno vivo em T'. Seja x um nó interno vivo qualquer representando o objeto u. Escolha uma aresta $e = (x, z)$ incidente a x, adicione um novo nó hipotético y dividindo a aresta e e adicione uma aresta conectando y a um novo nó folha rotulado com u. Seja a o peso da aresta e. Atribua os pesos $d'_{xy} = d'_{yu} = 0$, e $d'_{yz} = a$ para as novas arestas. Finalmente, transforme x em um nó hipotético. A Figura 2.16 ilustra esta operação. Quando nenhum nó interno vivo for encontrado mais, T terá exatamente n folhas, e como as distâncias entre os pares em T são as mesmas distâncias dos pares em T', temos que $Q(M, d) \leq K$. Então, T é uma resposta `sim` para PFBD.

Portanto, podemos concluir que PFBD é redutível polinomialmente a PFVBD, e então o Problema da Filogenia Viva Baseada em Distâncias é NP-difícil. É direto verificar uma solução para PFVBD em tempo polinomial, e então PFVBD pertence a NP. Concluímos que PFVBD é NP-completo. □

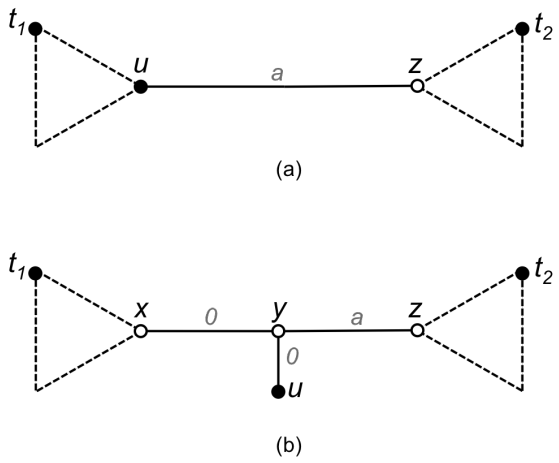


Figura 2.16: Transformação de um nó interno vivo u (a) em uma folha (b) em T'. Todas as distâncias entre os pares são preservadas.

2.2.3 Comentários

Uma vez que o problema da filogenia viva baseada em distâncias para o caso não aditivo é NP-difícil, passaremos a apresentar no próximo capítulo heurísticas para tentar resolvê-lo. Uma das heurísticas é baseada em promoção de folhas, ou seja, folhas com determinadas características são promovidas a nós internos. Outra heurística é baseada em uma ideia similar à utilizada no método NJ. Ambas são apresentadas, respectivamente, nos Capítulos 3 e 4.

Tanto a prova de NP-completude apresentada aqui neste capítulo, quanto a heurística baseada em promoção de folhas, foram originalmente apresentadas em [2].

Capítulo 3

Heurística baseada em promoção de folhas

Conforme visto no Capítulo 2, o problema de construir filogenias vivas com dados não aditivos é NP-difícil, o que nos remete ao desafio de projetar heurísticas eficientes.

Uma das heurísticas mais utilizadas para resolver o problema original da filogenia baseada em distâncias, em que não há nós internos vivos, é o método Neighbor-Joining (NJ) [82, 90], apresentado no Capítulo 2. Para o problema de filogenia original, NJ reconstrói a árvore correta quando a matriz é aditiva. Mais ainda, quando a matriz não é aditiva, mesmo que não haja um limitante para medir a qualidade das distâncias na árvore resultante, NJ cria a topologia certa quando as distâncias na matriz são suficientemente próximas das distâncias evolutivas verdadeiras [3, 63].

Neste capítulo, apresentamos a heurística denominada *Live-promotion*, publicada em [2], que utiliza como entrada a árvore obtida por NJ para a matriz não aditiva e faz pequenos ajustes, promovendo folhas com determinada característica a nós internos vivos. A escolha de quais são as folhas candidatas é feita com base no peso das arestas no entorno dessas folhas.

Na seção 3.1 apresentamos o ponto chave da nossa heurística, que é a escolha da folha para ser promovida a nó interno vivo. A heurística e sua complexidade são apresentadas na seção 3.2. Testes comparativos realizados com a heurística baseada em promoção de folhas e com a heurística NJ são apresentados na seção 3.3.

3.1 Folhas candidatas à promoção

Para filogenias vivas, o método NJ trabalha muito bem no processo de reconstrução de filogenias a partir de matrizes aditivas, em termos de distâncias, mas não cria nós internos vivos. Todos os objetos são sempre folhas. Nós mostramos que, para cada folha u na árvore produzida por NJ, que deveria ser um nó interno vivo, NJ cria arestas com peso zero na vizinhança de u, conforme mostrado no Teorema 3.1. Baseado neste resultado, podemos realizar pequenas mudanças naquela vizinhança, promovendo u a um nó interno vivo. Este é o ponto chave da nossa heurística descrita aqui.

Teorema 3.1. *Seja M uma matriz aditiva com pelo menos 4 objetos, T uma filogenia viva e T_{NJ} uma árvore NJ, ambas compatíveis com M. Para cada objeto u que é um nó interno em T, existem nós internos adjacentes v e w em T_{NJ} tais que as arestas (u, v) e (v, w) tem peso zero.*

Demonstração. Suponha que M possua exatamente 4 objetos. Logo, a árvore produzida por NJ, T_{NJ} é única e podemos rotulá-la, como mostrado na Figura 3.1. A filogenia viva T também é única, como mostrado na Figura 3.2.

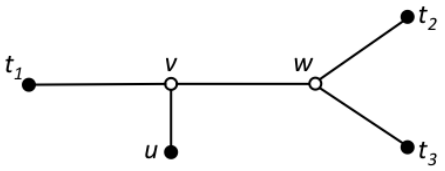


Figura 3.1: T_{NJ} é a árvore filogenética produzida por NJ para 4 objetos.

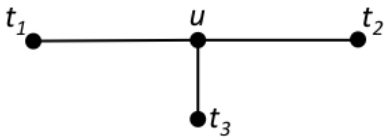


Figura 3.2: Árvore filogenética viva T com 4 objetos. Nó interno vivo u tem grau 3.

Uma vez que todas as distâncias em ambas as árvores são compatíveis com M, vamos denotá-las por d. Seja u o nó interno em T. O objeto u é uma folha em T_{NJ}, adjacente a v, o qual é adjacente a w, como representado. De T_{NJ},

$$d_{u,t_1} + d_{u,t_2} - d_{t_1,t_2} = 2d_{u,v}.$$

Tomando as distâncias em T, temos

$$d_{u,t_1} + d_{u,t_2} = d_{t_1,t_2}.$$

Assim, $d_{u,v} = 0$. Usando uma relação similar, em T_{NJ} temos que

$$d_{u,t_2} + d_{u,t_3} - d_{t_2,t_3} = 2d_{u,w},$$

e $d_{u,w} = 0$.

Suponha agora que existam mais que 4 objetos, então existem 3 subárvores separadas por u, com folhas t_1, t_2, t_3, t_4, conforme mostrado na Figura 3.3. Observe que cada subárvore tem pelo menos uma folha, e uma delas tem que ter mais de uma folha.

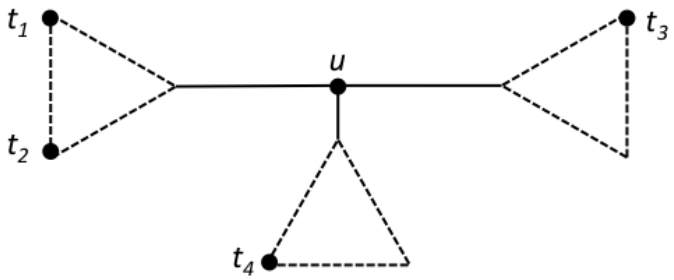


Figura 3.3: Árvore filogenética viva T com mais de 4 objetos.

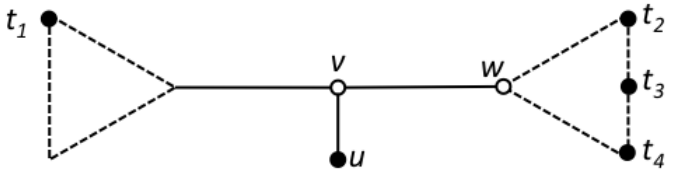


Figura 3.4: Filogenia T_{NJ} produzida por NJ para mais de 4 objetos, quando três objetos estão reunidos em uma subárvore.

Note que, até rotulá-las, NJ produzirá uma das árvores mostradas nas Figuras 3.4 e 3.5. Se NJ produzisse a árvore na Figura 3.4, então temos

$$d_{u,t_1} + d_{u,t_3} - d_{t_1,t_3} = 2d_{u,v}.$$

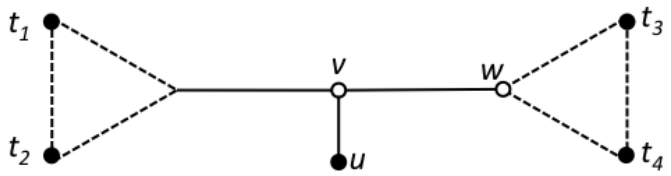


Figura 3.5: Filogenia T_{NJ} produzida por NJ para mais de 4 objetos, quando há dois objetos em duas subárvores separadas.

Tomando as distâncias em T, obtemos

$$d_{u,t_1} + d_{u,t_3} = d_{t_1,t_3}.$$

Assim, $d_{u,v} = 0$. Usando uma relação similar, em T_{NJ} temos que

$$d_{u,t_2} + d_{u,t_3} - d_{t_2,t_3} = 2d_{u,w},$$

e $d_{uw} = 0$. Se NJ produzir a árvore na Figura 3.5, podemos aplicar o mesmo raciocínio para concluir que $d_{u,v} = d_{u,w} = 0$. □

O Teorema 3.1 nos indica como usar NJ para resolver o Problema da Filogenia Viva quando a matriz de entrada é aditiva: para cada folha u e nós internos v, w tais que os pesos das arestas (u, v) e (v, w) são iguais a zero, contraia ambas as arestas fazendo $u = w$, tornando u um nó interno vivo.

A heurística *Live-promotion* usa a mesma ideia para matrizes não aditivas. Inicialmente, NJ é executado. Se a matriz é aditiva, todas as contrações possíveis são feitas, conforme explicado anteriormente para obter a filogenia viva. Se a matriz não é aditiva, então não há garantia de que NJ produzirá tão caracterizadamente a folha u e os nós v, w, como mencionados no Teorema 3.1. No entanto, a heurística *Live-promotion* procura por uma folha u e nós internos y, z, com a mesma topologia de u, v, w no caso aditivo, mas desta vez tendo arestas (u, y) e (y, z) com pesos suficientemente pequenos.

Seja M uma matriz de distâncias não aditiva com pelos menos 4 objetos. Seja T_{NJ} a árvore obtida de NJ, tal que existe uma folha u com a configuração mostrada na Figura 3.6, em que z, y são nós hipotéticos e os pesos das arestas a, b, c incidentes em y são tais que $b, c < a$. A folha u poderá ser promovida a nó interno vivo, conforme

veremos na próxima seção.

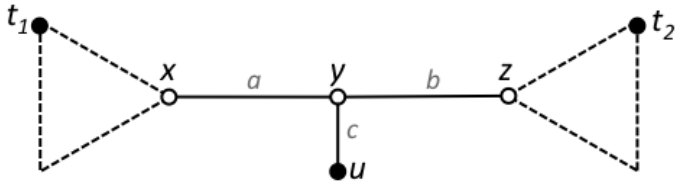


Figura 3.6: Árvore NJ. O objeto u é uma folha e $b, c < a$.

3.2 A heurística

O primeiro passo da heurística baseada em promoção de folhas consiste em executar NJ. Após NJ ser executado, a heurística visita as folhas da árvore. Seja T a árvore atual no início de um passo qualquer da *Live-promotion*. Tome uma tripla qualquer u, y, z, como mencionado acima, e verifique se a contração das arestas (u, y) e (y, z) (fazendo $z = u$ e tornando u um nó interno vivo) cria uma árvore T' melhor que T, como mostrado na Figura 3.7. Se T' é melhor que T, faça a contração e siga para o próximo passo, novamente procure outra tripla u, y, z da mesma maneira. O cálculo das distâncias em T', incluindo o peso e da aresta (x, u), bem como a decisão de quando T' é melhor que T serão explicados em subseções a seguir.

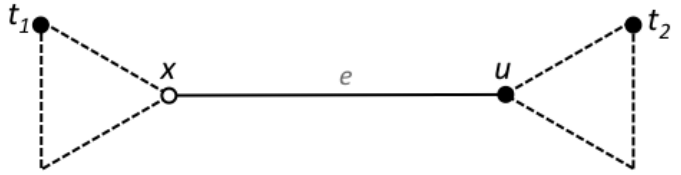


Figura 3.7: Árvore T'. Após a contração de arestas de T, u torna-se um nó interno vivo.

Calculando as distâncias em T'

É importante observar que as arestas $(x, y), (y, z), (u, y)$ em T, com pesos a, b, c, respectivamente, foram substituídas pela aresta (x, u) com peso e em T'.

Seja $d'_{i,j}$ a distância entre os objetos i e j em T'. Na nova árvore T', fazemos $e = a + \frac{b+c}{2}$ e atualizamos algumas distâncias de T':

- para cada par de objetos t_1, t_2 separados por y em T,

$$d'_{t_1,t_2} = d_{t_1,t_2} + \frac{c-b}{2};$$

- para cada objeto t_1 em T,

$$d'_{u,t_1} = d_{u,t_1} + \frac{b-c}{2}; \quad \text{e}$$

- para cada objeto t_2 em T,

$$d'_{u,t_2} = d_{u,t_2} - (b+c).$$

Observe que apenas essas distâncias são modificadas e precisam ser calculadas.

Ao escolher este tipo de subárvore, onde $b, c < a$, e calculando as distâncias como mostrado acima, estamos tentando nos aproximar da situação que tínhamos no caso aditivo, isto é, $b, c = 0$.

Avaliação e complexidade

Seja $d_{i,j}$ a distância entre os objetos i e j em T. A medida de variação entre as distâncias em M e em T é feita de acordo com a Equação 2.15(página 40).

A árvore T' é dita melhor que T se $Q' \leq Q + \delta$, com Q' calculado substituindo d por d' na Equação 2.15, e δ é um parâmetro que permite *Live-promotion* ser mais ou menos estrito. Se $Q' \leq Q + \delta$, substituímos T por T' e continuamos o processo para cada folha remanescente, até que não encontramos mais uma tripla u, y, z como mencionado acima, ou até que $Q' > Q + \delta$, para todas as folhas remanescentes. Ao término desse processamento, temos a árvore viva final T_F.

Vamos analisar os passos de *Live-promotion* e verificar sua complexidade. O passo inicial de *Live-promotion* é executar NJ que tem tempo de execução $O(n^3)$ [82]. Em seguida, cada um dos $O(n)$ passos de *Live-promotion* leva tempo $O(n^2)$ para calcular as distâncias em T', tempo constante para verificar se uma folha satisfaz a condição mencionada pelo Teorema 3.1 (ou, se $b, c < a$), e tempo $O(n^2)$ para aplicar a Equação 2.15 para calcular Q'. Assim, o tempo de execução de *Live-promotion* é $O(n^3)$.

3.3 Resultados

Para avaliarmos e validarmos a nova heurística proposta *Live-promotion*, realizamos uma comparação de sua *performance* com a do NJ, quando aumentamos o grau de não aditividade dos dados. Por *performance* devemos entender como a habilidade de minimizar o escore $Q(M, d)$, de acordo com a Equação 2.15.

Para isso, criamos conjuntos de matrizes não aditivas, agrupadas de acordo com três parâmetros: o número de objetos, o índice de não aditividade, explicado a seguir, e o percentual de nós internos vivos.

Foram construídos dois conjuntos de dados. No primeiro conjunto, avaliamos a *performance* de NJ, à medida que aumentávamos o número de objetos e o índice de não aditividade. No segundo conjunto, avaliamos a *performance* de *Live-promotion* conforme aumentávamos os três parâmetros.

Índice de não aditividade

Conforme visto na Seção 2.1.3, Lema 2.2 (C4P, pág. 13), uma matriz de distâncias M é aditiva se seu conjunto de objetos satisfaz as propriedades de um espaço métrico e também a condição dos quatro pontos. Então, para ser um espaço métrico, para qualquer tripla i, j, k, $M_{ij} \leq M_{ik} + M_{kj}$ (desigualdade triangular). Já a condição dos quatro pontos declara que, dado qualquer quádrupla de objetos, podemos rotulá-los i, j, k e l tal que $M_{ij} + M_{kl} = M_{ik} + M_{jl} \geq M_{il} + M_{jk}$.

Seja M uma matriz de distâncias. Seja α' o número de triplas em M que não satisfazem a desigualdade triangular e β' o número de quádruplas que não satisfazem a condição dos quatro pontos. O índice de não aditividade I_N de M é definido como

$$I_N = \frac{\alpha'/\alpha + \beta'/\beta}{2}, \tag{3.1}$$

em que α e β representam o número total de triplas e quádruplas de M, respectivamente. O valor de I_N é tal que, $0 \leq I_N \leq 1$.

Avaliação de *performance* de *Live-promotion* e NJ

Foram construídos dois conjuntos de matrizes não aditivas. O conjunto construído para avaliar a *performance* de NJ consiste de matrizes não aditivas agrupadas de

acordo com o número de objetos N ($N = 10, 20, ..., 100$) e seus I_N, nos seguintes intervalos $(0, 0.25], (0.25, 0.5], (0.5, 0.75], (0.75, 1]$. Para cada valor de N e cada intervalo de I_N, um subconjunto de 100 matrizes foram geradas.

Cada matriz de entrada é gerada através da criação de uma árvore randômica, e em seguida, uma matriz é gerada dessa árvore. Logo, por construção, tal matriz é aditiva. Na sequência, essa matriz é perturbada, escolhendo-se aleatoriamente uma tripla i, j, k e fazendo $M_{ij} = M_{ik} + M_{kj} + \delta$. Esta alteração, com certeza, muda α' e possivelmente muda β', o que consequentemente modifica o índice de não aditividade I_N. Nesses experimentos foi utilizado $\delta = 1$.

As Figuras 3.8, 3.9 e 3.10 mostram a variação de $Q(M, d)$ para as árvores construídas por NJ quando o índice I_N aumenta, para 10, 50 e 100 objetos, respectivamente. Como era esperado, quanto maior o valor de I_N, pior é a *performance* de NJ. Outro ponto importante a ser elucidado é que o escore produzido por NJ tende a crescer mais rápido quando o número de objetos N cresce.

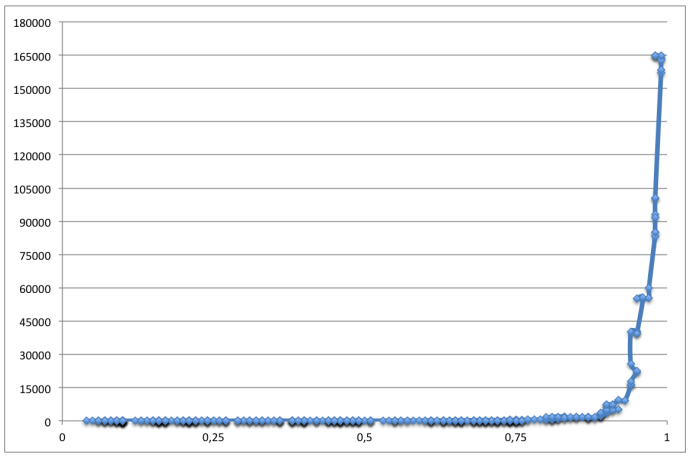


Figura 3.8: Escore NJ por I_N, para $N = 10$ objetos.

Para avaliarmos *Live-promotion*, utilizamos o mesmo método usado para avaliar o desempenho de NJ. Porém, é necessário considerar outro parâmetro na construção do conjunto de matrizes: o percentual de nós internos vivos. Além de controlar o valor de N e I_N, utilizamos também o percentual de nós internos vivos $P = 20\%, 40\%, 60\%, 80\%$ sobre o número de folhas. Neste caso, as matrizes aditivas foram geradas a partir de árvores aleatórias contendo $N = 10, 20, \ldots, 100$ folhas mais P por cento (sobre N) de nós internos vivos. Logo, para cada $N = 10, 20, \ldots, 100$, I_N em $(0, 0.25], (0.25, 0.5], (0.5, 0.75], (0.75, 1]$ e $P = 20\%, 40\%, 60\%, 80\%$ sobre N, um subconjunto de 100 matrizes foi gerado.

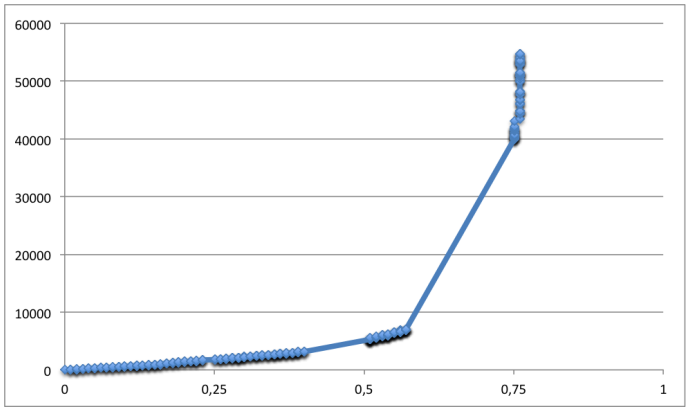


Figura 3.9: Escore NJ por I_N, para $N = 50$ objetos.

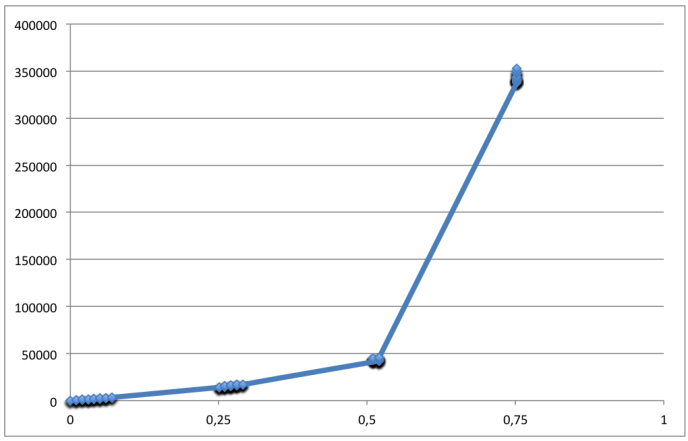


Figura 3.10: Escore NJ por I_N, para $N = 100$ objetos.

Os resultados para $N = 10$, 50 e 100 são mostrados nas Figuras 3.11, 3.12 e 3.13, respectivamente. Cada figura mostra o escore produzido por *Live-promotion* para todos os valores de P. Analisando os mesmos intervalos de I_N, verifica-se que *Live-promotion* apresenta uma *performance* melhor, em relação a NJ, até mesmo quando o percentual de nós internos vivos é alto.

Esta heurística e todos os resultados apresentados neste capítulo foram publicados em uma conferência internacional no ano de 2017 [2].

No próximo capítulo, uma nova heurística para o problema da filogenia viva será apresentada, em que testes foram realizados com dados biológicos reais e também com dados não biológicos.

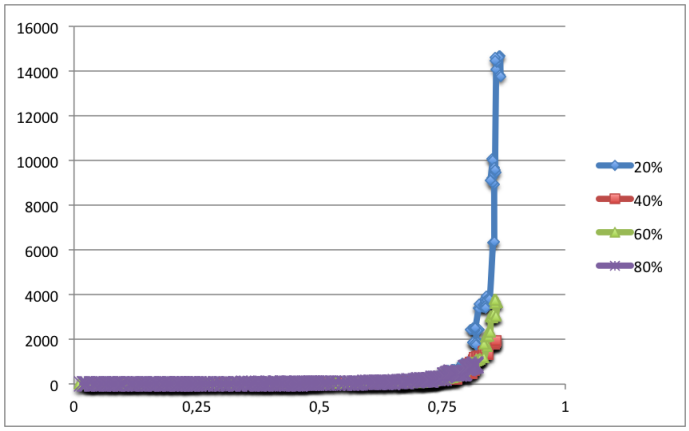


Figura 3.11: Escore *Live-promotion* por I_N, para 10 folhas mais $P\%$ de nós internos vivos.

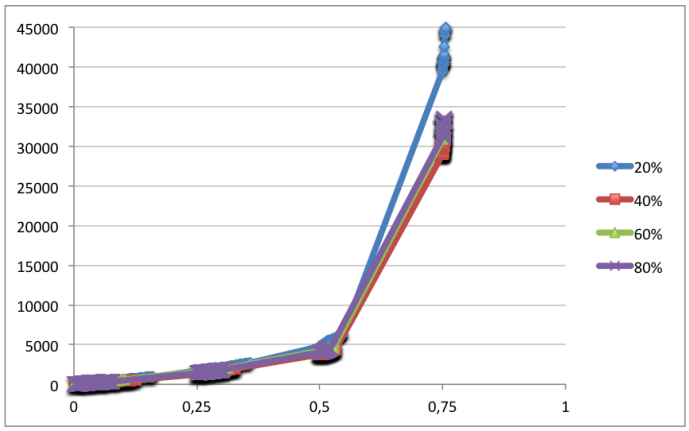


Figura 3.12: Escore *Live-promotion* por I_N, para 50 folhas mais $P\%$ de nós internos vivos.

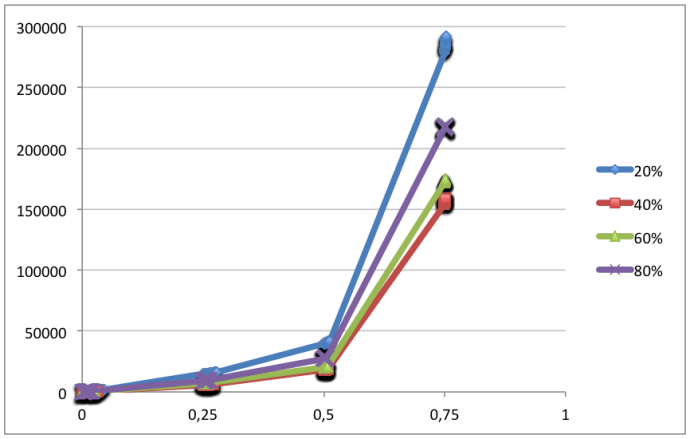


Figura 3.13: Escore *Live-promotion* por I_N, para 100 folhas mais $P\%$ de nós internos vivos.

Capítulo 4

Live Neighbor-Joining

Live Neighbor-Joining (LNJ) é o nome dado para uma outra heurística a ser apresentada neste capítulo, que foi projetada para a construção de filogenias vivas baseadas em distâncias para o caso em que a matriz de distâncias (M) entre objetos não é aditiva. A descrição da heurística, juntamente com os testes realizados usando dados de vírus, bactérias e dados não biológicos, foi publicada por Telles e colegas [94].

A heurística é uma extensão do raciocínio numérico utilizado em NJ, porém introduzindo o caso em que um nó interno vivo pode originar uma filogenia com soma de arestas com valor menor. Foi aplicada sobre diferentes conjuntos de genomas virais e bacterianos, introduzindo diferentes hipóteses para a relação dessas espécies. Além disso, será ilustrado ao longo do capítulo o uso de LNJ em conjuntos de dados não biológicos.

Na seção 4.1 descrevemos a heurística e sua complexidade. Nas seções 4.2, 4.3 e 4.4 são apresentados testes realizados sobre LNJ utilizando dados de vírus, bactérias e dados não biológicos, respectivamente.

4.1 A heurística

A heurística consiste em realizar junções de objetos por um ancestral comum, sendo que este pode ser um ancestral vivo ou hipotético. Se o nó ancestral for hipotético procede-se como no caso NJ, conforme visto na Figura 2.9, Seção 2.1.4, utilizando a Equação 2.9. Caso contrário, o nó ancestral sendo vivo, uma nova base matemática

será utilizada e é o que explicaremos a seguir.

Observando a Figura 4.1(a), temos n objetos formando uma estrela, e dois objetos i e j sendo agrupados como filhos de um outro objeto k (Figura 4.1(b)). Neste caso, a soma de arestas será dada por:

$$L_{ik} + L_{jk} + \sum_{\substack{1 \leq v \leq n \\ v \neq i,j}} L_{vx}.$$

O escore T é definido como

$$T_{ijk} = M_{ik} + M_{jk} + \frac{\sum_{\substack{1 \leq u < v \leq n \\ u,v \neq i,j}} M_{uv}}{n-3} \tag{4.1}$$

A ideia de LNJ é escolher, em cada iteração, o par ou tripla de objetos com menor

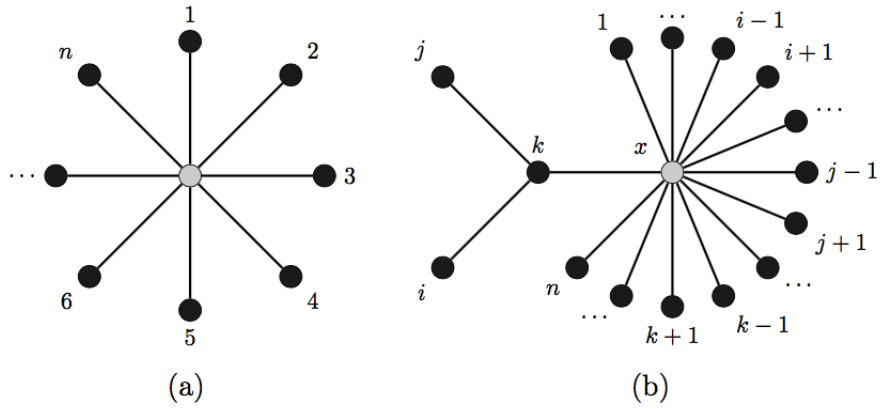


Figura 4.1: (a) Uma árvore estrela com n objetos como folhas. (b) Um ancestral vivo k é adicionado entre os objetos i e j.

escore. Quando um par de objetos é selecionado, ele funciona como na heurística NJ. Já quando uma tripla é selecionada, os objetos i e j serão removidos do conjunto de objetos e k será o nó ancestral vivo de i e j, e os tamanhos de arestas serão:

$$L_{ik} = M_{ik} \text{ e } L_{jk} = M_{jk}.$$

A heurística terminará quando restarem três nós, e estes serão conectados através de um nó hipotético, como acontece em NJ. Caso restem apenas dois nós, eles serão conectados por uma aresta (i, j), com peso M_{ij}. O pseudocódigo para LNJ é mostrado no Algoritmo 4.1.

Algoritmo 4.1 LIVE-NEIGHBOR-JOINING(M,n)

Entrada: Uma matriz de distâncias M entre n objetos.
Saída: Uma árvore $\mathcal{T}$ não enraizada.

Passo I: Inicialização
Crie uma floresta $\mathcal{T}$ com n nós rotulados $\{1, 2, \ldots, n\}$ e nenhuma aresta
Passo II: Iteração
while $n > 3$ **do**
  Encontre $\{i, j\}$ com escore S mínimo (Eq. 2.9)
  Encontre $\{u, v, w\}$ com escore T mínimo (Eq. 4.1) em que w é um objeto numa folha
  if $S_{ij} < T_{uvw}$ **then**
    Adicione um nó x e arestas (i, x) e (j, x) com pesos como nas Eqs. 2.6 à $\mathcal{T}$
    Remova i e j de M
    Adicione x à M e calcule M_{xk} para $k \in M$
    $n = n - 1$
  else
    Adicione as arestas (u, w) com peso M_{uw} e (v, w) com peso M_{vw} à $\mathcal{T}$
    Remova u e v de M
    $n = n - 2$
  end if
end while
Passo III: Conectando os nós restantes ($n = 2$ ou $n = 3$)
if $n = 3$ **then**
  Seja $M = \{i, j, k\}$
  Adicione à $\mathcal{T}$ o nó x e arestas (i, x),(j, x),(k, x) com pesos como nas Eqs. 2.8 à $\mathcal{T}$
else
  Seja $M = \{i, j\}$
  Adicione à $\mathcal{T}$ a aresta (i, j) com peso M_{ij}
end if
return $\mathcal{T}$

Tabela 4.1: Tempos de Execução. Média de tempo em segundos de NJ e LNJ para números diferentes de espécies. Uso de memória em bytes para NJ e LNJ.

n	tempo de NJ (s)	tempo de LNJ (s)	pico de memória (B)
128	0,0032	0,0318	115.763
256	0,0254	0,4234	360.755
512	0,2020	6,1179	1.244.275
1024	1,6120	92,0992	4.584.227
2048	12,9061	1.425,4783	17.557.539
4096	103,3255	22.410,6324	68.669.987

A entrada consiste em uma matriz de distâncias M de ordem n, em que n representa o número de objetos do conjunto analisado. Se a soma de todas as distâncias em M e a soma das distâncias de cada nó para os outros são mantidas pelo algoritmo, e se um vetor de *flags* é usado para acompanhar os ancestrais vivos, para que um objeto não seja escolhido duas vezes como nó interno, então avaliar T a cada iteração leva tempo $O(n^3)$, sendo que a atualização de M e as somas levam tempo $O(n)$. Então, LNJ é executada em $O(n^4)$ e requer um espaço adicional $O(n)$.

LNJ foi testada sobre diferentes e crescentes tamanhos de conjuntos de entrada, bem como seu tempo de execução foi avaliado. Foram geradas aleatoriamente dez matrizes aditivas de cada tamanho, definido nos experimentos, calculando-se a média do tempo de execução. Visando simular NJ para compará-lo a LNJ, forçamos o LNJ a sempre escolher um par de nós ao invés de uma tripla. Conforme pode ser visto na Tabela 4.1, os tempos de execução são estáveis, crescendo por fatores próximos a 8 para NJ e próximos a 16 para LNJ, quando o tamanho da entrada é duplicado. Os testes foram executados em um sistema com processador Intel Xeon E5-2630-v3 2,40 GHz com *cache* de 20MB, 384 Gb de RAM e um HD SATA de 13Gb, com GNU/Linux 64-bit (Debian 8, kernel 3.16.7). Os códigos-fontes foram compilados por GCC 4.9.2 com -O3. A Tabela 4.1 também mostra o uso de memória, que é a mesma para NJ e LNJ.

Nas próximas seções, serão apresentados testes realizados com dados reais envolvendo vírus e testes com dados não biológicos.

4.2 Testes com vírus

O RNA de vírus consiste em um ambiente muito adequado para testar abordagens de filogenia viva, pois apresentam as mais altas taxas de mutação entre os organismos, evoluindo muito rapidamente e possivelmente coexistindo com outros [46, 65, 73]. Vamos apresentar os resultados da aplicação de LNJ sobre três conjuntos diferentes de genomas de vírus: *Zika*, *Chikungunya* e *Ebola*.

A entrada para cada conjunto é uma matriz de distâncias dos genomas, construída através da utilização do pacote MUMi [25], que gera o chamado índice MUM de distância genômica para cada par de genomas, baseado em critérios de diversidade, como a quantidade média de nucleotídeos iguais e as proporções de DNA compartilhados por ambos os genomas. O índice MUM é calculado após a execução do MUMmer [24], uma ferramenta muito popular para o alinhamento de pares de genoma baseado em árvores de sufixos e sementes denominadas MUMs (*Maximal Unique Matches*). Os valores de MUMi estão sempre no intervalo [0, 1] e são inversamente proporcionais ao número de MUMs encontrados entre os dois genomas. Assim, quanto maior o MUMi, mais distantes são os genomas comparados [83].

Todas as matrizes de distâncias usadas aqui, bem como as árvores correspondentes, que foram geradas no formato Newick [33], além do código-fonte de LNJ, estão disponíveis no website `https://git.facom.ufms.br/bioinfo/LNJ`. As árvores mostradas nas próximas seções foram desenhadas usando os conhecidos pacotes FigTree (`http://tree.bio.ed.ac.uk/software/figtree`), PhyD3 (`https://phyd3.bits.vib.be`), GraphViz dot (`https://www.graphviz.org`) e D3.js (`https://d3js.org`).

Vírus da Zika

O vírus da Zika (*Zika* virus) é um RNA vírus da família *Flaviviridae*, gênero *Flavivirus*, transmitido pelo mosquito *Aedes*, como o *Aedes aegypti* e *Aedes albopictus*. Surtos desse vírus foram recentemente registrados nas Américas e na África. Como a infecção pelo vírus da Zika durante a gravidez tem sido associada a defeitos congênitos, tal como a microcefalia, esta atraiu considerável atenção da comunidade científica. Lanciotti e colegas [55], por exemplo, apresentaram uma filogenia de 20 cepas de vírus da Zika, derivada pelo método NJ. Nesta seção, apresentamos uma

filogenia viva, como topologia alternativa para o mesmo conjunto de sequências de genomas.

Para construir a filogenia viva dessas sequências, uma matriz de distâncias para os mesmos 20 genomas de vírus da Zika foi construída usando o *pipeline* descrito anteriormente, usando MUMi. Os tamanhos dos genomas variam de 10.247 a 10.807 bases. Uma filogenia para esta matriz foi construída usando o método NJ, e é mostrada na Figura 4.2. A filogenia viva construída por LNJ é mostrada na Figura 4.3.

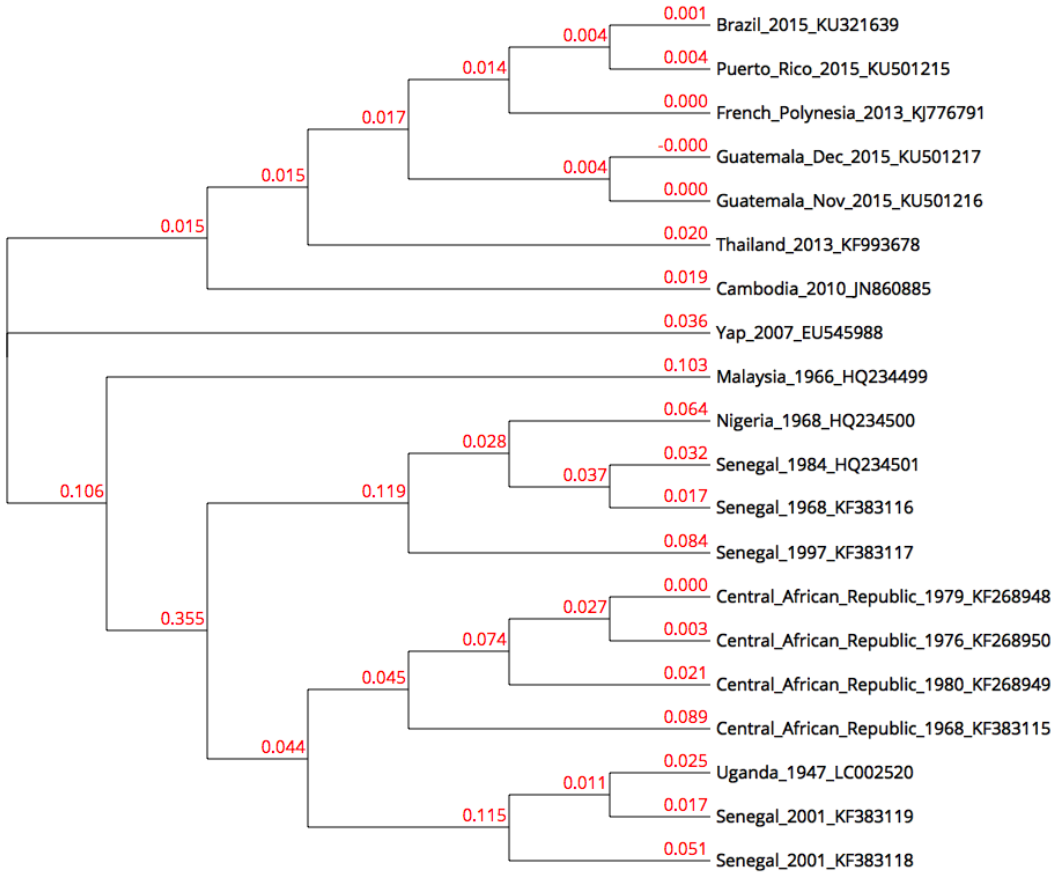


Figura 4.2: Árvore obtida por NJ para 20 sequências do vírus da Zika.

Os grupos East African, West African e Asian, identificados no trabalho de Lanciotti e colegas [55], estão agrupados na árvore obtida por NJ, exceto a sequência Yap 2007, que foi posicionada mais longe de outros membros do grupo asiático. Já na árvore obtida por LNJ, tem-se o grupo East African como uma subárvore distinta enraizada por um membro do grupo West African, cujos demais membros também formam uma subárvore distinta. Na árvore de LNJ, sete sequências tornaram-se

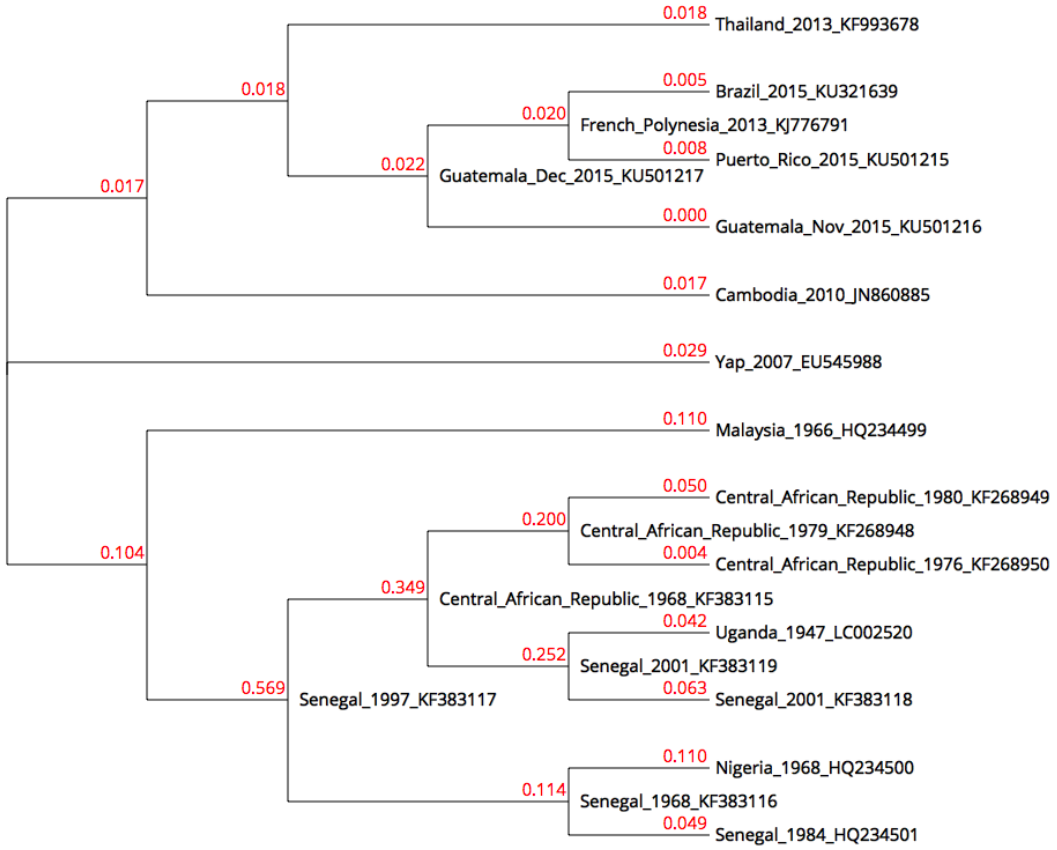


Figura 4.3: Árvore obtida por LNJ para 20 sequências do vírus da Zika.

ancestrais vivas, introduzindo a hipótese que pode ser considerada em uma análise mais profunda dos alinhamentos desses genomas.

O agrupamento dos ancestrais vivos preditos por LNJ (Figura 4.3) não foi alterado em relação à topologia predita por NJ (Figura 4.2). LNJ na verdade melhorou a topologia de NJ, sugerindo como as populações de vírus estão evoluindo. O vírus Zika foi descoberto na África na década de 1950 e todos os isolados africanos estão agrupados nas análises de NJ e LNJ. No entanto, LNJ sugeriu que a sequência KF383117 corresponde ao precursor do vírus que circula hoje. Curiosamente, a sequência KF383117 foi registrada em 1997, mais tarde do que outros isolados que datam de 1968. Como o LNJ coloca o KF383117 no primeiro nó de uma subárvore africana, é razoável considerá-lo como a sequência mais próxima do ancestral comum de sequências africanas analisadas neste trabalho.

Supõe-se que a epidemia do vírus da Zika em 2015 na América do Sul é devida à chegada de atletas da Polinésia que desembarcaram no Brasil para um campeonato mundial de canoagem [66]. Os dados da Figura 4.3 apoiam tal hipótese, uma vez que a sequência KJ776791 da Polinésia Francesa é colocada como ancestral vivo da sequência brasileira e da sequência de Porto Rico. No entanto, a sequência da Polinésia Francesa pode ter evoluído de uma sequência americana anterior, como sugere a sequência KU501217 da Guatemala, sendo que o isolado polinésio do vírus Zika pode ser oriundo da América, e não da Ásia ou da África.

Vírus Chikungunya

O vírus Chikungunya é outro importante vírus recentemente anotado. É da família *Togaviridae*, gênero *Alphavirus*. Os principais sintomas desta infecção são febre e dores nas articulações. As mesmas fêmeas que transmitem o vírus Zika disseminam o vírus Chikungunya. É por isso que esse vírus também atraiu a atenção dos pesquisadores.

Nunes e colegas [68] investigaram as origens e o potencial de propagação do vírus Chikungunya no Brasil desde os primeiros casos confirmados em 2014. Segundo eles, quatro genótipos foram identificados desde 1952, nomeados pelas regiões onde foram encontrados: *East-Central-South-African (ECSA)*, *West African*, *Asian* e *Indian Ocean (IOL)*.

Filogenias baseadas em sequências genômicas completas foram estimadas usando verossimilhança máxima em [68]. Foi utilizado um total de 76 genomas representando todos os quatro genótipos virais: 11 do genótipo *West African*, 12 do *ECSA*, 17 do *IOL*, e 30 do *Asian*, além de 6 novas linhagens brasileiras. O tamanho dos genomas varia de 11.569 a 12.189 bases. Nunes e seus colegas concluíram que as cepas associadas aos surtos da fase inicial no Brasil pertencem aos genótipos *Asian* e *ECSA*.

Utilizando a heurística LNJ, uma filogenia viva foi construída com 74 das 76 sequências usadas por Nunes e colegas, uma vez que duas delas (CNR20235 e CNR20236) não foram encontradas no Genbank, mesmo seguindo as indicações dos autores. A filogenia viva obtida é mostrada na Figura 4.4, contendo 27 nós internos vivos.

A Figura 4.5 mostra o agrupamento dos genótipos em ambas topologias, de Nunes e

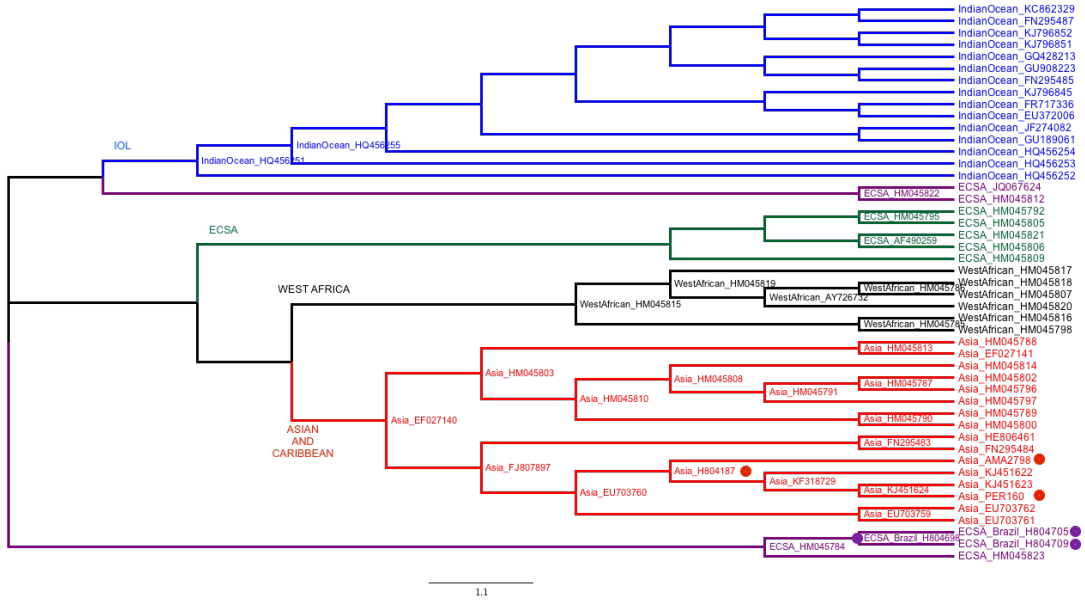


Figura 4.4: Árvore LNJ para 74 genomas do vírus Chikungunya. Os pontos representam cepas brasileiras (as cepas de cor púrpura são baianas e as vermelhas são de Pernambuco e do Pará).

colegas e a nossa. Embora o genótipo ECSA2 na topologia de Nunes (Figura 4.5(a)) tenha se separado em dois grupos vizinhos em nossa árvore (Figura 4.5(b)), as linhagens brasileiras H804698, H804705 e H804709 (Feira de Santana, Bahia) foram agrupadas de forma melhor, com H804698 sendo ancestral dos outros, como mostrado na Figura 4.4. Ao mesmo tempo, as distâncias na topologia LNJ mostram quão próximos estão ECSA2a e ECSA2b.

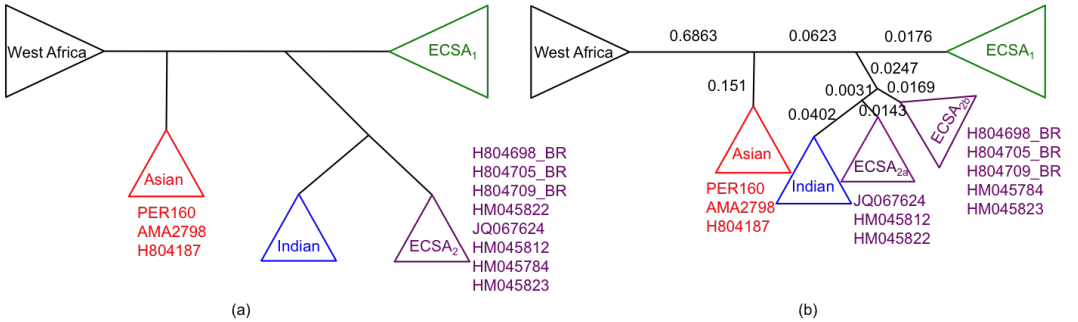


Figura 4.5: (a) Topologia de alto nível de Nunes *et al.* [68] e (b) Topologia gerada por LNJ. Linhagens brasileiras da Bahia (pontos em cor púrpura na Figura 4.4) foram separadas das do grupo $ECSA_2$ (pontos vermelhos) em nossa árvore.

No trabalho de Nunes e colegas, propõe-se que as sequências de Chikungunya de isolados brasileiros são derivadas dos genótipos Asian e ECSA. Entre as sequências classificadas como asiáticas, apenas uma foi considerada como um caso autóctone

(AMA2798). Curiosamente, a análise do LNJ sugere que ela é derivada de H804187, uma sequência isolada de um paciente que viajou da ilha caribenha Guadalupe para a cidade de Belém no Brasil, embora os dados do LNJ sugiram que a infecção do paciente P37 (AMA2798) tenha sido derivada do vírus importado pelo paciente P34 (H804187). Além disso, a sequência PER160 (P25) não deve ser relacionada ao vírus que circula em Belém, pois parece derivar do KJ45164, uma sequência isolada na Ilha Virgem das Caraíbas, um fato que está de acordo com dados epidemiológicos descritos em [68], que mostra que um paciente de fato viajou para a República Dominicana.

A utilização da heurística LNJ pode ajudar a melhorar a investigação epidemiológica, sugerindo uma cadeia mais precisa de infecção de um surto de vírus. Nos casos autóctones de Feira de Santana [68], a análise do LNJ sugere que os vírus que infectaram os pacientes P38 (H804709) e P39 (H804705) tinham um ancestral comum, H804698, que infectou o paciente P36, embora este último possa ter sido infectado por um vírus parental que infectou ainda os outros dois pacientes, todos vivendo na mesma área geográfica. Portanto, os dados de LNJ podem ser explorados para entender como um certo vírus evolui ao longo de uma cadeia de infecção.

Ebola

Ao contrário do Zika e Chikungunya, o vírus Ebola, que causa a doença Ebola (anteriormente conhecida como Febre Hemorrágica Ebola), é transmitido entre humanos por contato físico direto com fluidos corporais infectados, principalmente sangue, fezes e vômito. O primeiro surto conhecido ocorreu no Zaire em 1976. O último relatado foi na República Democrática do Congo, em maio de 2017 [17].

Dudas e colegas [29] propuseram uma árvore de verossimilhança máxima (com *bootstrap*=100) de 49 cepas do vírus Ebola. Eles usaram várias sequências de genoma do Genbank (incluindo Bundibugyo BDBV, Reston RESTV, Sudan SUDV, Tai Forest TAFV e Zaire EBOV) e as sequências do surto de Guiné em 2014. Os comprimentos das sequências genômicas variam de 18.774 a 18.961.

A partir das mesmas sequências, LNJ foi executado e gerou a filogenia mostrada na Figura 4.6, com todos os clados apresentados em [29] mantidos, exceto para a troca da cepa KC242800 e o clado de Luebo. Além disso, a cepa de Gabão (1994) foi posicionada no clado de Kikwit. O clado BDBV foi mantido com dois nós internos

vivos, o clado RESTV com 3 nós internos vivos e o clado EBOV com 7 nós internos vivos.

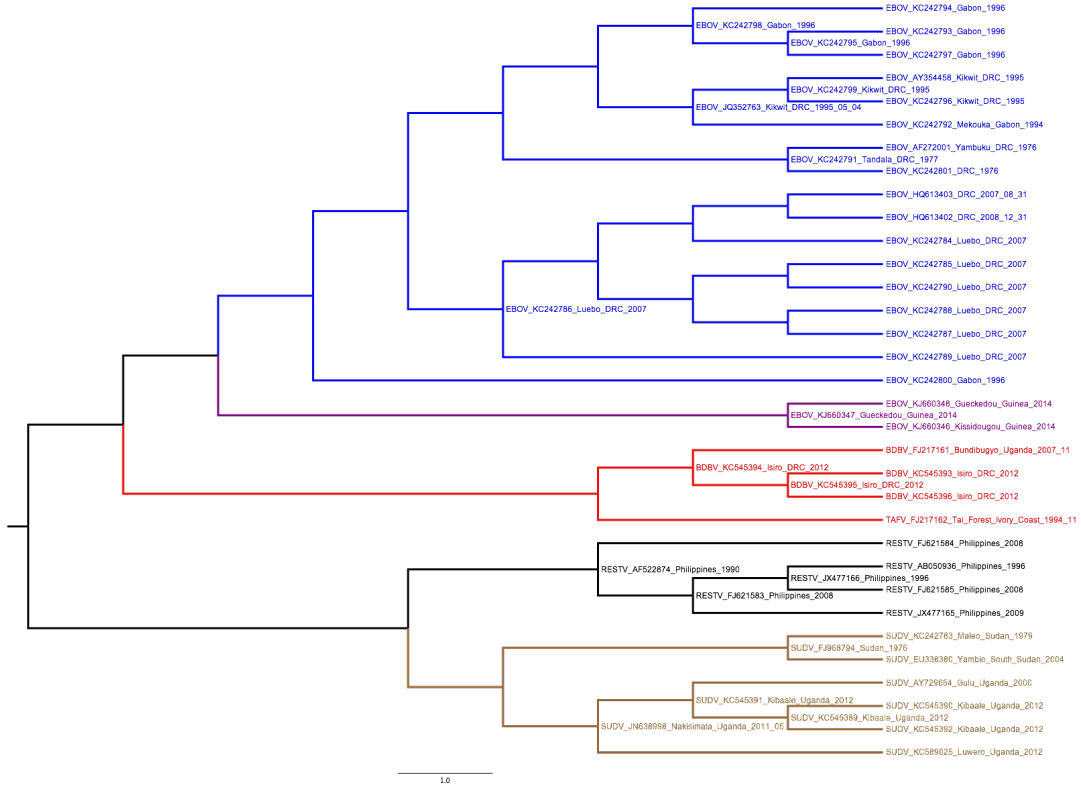


Figura 4.6: Árvore LNJ para as 49 sequências do vírus Ebola usadas em [29].

A cepa do vírus Ebola Reston é a única espécie não africana conhecida do Ebola, e foi agrupada como um clado único pelo LNJ. A subárvore marcada em preto na Figura 4.6 é, essencialmente, a árvore observada por Carroll e colegas [13], obtida usando análise de coalescência bayesiana. As diferenças são a previsão das sequências JX477166, FJ621583 e AF5222874 como nós internos. Além disso, o LNJ prediz que FJ621584, embora ainda seja um *outgroup* para o vírus Reston, evoluiu de AF5222874, que aparece como o nó mais interno dessa subárvore, embora o LNJ pareça tentar agregar uma dimensão temporal na filogenia biológica.

4.3 Teste com bactérias

Filogenia viva, apesar de parecer apropriada para organismos de evolução rápida, como vírus, também pode ser usada sobre outros tipos de organismos. Apresentamos aqui um estudo de caso usando LNJ sobre um conjunto de oito espécies de bactérias (com seus números de acesso e nomes abreviados):

- *Azotobacter vinelandii* CA (GCF_000380335, Azoto),
- *Pseudomonas syringae* pv. *cerasicola* (GCF_900235885, Pseudo),
- *Escherichia coli* str. K-12 (GCF_000005845, Ecoli),
- *Xylella fastidiosa* str. DSM 10026 (GCF_900129695, Xylella),
- *Xanthomonas fuscans* subsp. *fuscans* 4834-R (GCF_000969685, Xanthofuscans),
- *Xanthomonas axonopodis* pv. *citri* str. 306 (GCF_000007165, Xantho306),
- *Mycobacterium tuberculosis* H37Rv (GCF_000195955, Mtuberc) e
- *Mycobacterium bovis* AF2122/97 (GCF_000195835, Mbovis).

Estes organismos formam quatro clados distintos, de acordo com seus hospedeiros e respectivas doenças que causam.

Na Figura 4.7 é apresentada a topologia obtida por Orthologsorter [83], um *pipeline* automático para comparar genomas em termos de seus genes codificadores de proteínas, usando uma abordagem de supermatriz. Um alinhamento múltiplo de sequências que representa a concatenação de famílias ortólogas de proteínas é utilizado como entrada para RAxML [88], que gera uma árvore filogenética sem raiz, usando por *default* o modelo de substituição PROTCATJTT, com *bootstrap* (100 réplicas) e subsequente pesquisa de verossimilhança máxima.

Tomando o mesmo alinhamento múltiplo de sequências, neste caso contendo 75.738 colunas, usamos PROTDIST [35] para construir uma matriz de distâncias. A partir desta matriz, NJ construiu a mesma topologia que a mostrada na Figura 4.7. A Figura 4.8 mostra a árvore LNJ, que manteve os mesmos clados, mas fazendo Xantho306 um ancestral vivo de Xylella e Xanthofus.

Como outra abordagem usando o mesmo conjunto de dados, mas tomando como entrada a sequência cromossômica de cada organismo, novamente usamos MUMi [25] para construir uma matriz de distâncias de entrada. NJ e LNJ obtiveram a mesma topologia que a mostrada na Figura 4.7, sem nós internos vivos. Isto pode ser explicado pelo fato de que o MUMi é baseado no conteúdo de DNA completo, que inclui grandes porções de transposições (muito comum em bactérias) e pode não capturar similaridades presentes em algumas famílias de proteínas compartilhadas.

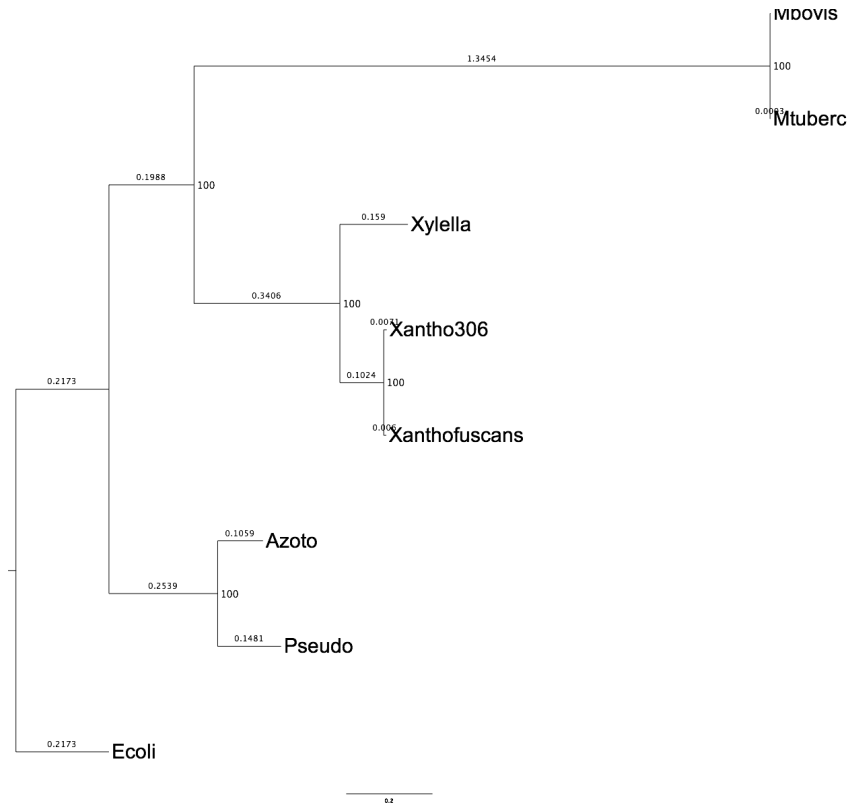


Figura 4.7: Árvore Orthologsorter. Os números nos nós representam os valores de *bootstrap*.

4.4 Testes com dados não biológicos

Na visualização de dados exploratória, uma tarefa importante é a construção de representações visuais que possibilitem aos usuários a busca por grupos de dados relacionados, na descoberta de relações entre itens de dados, na identificação de *outliers* e em outras tarefas [98]. Ferramentas de interação e sumarização são normalmente fornecidas em tais representações visuais.

Uma representação visual generalizada deve ser criada, mapeando cada item de dados em um ponto no espaço visual, de modo que quanto mais relacionados estiverem seus conteúdos, mais próximos estarão seus pontos no leiaute. Este é um problema difícil em geral e foi resolvido na prática usando técnicas de redução dimensional, especialmente projeções multidimensionais [92].

O uso de uma filogenia como técnica de posicionamento de pontos foi analisada por Cuadros e colegas [20]. A Figura 4.9 mostra um exemplo da técnica para um conjunto de imagens de flores. Características interessantes de uma filogenia visual

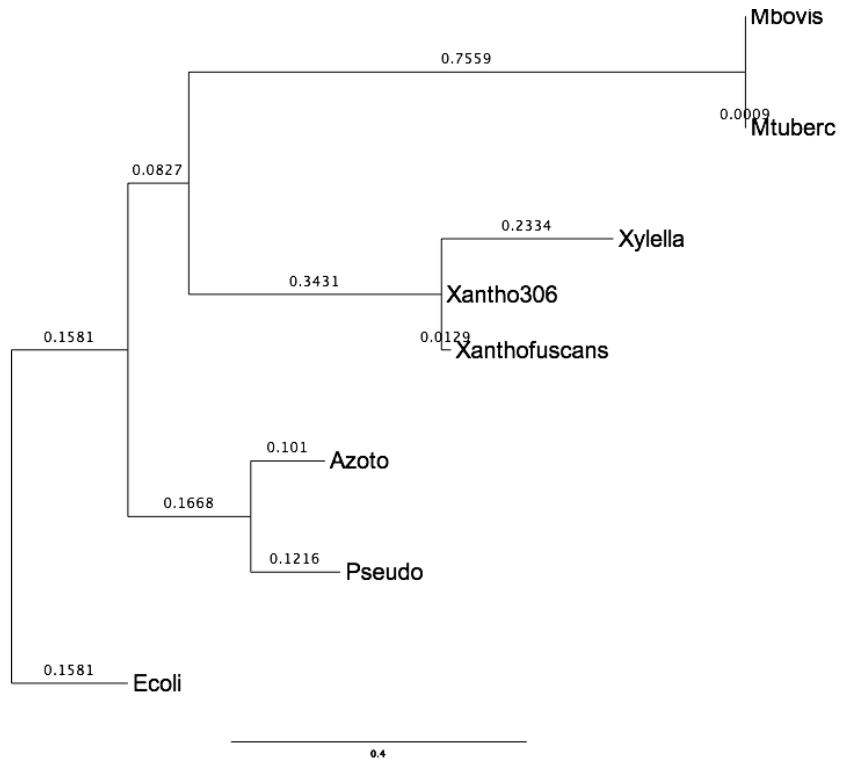


Figura 4.8: Árvore LNJ sobre o conjunto de 8 espécies de bactéria. Xantho306 tornou-se um nó interno vivo.

incluem o fato de que a árvore organiza os dados em ramos de similaridade que são passíveis de exploração e proporcionam uma separação mais clara entre os itens de dados, tanto em níveis pequenos quanto grandes de *zoom*.

Uma desvantagem das filogenias visuais quando comparadas, por exemplo, às projeções, é a ocupação do espaço visual. Uma árvore filogenética para n itens de dados terá $n-2$ nós hipotéticos que representam ancestrais hipotéticos, mas para textos, imagens e outros tipos de dados não biológicos, as noções de evolução e ancestral não são bem definidas a menos que exista uma história de operações de edição e seja conhecida. Isto é uma consequência do fato de que medidas de similaridade entre texto, imagens e outros dados não biológicos não são formuladas para capturar a noção de evolução, como frequentemente fazem as medidas de similaridade para sequências moleculares, mas apenas aspectos de similaridade. Além disso, medir a similaridade entre os itens de dados é um problema difícil por si só.

No entanto, contamos com as medidas de similaridade existentes para a construção de filogenias de dados não-biológicos, porque essas árvores oferecem um bom leiaute para a exploração de dados. LNJ pode ser uma alternativa interessante para a cons-

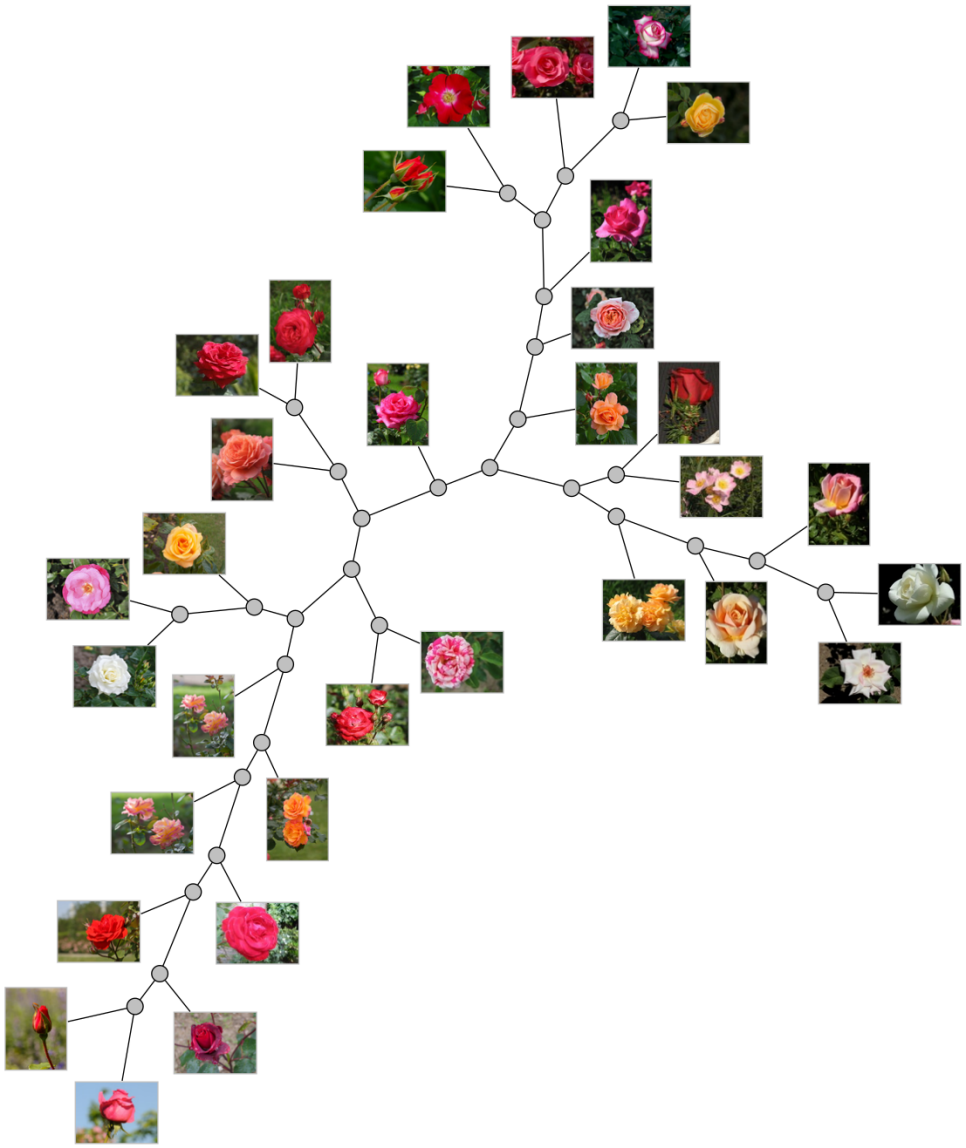


Figura 4.9: Árvore NJ de imagens. Uma filogenia visual por NJ para 32 imagens de Wikipedia Commons (`https://commons.wikimedia.org`).

trução de tais mapas visuais, pois diferentes relações de dados podem ser reveladas e também porque um leiaute mais compacto pode ser útil, já que o número de nós hipotéticos é potencialmente menor.

Em relação à ocupação do espaço visual, o LNJ pode ser ajustado para produzir um leiaute mais compacto se adicionarmos um limiar para a comparação na Linha 5 do algoritmo proposto na Seção 4.1, transformando o teste em $S_{ij} < \alpha T_{uvw}$, para um número real $\alpha > 0$. Ter α menor que 1 favorecerá o posicionamento dos dados como nós internos, e valores maiores de α forçarão LNJ a se comportar como NJ. É claro

que a diferença entre as distâncias na árvore e as distâncias na matriz de entrada irá piorar com a redução de α, mas um equilíbrio adequado pode ser alcançado na prática e a introdução de α amplia a aplicabilidade do LNJ.

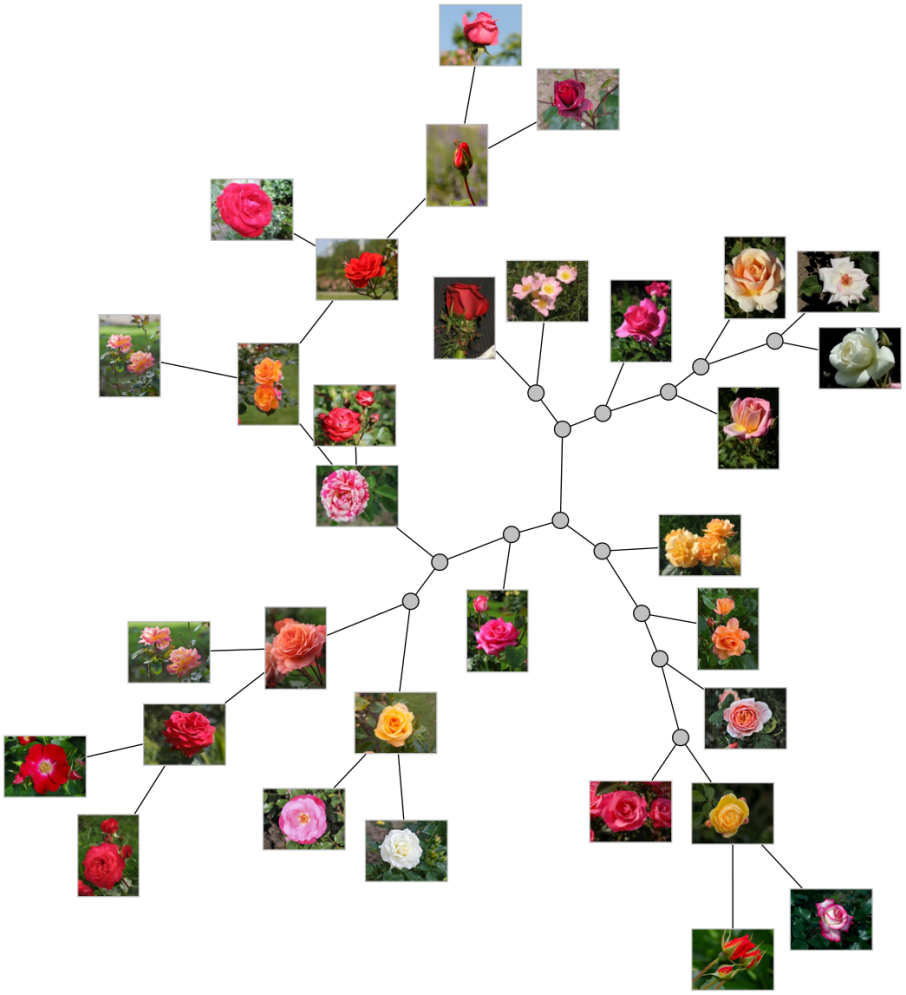


Figura 4.10: Árvore LNJ de imagens. Uma fiogenia visual por LNJ ($\alpha = 0.9$) para 32 imagens de Wikipedia Commons (`https://commons.wikimedia.org`), as mesmas imagens da Fig. 4.9.

A árvore na Figura 4.9 foi construída por Neighbor-Joining sobre distâncias de pares avaliados por semelhança estrutural [97] entre 32 imagens de flores da Wikipedia Commons (`https://commons.wikimedia.org`) cortadas e redimensionadas para 939×704 pixels. A Figura 4.10, por sua vez, mostra a árvore LNJ com $\alpha = 0.9$, que tem menos nós internos e preserva grande parte das relações locais da árvore NJ.

Para ilustrar ainda mais a questão do uso do espaço, a Figura 4.11(a) mostra uma filogenia por NJ para 256 livros gratuitos do projeto de Gutenberg (`http://www.`

`gutenberg.org`) com 510 nós, e a Figura 4.11(b) mostra uma filogenia do LNJ para os mesmos dados com 300 nós e 105 ancestrais vivos.

Os livros em formato ASCII foram pré-processados, visando a remoção do preâmbulo e da licença do Projeto Gutenberg. Em seguida a Distância de Compressão Normalizada [57] para cada par de livros foi avaliada usando o bzip2. Os nós nas árvores foram posicionados por um algoritmo baseado em posicionamento direcionado à força (*Force-Directed Placement*(FDP)). Originalmente proposto como uma heurística de desenho gráfico, o modelo FDP visa trazer um sistema composto por instâncias conectadas por fontes imaginárias em um estado de equilíbrio [20]. As instâncias são inicialmente colocadas aleatoriamente e as forças da mola (fonte imaginária) empurram e puxam iterativamente até atingir o equilíbrio. Em nosso exemplo, os livros representam as instâncias. A mola é representada pela aresta conectando dois nós. Outros exemplos deste modelo podem ser encontrados em [20, 70].

A experiência nos diz que um mapa de dados com mais de mil pontos parece ser muito para explorar de uma vez, e para conjuntos de dados ainda maiores uma abordagem muti-escala combinada com técnicas de sumarização é imperativa. A Tabela 4.1 mostra que LNJ será prático para a construção de leiautes para algumas centenas de itens de dados e também sugere que uma visualização multiescala que particiona o conjunto de dados ainda pode usar LNJ para construir leiautes em níveis mais refinados de um esquema de visualização.

4.5 Comentários

Apesar de LNJ ser caro, podemos apontar como vantagem que, para algumas centenas de objetos, o tempo de execução é pequeno, o que se ajusta a muitos conjuntos de dados biológicos e não biológicos. Além disso, extensões com execuções mais rápidas que já foram propostas para o Neighbor-Joining podem ser aplicadas ao Live Neighbor-Joining, com vantagens e desvantagens que devem ser abordadas em pesquisas futuras.

Com o objetivo de melhor caracterizar o problema da filogenia viva, vamos descrever no próximo capítulo uma outra abordagem, supondo agora que a topologia da árvore também é conhecida e faz parte da entrada do problema, além obviamente da matriz de distâncias. Supostamente esse problema é mais fácil de resolver e poderá, eventualmente, ser útil no desenvolvimento de trabalhos futuros.

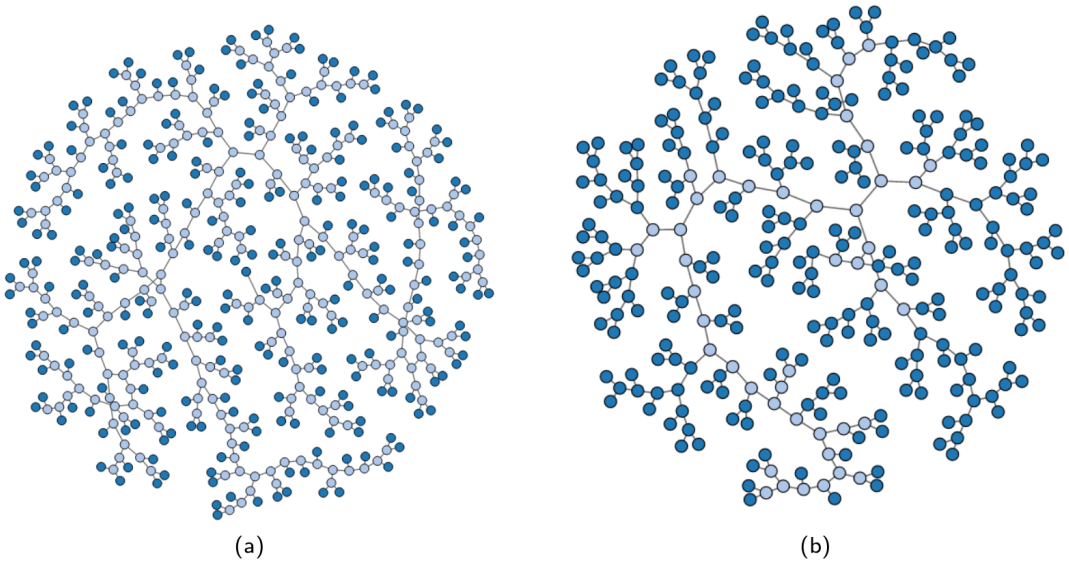


Figura 4.11: Filogenias visuais para 256 livros. (a) Filogenia NJ com 510 nós e 254 ancestrais hipotéticos. (b) Filogenia LNJ com 300 nós e 105 ancestrais vivos.

Capítulo 5

Filogenia com topologia conhecida

Há duas classes principais de técnicas utilizadas para reconstruir filogenias. A primeira compreende os métodos de agrupamento que começam de um pequeno número de objetos e gradualmente adicionam um objeto a cada passo. Eles produzem como saída uma única árvore que tenta recuperar as relações evolutivas entre os objetos [103]. Fazem parte deste primeiro grupo, por exemplo, os métodos UPGMA, Neighbor-Joining e Fitch-Margoliash, apresentados no Capítulo 2.

Já o segundo grupo de métodos gera diferentes topologias e testa cada uma contra os dados na busca por aquela que é ótima ou está mais próxima da ótima, de acordo com algum critério. Esses métodos de construção de árvore geram múltiplas topologias de árvore possíveis, as quais são avaliadas em um procedimento subsequente para obter comprimentos/pesos de arestas e medidas de otimalidade. A geração da topologia é, a princípio, separada da avaliação, mas em muitos casos a medida de otimalidade é calculada como resultado da construção de árvore e é usada como *feedback* durante o processo de construção para excluir algumas topologias [103].

De acordo com Bryant e Waddell [11], é geralmente mais desejável otimizar o ajuste de dados para um modelo de topologia assumido, em vez de simplesmente aplicar um algoritmo para a construção da topologia. Exemplos de critérios de ajuste para árvores incluem mínimos quadrados (*least-squares*), os quais podem ser: mínimos quadrados ordinários ou não ponderados (*Ordinary Least-Squares* – OLS), mínimos quadrados ponderados (*Weighted Least-Squares* – WLS) e mínimos quadrados generalizados (*Generalized Least-Squares* – GLS). Além disso, a literatura

também propõe como modelos de estimativa de comprimentos de arestas os modelos de programação linear, mas esses não serão discutidos neste trabalho e podem ser encontrados em [7, 99].

Vale notar que o método Fitch-Margoliash, apresentado na Seção 2.1.5, pode ser usado para calcular tamanhos de arestas para uma árvore de topologia específica, desde que todos os nós internos tenham três arestas incidentes [103]. Selecionando um nó por vez a partir da topologia específica, as fórmulas da Equação 2.13 (pág. 29) podem ser usadas para calcular os tamanhos de arestas e então reconstruir a árvore. Além disso, o método NJ também pode ser aplicado de maneira equivalente [103].

Neste capítulo, portanto, o problema consiste em, dada uma topologia T e uma matriz de distâncias M entre n objetos, atribuir pesos às arestas de T, de forma a respeitar as distâncias entre os objetos, dadas pela matriz M.

Primeiramente, na Seção 5.1, vamos tratar o caso em que a matriz é aditiva, apresentando um algoritmo polinomial para ambos os casos de filogenia tradicional e filogenia viva. Para matrizes não aditivas, mostraremos, na Seção 5.2, uma modelagem do problema baseada em notação matricial, devida a Cavalli-Sforza e Edwards [16]. Essa notação será a base matemática para os dois algoritmos que resolvem o problema, baseados em OLS, que serão descritos na Seção 5.3. Ao final, na Seção 5.4, alguns comentários acerca do problema e suas especificidades com relação à filogenia viva são apresentados.

5.1 Matriz aditiva

Apresentaremos aqui um algoritmo de tempo linear para atribuir pesos às arestas de uma topologia dada, considerando a matriz de distâncias aditiva, e considerando filogenia tradicional. Em seguida, fazemos o mesmo considerando filogenia viva. Ambos os algoritmos utilizam conceitos de grafos na solução.

Matriz aditiva e filogenia tradicional

No caso da filogenia tradicional, a topologia T possui exatamente n folhas e, dada a matriz de distâncias M, queremos atribuir pesos às arestas de T, de forma que a distância entre dois objetos i, j em T, dada por d_{ij}, seja igual a M_{ij}.

O Algoritmo 5.1 utiliza o conceito de busca em grafos e o conceito de aresta externa e interna. Uma aresta externa é uma aresta que incide em uma folha. Uma aresta interna é uma aresta cujos extremos são nós internos. Além disso, a ideia segue a linha de raciocínio do algoritmo para construir uma filogenia de Waterman e colegas [99], apresentado no Capítulo 2. Apesar de se tratar de uma analogia óbvia, não encontramos na literatura tal descrição.

Inicialmente, como um pré-processamento (Passo I do Algoritmo 5.1), realiza-se uma busca em profundidade na topologia T a fim de se conhecer as folhas alcançáveis por cada aresta incidente em um nó interno c de T, lembrando que em cada nó interno incidem três arestas. Esse pré-processamento fará com que o cálculo do peso de uma aresta (tanto interna quanto externa) seja feito em tempo constante. Em seguida são calculados os pesos das arestas externas (Passo II do Algoritmo 5.1) e, por fim, os pesos das arestas internas (Passo III do Algoritmo 5.1).

A Figura 5.1 ajudará no entendimento do algoritmo aqui descrito, especificamente na parte em que o peso das arestas internas é calculado.

A Figura 5.2 mostra, passo a passo, as três distâncias consideradas para o cálculo do peso b de uma aresta externa, cujas extremidades são uma folha v_i e um nó interno c. Este cálculo é realizado no Passo II do Algoritmo 5.1.

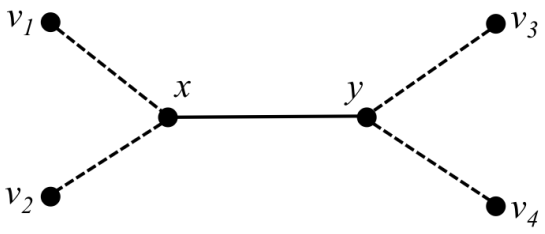


Figura 5.1: Caminhos e nós considerados para o cálculo do peso de uma aresta interna (x, y).

A busca em profundidade é realizada em tempo proporcional ao número de arestas, $2n - 3$. Para cada aresta externa e interna gasta-se tempo constante para realizar o cálculo do peso da aresta, uma vez que já foi feito o pré-processamento e, com isso, já se conhece, para cada nó interno, as folhas alcançáveis por cada uma das arestas que nele incidem. Assim, o tempo de execução do algoritmo para atribuir pesos às arestas é $O(n)$, linear no número de objetos, sem considerar o tempo de leitura da matriz M, que é $O(n^2)$.

Algoritmo 5.1 ROTULAR-ARESTAS-TRADICIONAL-ADITIVA(T,M)

Entrada: (1) Uma topologia tradicional T, (2) Uma matriz de distâncias M entre n objetos.

Saída: Um vetor b com os pesos das arestas de T.

Passo I: Pré-processamento (busca em profundidade na topologia T)

1: Escolha uma folha e faça uma busca em profundidade em T, anotando, para cada nó interno c, uma folha qualquer alcançável por cada uma das 3 arestas incidentes a c.

Passo II: Cálculo dos pesos das arestas externas

2: **for** cada aresta externa e adjacente a um nó interno c **do**

3: Sejam v_i, v_j, v_k as folhas alcançáveis a partir de c

4: Suponha que v_i seja a folha adjacente a c

5: Calcule o peso $b(v_i, c) = \frac{M_{v_i v_j} + M_{v_i v_k} - M_{v_j v_k}}{2}$

6: **end for**

Passo III: Cálculo dos pesos das arestas internas

7: **for** cada aresta $e = (x, y)$ interna **do**

8: Sejam v_1, v_2 as folhas alcançáveis por x e v_3, v_4 por y

9: Suponha que v_i seja a folha adjacente a c

10: Calcule $b(x, y) = M_{v_1 v_3} - \frac{M_{v_1 v_2} + M_{v_1 v_3} - M_{v_2 v_3}}{2} - \frac{M_{v_1 v_3} + M_{v_3 v_4} - M_{v_1 v_4}}{2}$

11: **end for**

12: **return** b

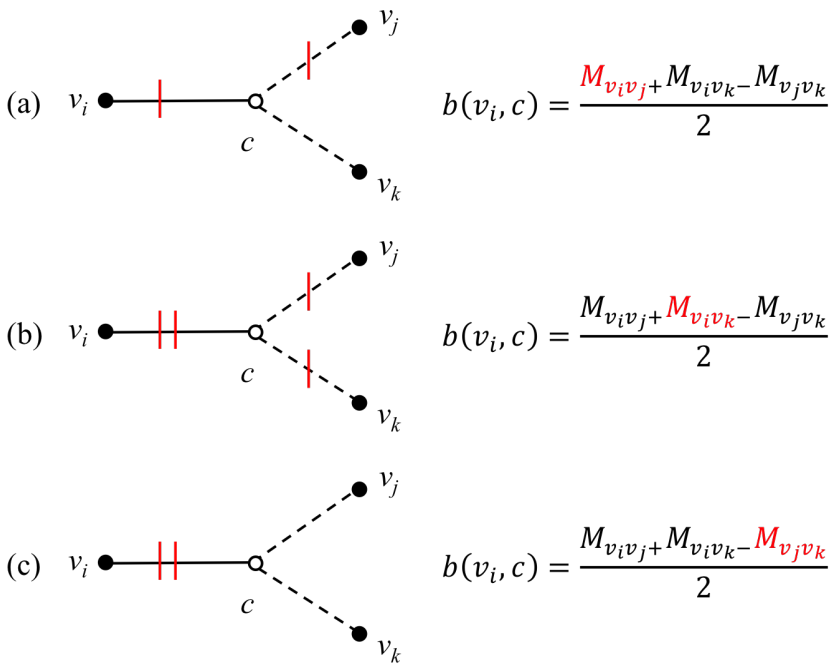


Figura 5.2: (a) $M_{v_i v_j}$ sendo adicionada. (b) $M_{v_i v_k}$ sendo adicionada e (c) $M_{v_j v_k}$ sendo subtraída

A implementação do Algoritmo 5.1 está disponível em `https://git.facom.ufms.br/bioinfo/topology/tree/master/traditional/`.

Matriz aditiva e filogenia viva

No caso de uma filogenia viva, a topologia T possui no máximo n folhas, e dada a matriz de distâncias M, de ordem n, queremos atribuir pesos às arestas de T, de forma que a distância (d_{ij}) entre dois objetos i e j em T seja igual a M_{ij}.

O algoritmo para o filogenia viva é muito similar ao da tradicional que acabamos de descrever. Seu pseudocódigo é apresentado no Algoritmo 5.2. Inicialmente, realiza-se uma busca em profundidade na topologia T a fim de se conhecer as folhas alcançáveis por cada aresta incidente em um nó interno de T (Passo I do Algoritmo 5.2). Em seguida são calculados os pesos das arestas externas (Passo II do Algoritmo 5.2). É importante lembrar que, neste caso, uma aresta externa que incide em uma folha pode ter como outra extremidade um nó interno vivo. Logo, o peso da aresta externa será exatamente igual à distância dada na matriz de entrada. Por fim, calculam-se os pesos das arestas internas (Passo III do Algoritmo 5.2), em que também podemos ter o caso em que aresta interna incide em dois nós internos vivos e, com isso, o peso da aresta será igual à distância dada na matriz entre aquele par de objetos vivos. O Algoritmo 5.2 foi desenvolvido neste trabalho e resolve o problema em tempo linear, sem considerar o tempo de leitura da matriz M, que é $O(n^2)$.

O algoritmo para filogenia viva é bastante similar ao da filogenia tradicional, como já mencionado. Também possui um passo de pré-processamento, que, igualmente, gasta tempo proporcional ao número de arestas, $2n - 3$, seguido dos cálculos dos pesos das arestas externas e internas realizados em tempo constante. Portanto, o tempo de execução do algoritmo para atribuir pesos às arestas é $O(n)$, linear no número de objetos, sem considerar o tempo de leitura da matriz M, que é $O(n^2)$.

Experimentos

Ambos os algoritmos apresentados nesta seção para matriz aditiva, considerando filogenia tradicional e viva, foram implementados e testados com matrizes de tamanho n, sendo $10 \leq n \leq 100$, em intervalos de tamanho 10. Para cada valor de n, um subconjunto de 100 matrizes foi gerado.

Algoritmo 5.2 ROTULAR-ARESTAS-VIVA-ADITIVA(T,M)

Entrada: (1) Uma topologia viva T, (2) Uma matriz de distâncias M entre n objetos.

Saída: Um vetor b com os pesos das arestas de T.

Passo I: Pré-processamento (busca em profundidade na topologia T)

1: Escolha uma folha e faça uma busca em profundidade em T, anotando, para cada nó interno c, uma folha qualquer alcançável por cada uma das 3 arestas incidentes a c.

Passo II: Cálculo dos pesos das arestas externas

2: **for** cada aresta externa e adjacente a um nó interno c **do**
3: Seja v_i uma folha adjacente a c
4: **if** c é vivo **then**
5: $b(v_i, c) = M_{v_i c}$
6: **else**
7: Sejam v_j, v_k as outras folhas alcançáveis a partir de c
8: Calcule o peso $b(v_i, c) = \frac{M_{v_i v_j} + M_{v_i v_k} - M_{v_j v_k}}{2}$
9: **end if**
10: **end for**

Passo III: Cálculo dos pesos das arestas internas

11: **for** cada aresta $e = (x, y)$ interna **do**
12: **if** x e y são vivos **then**
13: $b(x, y) = M_{xy}$
14: **else**
15: **if** x é vivo **then**
16: Sejam v_j, v_k as folhas alcançáveis a partir de y
17: Calcule o peso $b(x, y) = \frac{M_{x v_j} + M_{x v_k} - M_{v_j v_k}}{2}$
18: **else if** y é vivo **then**
19: Sejam v_j, v_k as folhas alcançáveis a partir de x
20: Calcule o peso $b(x, y) = \frac{M_{y v_j} + M_{y v_k} - M_{v_j v_k}}{2}$
21: **else**
22: Sejam v_1, v_2 folhas alcançáveis por x e v_3, v_4 por y
23: Calcule $b(x, y) = M_{v_1 v_3} - \frac{M_{v_1 v_2} + M_{v_1 v_3} - M_{v_2 v_3}}{2} - \frac{M_{v_1 v_3} + M_{v_3 v_4} - M_{v_1 v_4}}{2}$
24: **end if**
25: **end if**
26: **end for**
27: **return** b

Cada matriz de entrada é gerada através da criação de uma árvore randômica, e em seguida, uma matriz é gerada dessa árvore. Logo, por construção, tal matriz é aditiva. A topologia da árvore e a matriz de distâncias são armazenadas em arquivos separados. O procedimento geral de geração de topologia e matriz de distâncias é o mesmo tanto para o caso da filogenia tradicional como para a viva. No caso da filogenia viva, geramos as árvores com nós internos vivos, sendo que as mesmas foram geradas contendo de 1 até $n - 2$ nós internos vivos.

Ambos os algoritmos obtiveram 100% de êxito, ou seja, os pesos das arestas foram atribuídos respeitando as distâncias dadas na matriz de entrada.

A implementação do Algoritmo 5.2 para atribuir pesos em filogenia viva está disponível em `https://git.facom.ufms.br/bioinfo/topology/tree/master/live/`.

Os algoritmos apresentados para a tarefa de atribuir pesos às arestas de uma topologia, tanto para filogenia tradicional como para filogenia viva, são lineares. Porém, nos dois casos, para obtermos a distância entre os objetos, realiza-se uma busca em profundidade na topologia para cada um dos objetos, com custo $O(n^2)$.

Na próxima seção apresentaremos uma abordagem conhecida como mínimos quadrados, para resolver o problema de atribuir pesos às arestas de uma topologia, considerando uma matriz não aditiva.

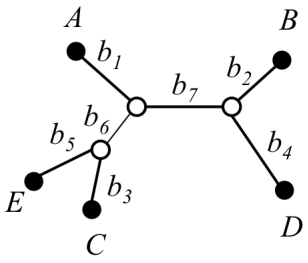
5.2 Método dos mínimos quadrados

A idéia fundamental dos métodos de matriz de distâncias é que temos uma matriz de distâncias (M_{ij}) entre objetos e que qualquer árvore em particular com pesos nas arestas leva a um conjunto predito de distâncias (aqui denotado por d_{ij}). A previsão da distância entre duas espécies é dada pela soma dos pesos das arestas no caminho entre as duas espécies. A Figura 5.3 mostra uma matriz de distâncias observadas e uma topologia. O método dos mínimos quadrados pode ser utilizado para resolver o problema de atribuir pesos às arestas da topologia.

Os modelos de mínimos quadrados foram introduzidos pela primeira vez por Cavalli-Sforza e Edwards [16], que consideraram cada distância evolutiva entre os pares de objetos como variáveis aleatórias independentes uniformemente distribuídas, satisfazendo a propriedade aditiva (Lema 2.2, pág. 13). Este método não estima as

	A	B	C	D	E
A	0	23	16	20	17
B	23	0	23	17	24
C	16	23	0	15	11
D	20	17	15	0	21
E	17	24	11	21	0

(a)



(b)

Figura 5.3: (a) Uma matriz de distâncias entre 5 objetos. (b) Uma topologia com os pesos das arestas como variáveis.

posições dos nós no tempo ou no espaço, mas tem a vantagem de poder ser aplicado a distâncias que não podem ser representadas em um espaço métrico [16].

Em outras palavras, Cavalli-Sforza e Edwards assumiram que cada entrada M_{ij} poderia ser considerada como a soma resultante dos eventos de mutação b_e acumulados em cada aresta e e pertencentes ao caminho p_{ij} que liga os táxons i e j na árvore.

Suponha que uma topologia seja representada na forma de uma matriz X, em que cada linha representa um par de objetos e cada coluna uma aresta da topologia. Então, o valor de cada célula de X é dado por:

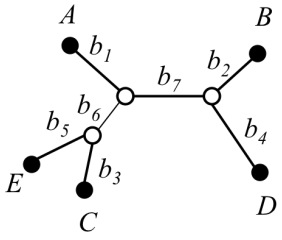
$$X_{ij} = \begin{cases} 1, & \text{se a aresta da coluna } j \text{ está no caminho do par de objetos na linha } i \\ 0, & \text{caso contrário.} \end{cases}$$

A matriz X para a topologia da Figura 5.3 é mostrada na Figura 5.4, assim como as distâncias M_{ij} colocadas em um vetor D, dispostas conforme ordenação crescente dos nomes dos pares de objetos. A matriz X, às vezes, será referenciada no texto como matriz topológica.

Desta forma, o relacionamento entre a topologia e o vetor de distâncias D pode ser descrito no formato geral de matriz como:

$$Xb = D \tag{5.1}$$

onde D é o vetor com $n(n-1)/2$ componentes, mostrado acima, obtidos por tomar linha por linha as entradas da matriz triangular estritamente superior de M, b é o vetor com os pesos das arestas da topologia. Em geral, a Equação 5.1 pode



$$X = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \quad D = \begin{bmatrix} D_{AB} \\ D_{AC} \\ D_{AD} \\ D_{AE} \\ D_{BC} \\ D_{BD} \\ D_{BE} \\ D_{CD} \\ D_{CE} \\ D_{DE} \end{bmatrix}$$

Figura 5.4: Uma topologia, a matriz X correspondente à topologia, o vetor b com os pesos das arestas da topologia e o vetor D contendo as distâncias entre os objetos.

não admitir soluções. Por essa razão, os autores propuseram o uso dos mínimos quadrados ordinários (*Ordinary Least-Squares* - OLS) para encontrar as m entradas do vetor b [15] (m é o número de arestas da topologia). Especificamente, os autores sugeriram que os valores $d_{ij} = \sum_{e \in p_{ij}} X_{ij,e} b_e$, chamados de *estimativas de distância esperada*, deveriam minimizar a função a seguir, que é uma medida de discrepância entre as distâncias observadas e as esperadas [37]:

$$Q = \sum_{i=1}^{n} \sum_{j=1:j\neq i}^{n} (M_{ij} - d_{ij})^2 = \sum_{i=1}^{n} \sum_{j=1:j\neq i}^{n} (M_{ij} - \sum_{e \in p_{ij}} X_{ij,e} b_e)^2. \tag{5.2}$$

Vamos descrever a seguir como obter os valores do vetor b, contendo os pesos das m arestas de uma topologia.

Comprimentos de arestas por mínimos quadrados

Para encontrar os comprimentos de arestas em uma determinada topologia usando mínimos quadrados, devemos minimizar Q. A expressão para Q na Equação 5.2 é quadrática nos comprimentos das arestas. Uma maneira de minimizá-la é resolver um conjunto de equações lineares. Estas são obtidas tomando derivadas de Q em relação aos comprimentos das arestas e igualando-as a zero. A solução das equações resultantes minimizará Q.

Se diferenciarmos Q em relação a um dos b's, por exemplo, b_k, e igualarmos a

derivada à zero, obtemos a equação

$$\frac{dQ}{db_k} = -2\sum_{i=1}^{n}\sum_{j:j\neq i}((M_{ij} - \sum_{k} X_{ij,k}b_k) = 0. \tag{5.3}$$

O valor -2 pode ser descartado.

Uma maneira de se fazer uma estimativa de comprimentos de arestas por mínimos quadrados é resolver este conjunto de equações lineares. Existem métodos exatos e iterativos para fazer isso. No caso dos métodos de mínimos quadrados não ponderados originais de Cavalli-Sforza e Edwards, as equações são particularmente simples. Isso nos levará a uma boa forma matricial, e o caso mais geral pode ser colocado dessa forma. Para o caso não ponderado, para a topologia da Figura 5.3, as equações são descritas como abaixo, uma equação para cada aresta:

$$\begin{aligned}
AB + AC + AD + AE &= 4b_1 + b_2 + b_3 + b_4 + b_5 + 2b_6 + 2b_7 \\
BA + BC + BD + BE &= b_1 + 4b_2 + b_3 + b_4 + b_5 + 2b_6 + 3b_7 \\
CA + CB + CD + CE &= b_1 + b_2 + 4b_3 + b_4 + b_5 + 3b_6 + 2b_7 \\
DA + DB + DC + DE &= b_1 + b_2 + b_3 + 4b_4 + b_5 + 2b_6 + 3b_7 \\
EA + EB + EC + ED &= b_1 + b_2 + b_3 + b_4 + 4b_5 + 3b_6 + 2b_7 \\
EA + EB + ED + CA + CB + CD &= 2b_1 + 2b_2 + 3b_3 + 2b_4 + 3b_5 + 6b_6 + 4b_7 \\
AB + AD + EB + ED + CB + CD &= 2b_1 + 3b_2 + 2b_3 + 3b_4 + 2b_5 + 4b_6 + 6b_7
\end{aligned} \tag{5.4}$$

Uma vez representada a topologia pela matrix X, as distâncias entre pares de objetos pelo vetor D, as equações podem ser expressas de maneira compacta através de notação matricial, a partir da equação 5.1, como:

$$X^t D = (X^t X)b,$$

onde X^t é a transposta de X. Multiplicando o lado esquerdo pela inversa de $(X^t X)$, os comprimentos de arestas por mínimos quadrados podem ser resolvidos:

$$b = (X^t X)^{-1} X^t D \tag{5.5}$$

Este é um método padrão de expressar problemas de mínimos quadrados em notação matricial e resolvê-los.

Alguns autores discordaram do modelo de Cavalli-Sforza e Edwards. Especificamente, Fitch e Margoliash [39] *apud* Catanzaro [15], observaram que, devido à história evolutiva comum dos táxons analisados e à presença de erros de amostragem em dados moleculares, a suposição de que as distâncias evolutivas M_{ij} são variáveis aleatórias independentes distribuídas uniformemente geralmente não pode ser considerada verdadeira. É bem conhecido que as estimativas obtidas a partir de sequências não tem a mesma variância, pois as distâncias maiores são muito mais variáveis que as menores e são mutuamente dependentes quando compartilham uma história (ou caminho) em comum na filogenia verdadeira [26].

Dessa forma, os autores propuseram modificar o modelo de Cavalli-Sforza e Edwards, introduzindo as grandezas ω_{ij} representando as variâncias de M_{ij}. Fitch e Margoliash chamaram o novo modelo de mínimos quadrados ponderados (*Weighted Least-Squares* - WLS) e propuseram minimizar a função:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\omega_{ij}(M_{ij}-\sum_{e\in p_{ij}}X_{ij,e}b_e)^2. \tag{5.6}$$

Fitch e Margoliash propuseram definir $\omega_{ij} = 1/M_{ij}^2$ ([39] *apud* [15]), enquanto que, com argumentos análogos, Beyer e colegas [7] *apud* Catanzaro [15], definiram $\omega_{ij} = 1/M_{ij}$. No modelo WLS, a função a ser minimizada torna-se:

$$(X^t\Omega X)b = X^t\Omega D \tag{5.7}$$

em que Ω é uma matriz diagonal das variâncias das estimativas de M_{ij}.

Subsequentemente, Bulmer [12] *apud* Catanzaro [15], Chakraborty [18] *apud* Catanzaro [15] e Hasegawa e colegas [50] *apud* Catanzaro [15], observaram que os pesos ω_{ij} são responsáveis pela variação de M_{ij}, mas não por suas dependências. Consequentemente, eles propuseram substituir os valores ω_{ij} pelas covariâncias de M_{ij} e chamaram o novo modelo de mínimos quadrados generalizados (*Generalized Least-Squares* - GLS). Especificamente, Chakraborty [18] *apud* Catanzaro [15], modelou a evolução molecular como um processo de Poisson no qual mutações são eventos aleatórios ocorrendo ao longo de cada caminho na filogenia, e derivou as covariâncias das distâncias evolutivas considerando, em cada caminho, o número de eventos de mutação observados entre pares de dados moleculares. Uma abordagem muito semelhante foi usada por Bulmer [12] e Hasegawa e colegas [50] *apud* Catanzaro [15]:

o primeiro usou uma aproximação do processo de Poisson para calcular as covariâncias das distâncias evolutivas, enquanto o segundo usou um modelo de Markov. Considerando o modelo GLS, a função a ser minizada torna-se:

$$(X^t \psi^{-1} X)b = X^t \psi^{-1} D \tag{5.8}$$

em que ψ é a matriz de covariância das distâncias evolutivas.

A complexidade computacional necessária para resolver, por meio de fórmulas matriciais, os modelos acima ($O(n^4)$ para os modelos OLS e WLS e $O(n^6)$ para o modelo GLS [11]) representou nas décadas de 1970 e 1980 um sério gargalo para sua aplicação empírica [15]. Por esse motivo, vários autores investigaram estratégias alternativas para reduzir o esforço computacional necessário para implementá-las.

Vach [95] *apud* Catanzaro [15] observou que a bipartição (também chamada *split* [5] *apud* [15]) induzida por qualquer aresta de uma filogenia pode ser usada para aproximar o modelo OLS. Especificamente, dada uma topologia X e assumindo o modelo de estimativa de peso de aresta OLS, Vach provou que: i) o valor de um peso de aresta é uma função da distância média entre as folhas pertencentes a uma bipartição induzida pela aresta e; ii) tal valor não depende da filogenia, mas apenas das folhas contidas na bipartição [15].

Este resultado foi alcançado independentemente por Rzhetsky e Nei [79] que forneceram um algoritmo $O(n^3)$ para resolver o modelo OLS [80]. Este algoritmo foi melhorado por Gascuel [41], que diminuiu sua ordem de complexidade para $O(n^2)$. Finalmente, Bryant e Waddell [11] propuseram uma estrutura unificada e generalizada para acelerar a solução dos modelos OLS, WLS e GLS. Especificamente, os autores forneceram um algoritmo ideal para resolver o modelo OLS, que será apresentado na Seção 5.3, um algoritmo $O(n^3)$ para resolver o modelo WLS e um algoritmo $O(n^4)$ para resolver o modelo GLS.

Finalmente, Makarenkov e Lapointe [61] *apud* Catanzaro [15] introduziram um modelo WLS específico que pode ser utilizado em todos os casos em que algumas distâncias evolutivas são parcialmente dadas ou incertas (casos geralmente encontrados, por exemplo, quando se trabalha com dados fósseis [61] *apud* [15]). O modelo assume que as propriedades de aditividade são válidas para a matriz de distâncias M e atribui $\omega_{ij} \in \{0, 1/2, 1\}$ como uma função do grau de incerteza das entradas M_{ij}.

Tabela 5.1: Complexidade das soluções para os modelos OLS e WLS.

OLS	WLS
Aplicação direta da fórmula – $O(n^4)$	Aplicação direta da fórmula – $O(n^4)$
Rzhetsky & Nei (1994) – $O(n^3)$	Felsenstein (1997) – $O(n^3)$
Bryant & Waddel (1998) – $O(n^2)$	Bryant & Waddel (1998) – $O(n^3)$

Os autores provaram que resolver esta versão particular do Problema da Evolução Mínima é NP-difícil.

Na Tabela 5.1 são citadas algumas das complexidades das soluções para o modelo OLS e WLS, segundo Catanzaro [15].

Descrevemos nesta seção os principais conceitos relacionados aos métodos dos mínimos quadrados. Dentre os três modelos apresentados OLS, WLS e GLS, vamos nos concentrar no primeiro, pela complexidade mais baixa em relação aos outros.

A seguir apresentaremos dois algoritmos para resolver o problema no modelo OLS e que também podem ser utilizados no caso de uma matriz não aditiva.

5.3 Matriz não aditiva

A maioria dos algoritmos e trabalhos já mencionados até aqui buscou resolver o problema de atribuir pesos às arestas de uma topologia dada, baseando-se num modelo de árvore aditiva, sendo que alguns seguiam o modelo dos mínimos quadrados. Quando a matriz de dados de entrada é aditiva, o problema é resolvido, porém, às vezes, com um custo elevado de tempo de execução.

No entanto, é possível aplicar o método em casos não aditivos [16]. Porém, não se garante a positividade nos pesos das arestas. Segundo Gascuel [43], as atribuições de peso de aresta algébrico dados pelos métodos dos mínimos quadrados tem a propriedade indesejável de poder atribuir pesos negativos a várias das arestas de uma topologia. Pesos de arestas negativos não são aprovados por biólogos evolucionistas, uma vez que a evolução não pode retroceder. Além disso, quando se usa uma abordagem de mínimos quadrados, ou seja, quando não apenas os pesos das arestas são selecionados usando um critério de mínimos quadrados, mas também a topologia da árvore, permitindo pesos de arestas negativos, isto possibilita um alto grau de liberdade e pode resultar em árvores sub-ótimas usando pesos de arestas negativos

para produzir um erro aparente baixo [43].

Na seção a seguir apresentaremos a formulação utilizada para estimar os pesos das arestas de uma dada topologia, segundo Rzhetsky e Nei [79], sob o modelo OLS.

A formulação de Rzhetsky e Nei

Usar o método matricial dos mínimos quadrados para um grande número de topologias demanda uma grande quantidade de recursos de computador. No entanto, há métodos mais simples e rápidos. A seguir, apresentaremos uma formulação mais simples e com menor custo computacional, em relação à solução original de Cavalli-Sforza e Edwards [16].

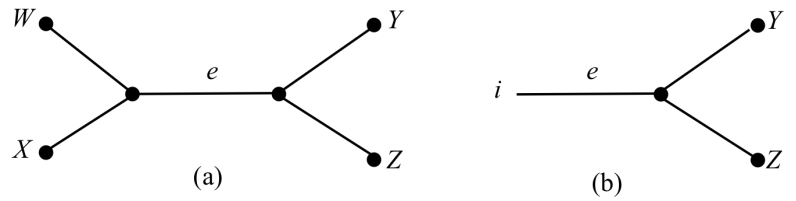


Figura 5.5: (a) Uma topologia evidenciando uma aresta interna. (b) Uma topologia evidenciando uma aresta externa.

Dada uma topologia T, como a mostrada na Figura 5.5(a), em que temos quatro subárvores (cada uma formando um *cluster* de objetos) denominadas W, X, Y e Z, cujos tamanhos são, respectivamente, N_W, N_X, N_Y e N_Z, suponha que queremos atribuir um peso para a aresta interna e da figura, que separa as subárvores W e X das subárvores Y e Z. A estimativa OLS para o peso da aresta interna e, segundo [43, 79], é dada por:

$$b_e = \frac{1}{2}[\lambda(\Delta_{WY} + \Delta_{XZ}) + (1-\lambda)(\Delta_{WZ} + \Delta_{XY}) - (\Delta_{WX} + \Delta_{YZ})] \tag{5.9}$$

com

$$\lambda = \frac{N_W N_Z + N_X N_Y}{N_{W\cup X} N_{Y\cup Z}}, \tag{5.10}$$

$$\Delta_{YZ} = \frac{\sum_{i\in Y, j\in Z} D_{ij}}{N_Y N_Z}, \tag{5.11}$$

Δ_{YZ} é definido como a média aritmética de todas as distâncias entre os *clusters* Y e Z.

Da mesma forma, a estimativa OLS para uma aresta externa e [43, 79] (Figura 5.5(b)) é dada por:

$$b_e = \frac{1}{2}(\Delta_{\{i\}Y} + \Delta_{\{i\}Z} - \Delta_{YZ}) \tag{5.12}$$

Ao aplicarmos as fórmulas acima sobre uma topologia dada, calculando o peso das m arestas, o procedimento como um todo acarretará um consumo de tempo $O(n^3)$. No entanto, outros dois algoritmos foram propostos para resolver o problema OLS, com melhor tempo de execução, sendo um deles por Gascuel [41] e o outro por Bryant e Wadell [11]. Ambos resolvem o problema em tempo $O(n^2)$. O último será apresentado a seguir.

Algoritmo de Bryant e Wadell

O algoritmo apresentado por Bryant e Wadell [11] para resolver o problema de atribuir pesos às arestas de uma topologia dada, usando o modelo dos mínimos quadrados não ponderado, é um dos mais eficientes encontrados na literatura [15]. Vamos descrever ao longo desta seção conceitos e algoritmos utilizados para encontrar os valores do vetor b, cuja fórmula foi apresentada na Equação 5.5 (pág. 80).

Um *split* $A|B$ é uma partição do conjunto de objetos (táxons) O em duas partes, A e B [11]. Cada aresta e de uma árvore corresponde a um único *split*, pois, ao removê-la, a árvore e, consequentemente o conjunto de objetos da árvore, é dividida em duas partes.

Qualquer *split* dado tem um *split metric* associado, ou seja, uma distância no conjunto de objetos, em que dois objetos estão separados por uma distância 1 se estiverem em lados diferentes da divisão e por uma distância 0 se estiverem do mesmo lado da divisão. As colunas da matriz topológica X de uma árvore são exatamente os *split metrics* associados aos *splits* na árvore. A coluna k da matriz X é o *split metric* para o *split* correspondente à aresta e_k.

Um dos truques apresentado por Bryant and Wadell é um método para multiplicar um vetor pela transposta da matriz topológica de uma árvore em uma fração do tempo tomado pela multiplicação padrão de matrizes [11].

Como pode ser observado, as formulações para os pesos das arestas no modelo OLS, WLS e GLS (Equações 5.5, 5.7, 5.8), utilizam a multiplicação de um vetor (ou matriz) por X^t, a transposta da matriz topológica.

As colunas de X são os *split metrics* δ_1, δ_2, ..., δ_m correspondentes às arestas da topologia, então os elementos de X^tD são as quantidades δ_1^tD, δ_2^tD, ..., δ_m^tD.

O cálculo de δ_i^tD para todos os *split metrics* δ_i que correspondem a arestas externas de T será mostrado na equação a seguir. Seja e_i uma aresta externa, adjacente por exemplo, a um objeto x, então

$$\delta_i^tD = \sum_{y \in O-\{x\}} D_{xy}. \tag{5.13}$$

Considere as árvores binárias da Figura 5.6 para o entendimento dos cálculos de δ^tD. O aumento na velocidade do algoritmo de Bryant e Wadell depende do relacionamento entre δ_i^tD e δ_j^tD para arestas adjacentes e_i e e_j [11]. O cálculo do valor de δ^tD para uma aresta interna é apresentado no Teorema 5.1.

Teorema 5.1. *Seja e_i uma aresta interna de uma árvore binária e sejam e_j, e_k as arestas adjacentes na mesma extremidade de e_i. Sejam C_j, C_k e C_i os clusters correspondentes, conforme visto na Figura 5.6(a) (note que $C_i = C_l \cup C_m$) [11]. Então,*

$$\delta_i^tD = \delta_j^tD + \delta_k^tD - 2 \sum_{x \in C_j, y \in C_k} D_{xy}. \tag{5.14}$$

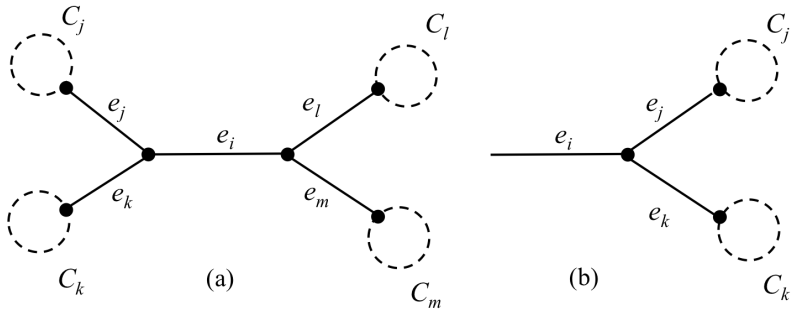


Figura 5.6: Arestas e *clusters* considerados para o cálculo do peso de aresta interna (a) e externa (b) no algoritmo de Bryant e Wadell[1] [11].

[1]Figura adaptada de [11]

A demonstração do Teorema 5.1 pode ser encontrada no trabalho de Bryant e Waddell [11](pág. 1349).

No Algoritmo 5.3, apresentamos o pseudocódigo de FastMTM, uma abreviação em inglês para multiplicação rápida por matriz topológica, para calcular $\delta^t D$ para todas as arestas.

Algoritmo 5.3 FastMTM(T,D)

Entrada: (1) Uma topologia T, (2) Um vetor D contendo as distâncias entre os pares de objetos.

Saída: O valor de $\delta^t D$ para cada aresta e_i.

Passo I: Cálculo de $\delta^t D$ para as arestas externas

1: **for** cada aresta externa e_i **do**

2: $\quad \delta_i^t D \leftarrow \sum_{y \in O - \{x\}} D_{xy}$, em que x é o objeto adjacente à aresta e_i

3: **end for**

Passo II: Cálculo de $\delta^t D$ para as arestas internas

4: Escolha uma aresta interna arbitrária e_j, insira-a em uma lista de arestas L. Adicione as arestas adjacentes a e_j ao final da lista L, e depois as adjacentes a estas e, assim por diante, até que todas as arestas internas estejam na lista.

5: Percorra a lista L no sentido fim para o começo, calculando $\delta_i^t D$ para cada aresta usando a Equação 5.14. Neste ponto, note que todas as arestas adjacentes a uma das extremidades já terá seu $\delta^t D$ calculado.

6: **retorne** $\delta^t D$

O algoritmo FastMTM realiza no máximo $3n^2 - n/2$ operações. A quantidade de memória utilizada, em adição à memória necessária para armazenar o vetor D, é $O(n)$, em que n é o número de objetos [11].

Um algoritmo rápido para o cálculo de b no modelo OLS

Vamos analisar a fórmula de projeção para os pesos das arestas no modelo OLS (Equação 5.5). O maior obstáculo para se obter um algoritmo de tempo $O(n^2)$ para o cálculo de b é a construção e inversão da matriz $(X^t X)^{-1}$. Bryant e Wadell observaram que esta matriz é formada principalmente de zeros. A razão é que o peso atribuído a uma aresta e_i no modelo OLS não é afetado pela forma de uma árvore além daquelas arestas diretamente adjacentes a e_i. Consequentemente, o peso de uma aresta e_i pode ser escrito em termos de $\delta_i^t D$, $\{\delta_{jl}^t D : e_{jl}$ adjacente a $e_i\}$ e as quantidades de objetos nas subárvores correspondentes [11]. Tal observação foi feita em terminologia ligeiramente diferente por Vach [11, 95] e, mais tarde, de forma independente, por Bryant [9].

Dois casos são considerados para a formulação dos pesos das arestas em OLS: arestas internas e externas. Seja e_i uma aresta interna qualquer em uma árvore binária T, conforme mostrado na Figura 5.6(a). As arestas adjacentes a e_i são denotadas por e_j, e_k, e_l e e_m, e as subárvores de T adjacentes a estas arestas são representadas por círculos tracejados. Sejam N_j, N_k, N_l e N_m as quantidades de objetos nas subárvores adjacentes a e_j, e_k, e_l e e_m, respectivamente. O peso ótimo b_i da aresta interna e_i sob OLS é dado por [11]:

$$\begin{aligned} b_i &= [(\frac{N}{N_m}+\frac{N}{N_l}+\frac{N}{N_k}+\frac{N}{N_j}-4)\delta_i^t D \\ &\quad +\frac{N_j+N_k}{N_jN_k}((2N_k-N)\delta_j^t D+(2N_j-N)\delta_k^t D) \\ &\quad +\frac{N_l+N_m}{N_lN_m}((2N_m-N)\delta_l^t D+(2N_l-N)\delta_m^t D)] \\ &\quad \times\frac{1}{4(N_j+N_k)(N_l+N_m)}. \end{aligned} \tag{5.15}$$

A fórmula é derivada por construção e resolvendo um conjunto apropriado de equações de matriz ([9], pág. 136).

A fórmula para arestas externas em árvores binárias é mais simples. Seja e_i uma aresta externa qualquer de uma árvore binária T, conforme mostrado na Figura 5.6(b). Sejam e_j e e_k as arestas adjacentes, e sejam N_j e N_k as quantidades de objetos nas subárvores adjacentes a estas arestas. O peso ótimo b_i para a aresta externa e_i sob OLS é dado por

$$\begin{aligned} b_i = \frac{1}{4N_jN_k} \quad & [(1+N_j+N_k)\delta_i^t D-(1+N_j-N_k)\delta_j^t D \\ & -(1-N_j+N_k)\delta_k^t D]. \end{aligned} \tag{5.16}$$

O Algoritmo 5.4 (BinaryEdges) é o algoritmo final para calcular os pesos das arestas de uma topologia T.

Ele realiza no máximo $\frac{3}{2}n^2+\frac{127}{2}n-126$ operações. A quantidade de memória utilizada, em adição à memória necessária para armazenar o vetor D, é $O(n)$, em que n é o número de objetos [11].

Como forma de validar o algoritmo, realizamos testes com ele, considerando matrizes aditivas com n objetos, sendo $10 \leq n \leq 100$, em intervalos de tamanho 10. Em todos os testes o algoritmo obteve 100% de êxito, atribuindo pesos de forma a respeitar as

Algoritmo 5.4 BinaryEdges(T, D)

Entrada: (1) Uma topologia T, (2) Um vetor D contendo as distâncias entre os pares de objetos.
Saída: Um vetor b com os pesos das arestas de T.
Passo I: Calcule $\delta_i^t D$ para cada aresta e_i em T
1: FastMTM(T,D)
Passo II: Cálculo do peso das arestas
2: **for** cada aresta e_i em T **do**
3: **if** e_i é interna **then**
4: Calcule b_i usando Equação 5.15
5: **else**
6: Calcule b_i usando Equação 5.16
7: **end if**
8: **end for**
9: **retorne** b

distâncias dadas na matriz M. As matrizes consideradas foram as mesmas utilizadas na Seção 5.1.

A implementação do algoritmo BinaryEdges para atribuir pesos às arestas de uma filogenia tradicional está disponível em `https://git.facom.ufms.br/bioinfo/topology/tree/master/bryant-wadell/`.

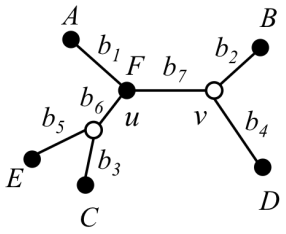
Filogenia viva

Vamos analisar nesta seção o problema de atribuir pesos às arestas de uma dada topologia, quando se trata de uma filogenia viva.

Considere, inicialmente, o método dos mínimos quadrados. Analisando a Figura 5.7, temos um nó interno vivo F na topologia. Ao representarmos a topologia desta figura por meio de equações e também utilizando uma matriz topológica X, conforme já descrito na Seção 5.2, para o modelo dos mínimos quadrados OLS, encontraremos fórmulas equivalentes às apresentadas na Equação 5.4 (pág. 80) pois, uma vez que um ou mais nós internos vivos sejam introduzidos na topologia, a forma de preenchimento da matriz X não será alterada. Além disso, para obtermos uma estimativa de b pelo método OLS, para a topologia da Figura 5.7, devemos resolver um conjunto de equações lineares, da mesma maneira como apresentado na Seção 5.2, Equação 5.4 (pág. 80), ou seja, são equações muito similares, em que a diferença se encontra apenas nos coeficientes das variáveis, sendo uma equação para cada aresta

	A	B	C	D	E	F
A	0	23	16	20	17	8
B	23	0	23	17	24	15
C	16	23	0	15	11	8
D	20	17	15	0	21	12
E	17	24	11	21	0	9
F	8	15	8	12	9	0

(a)



(b)

Figura 5.7: (a) Uma matriz de distâncias entre 6 objetos. (b) Uma topologia viva para atribuir pesos (b_i) às arestas, respeitando as distâncias na matriz em (a).

da topologia.

As equações para a topologia viva dada na Figura 5.7 são:

$$\begin{aligned}
AB + AC + AD + AE + AF &= 5b_1 + b_2 + b_3 + b_4 + b_5 + 2b_6 + 2b_7 \\
BA + BC + BD + BE + BF &= b_1 + 5b_2 + b_3 + b_4 + b_5 + 2b_6 + 4b_7 \\
CA + CB + CD + CE + CF &= b_1 + b_2 + 5b_3 + b_4 + b_5 + 4b_6 + 2b_7 \\
DA + DB + DC + DE + DF &= b_1 + b_2 + b_3 + 5b_4 + b_5 + 2b_6 + 4b_7 \\
EA + EB + EC + ED + EF &= b_1 + b_2 + b_3 + b_4 + 5b_5 + 4b_6 + 2b_7 \\
EA + EF + EB + ED + CA + CF &= 2b_1 + 2b_2 + 4b_3 + 2b_4 + 4b_5 + 8b_6 + 4b_7 \\
+CB + CD & \\
AB + AD + FB + FD + EB + ED &= 2b_1 + 4b_2 + 2b_3 + 4b_4 + 2b_5 + 4b_6 + 8b_7. \\
+CB + CD & \qquad (5.17)
\end{aligned}$$

Vamos descrever agora como é a configuração geral dessas equações. A configuração é válida para filogenia viva e também para filogenia tradicional. Temos dois tipos de configurações: uma para aresta externa e outra para aresta interna. Cada equação possui uma variável para cada aresta na topologia, representando o seu peso, o qual queremos determinar. Para a formação de cada equação de aresta, consideramos todos os caminhos entre objetos que passam por ela.

Analisando a Figura 5.7(b), temos o seguinte formato de equação para uma aresta externa (e_i).

$$\ldots + (n-1)b_i + \ldots + \mathscr{C}b_j + \ldots = k_i \qquad (5.18)$$

em que n é o número de objetos, $k_i \in \mathbb{N}^*$ representa a soma dos pesos de todos os caminhos entre objetos que passam por e_i. Na formação da equação de e_i, o coeficiente de b_j será $\mathscr{C} = 1$ se b_j corresponde ao peso de uma aresta externa. Caso b_j corresponda ao peso de uma aresta interna $e_j = (u, v)$, com o número de objetos à esquerda de u igual a n_1 e à direita de v igual a n_2, o coeficiente de b_j será $\mathscr{C} = n_2$ (se e_i estiver em alguma subárvore à esquerda do nó u) ou $\mathscr{C} = n_1$ (se e_i estiver em alguma subárvore à direita do nó v). Note que, se u ou v forem nós internos vivos, eles irão contribuir na contagem para n_1 ou n_2.

Já uma equação para aresta interna $e_j = (u, v)$, com n_1 objetos à esquerda de u e n_2 objetos à direita de v, terá o seguinte formato:

$$\ldots + \mathscr{C} b_i + \ldots + n_1 n_2 b_j + \ldots = k_j \tag{5.19}$$

em que $k_j \in \mathbb{N}^*$, $\mathscr{C} = n_2$ (se e_i está localizada à esquerda de u) ou $\mathscr{C} = n_1$ (se e_i está localizada à direita de v). Novamente, se u ou v forem vivos, deverão ser contados para n_1 ou n_2.

No caso da filogenia viva, o que muda na formação das equações é o fato de que, ao considerarmos uma aresta interna ou externa, os nós internos poderão ser nós vivos e, então, deverão ser considerados na contagem do número de objetos e irão contribuir na formação dos coeficientes dos pesos das arestas. É importante notar que tais coeficientes nunca serão valores menores ou iguais a zero, visto que são formados pelo número de vezes que os caminhos entre os objetos passam por cada aresta.

O conjunto de equações lineares obtido a partir das derivações obtidas da Equação 5.3, sejam elas obtidas de uma topologia com nós internos vivos ou não, constitui um sistema linear formado por m equações e m incógnitas, em que m é o número de arestas da árvore e pode ser resolvido usando, por exemplo, o método dos mínimos quadrados.

Considerando agora o algoritmo de Bryant e Wadell, apresentado na seção anterior, pode-se observar que precisam ser considerados dois pontos na tentativa de criar um algoritmo para o caso da filogenia viva. O primeiro é em relação ao cálculo de $\delta^t D$, feito no algoritmo FastMTM e o segundo é o cálculo do peso de cada aresta feito no algoritmo BinaryEdges, cuja obtenção da formulação foi apresentada com um

pouco mais de detalhes em [9].

Como uma filogenia viva pode possuir nós internos vivos, isto precisa ser levado em conta ao se realizar o cálculo de $\delta^t D$. Observando a Figura 5.6(b), no caso de uma aresta externa $e_i = (x, u)$, em que x é folha e u é nó interno, u poderia ser acrescentado para algum dos conjuntos C_j ou C_k. Já no caso de uma aresta interna $e_i = (u, v)$ (veja Figura 5.6(a) e Equação 5.14), suponha que u seja o nó interno no qual incidem e_i, e_j e e_k, u deveria contribuir também no cálculo da Equação 5.14.

Além disso, as fórmulas para cálculo do peso das arestas internas (Equação 5.15) e externas (Equação 5.16) deveriam ser reescritas, considerando o mesmo modo de desenvolvimento apresentado em [9]. Tais alterações não foram feitas neste trabalho e ficarão como trabalhos futuros.

5.4 Comentários

O problema de atribuir pesos às arestas de uma topologia viva, no caso de uma matriz não aditiva, pode ser resolvido usando o método dos mínimos quadrados, conforme visto na seção anterior. Os testes com matrizes não aditivas não foram realizados efetivamente neste trabalho e ficarão como trabalhos futuros.

Vale lembrar que os cálculos efetuados no método dos mínimos quadrados permitem gerar pesos de arestas negativos, o que não tem muito significado biologicamente.

Dois trabalhos obtiveram bons resultados nessa abordagem de problema no caso da filogenia tradicional, restringindo os pesos das arestas a serem positivos. O primeiro trabalho, de autoria de Felsenstein [36], encontra os pesos de arestas por mínimos quadrados para três arestas da topologia por vez, podando todas as demais arestas e resolvendo exatamente para as três arestas restantes. Repetindo esta operação para diferentes partes da topologia, o método se aproxima assintoticamente de uma solução de mínimos quadrados.

O segundo trabalho, de Makarenkov e Leclerc [62], utiliza a ideia do sistema de equações lineares, apresentado na seção dos mínimos quadrados (seção 5.2). No entanto, aplicam um método de Gauss-Seidel levemente modificado para encontrar soluções não negativas para o problema.

Ambos os resultados podem ser utilizados como base para tentar obter uma solução desta abordagem no caso da filogenia viva.

Capítulo 6

Conclusão

A construção de filogenias é um componente essencial em pesquisas modernas nas áreas da Medicina e da Biologia, para descobrir novas drogas, entender rapidamente as mutações de patógenos, a dispersão de espécies, a evolução dos genomas, dentre outras aplicações. As filogenias são construídas com base em dados resultantes das comparações entre as espécies, que podem ser distâncias ou características.

Neste trabalho aprofundamos um pouco mais o formalismo na descrição do problema da filogenia viva baseada em distâncias, uma nova classe de problemas de filogenia, usando como dados de entrada as distâncias entre as espécies. Este novo conceito considera que objetos possam existir no presente momento e, ao mesmo tempo, serem também ancestrais de outros objetos, o que permite analisar, por exemplo, populações de vírus ou outros organismos de evolução rápida. Como vimos, as árvores filogenéticas vivas também podem ser usadas na análise de objetos não biológicos, tais como documentos, imagens e entradas de bancos de dados relacionais grandes, melhorando as técnicas de mineração para repositório de dados.

Contribuições

O escopo deste trabalho incluiu o problema da filogenia viva baseada em distâncias entre objetos quando a matriz de entrada não é aditiva. Para isso, foram revisados e apresentados os conceitos relacionados à construção de filogenia viva quando os dados são aditivos. O *abstract* deste trabalho foi publicado em [1].

Para o caso em que os dados não são aditivos, foi provado que o problema da construção de filogenia viva baseada em distância é NP-difícil. Além disso, uma heurística baseada em promoções de folhas à nó interno vivo foi proposta. Adicionado a estes dois resultados, também foi proposto um índice de não aditividade sobre as matrizes de dados de entrada, que permitiu analisar melhor o desempenho da heurística. Todos estes resultados foram publicados em uma Conferência Internacional no ano de 2017 [2].

Outra contribuição do trabalho é a heurística LNJ utilizada na construção de filogenias vivas baseadas em distâncias para o caso em que os dados não são aditivos e projetada a partir do raciocínio matemático utilizado em NJ. A heurística foi aplicada sobre diferentes conjuntos de genomas virais e bacterianos, introduzindo diferentes hipóteses para a relação dessas espécies. A heurística e os resultados de sua aplicação sobre dados não biológicos e dados de vírus e bactérias foram publicados na revista *BMC Bioinformatics* [94].

Uma abordagem diferente o problema da filogenia também foi apresentada em nosso trabalho, na qual o problema consiste em atribuir pesos às arestas de uma topologia dada, de modo a atender as distâncias dadas na matriz. Inicialmente, foi feita uma revisão bibliográfica do problema para a filogenia tradicional. Na sequência, descrevemos o problema para o caso da filogenia viva, em que um algoritmo de tempo linear foi apresentado para o caso onde a matriz é aditiva. Já no caso da matriz ser não aditiva, mostramos que, para uma filogenia viva, o método dos mínimos quadrados pode ser aplicado, visto que as equações lineares geradas são geradas da mesma maneira como nas equações no caso de uma topologia tradicional, com a modificação de levar em conta o nó interno vivo, quando este aparecer.

Trabalhos Futuros

Este trabalho é um dos primeiros a abordar o problema da filogenia viva baseada em distâncias. Muitas possibilidades podem ainda ser exploradas no tema. Entre elas, podemos citar:

- análise das distâncias dos nós internos vivos às folhas;
- estudo do impacto das triplas e quádruplas no índice de não aditividade;

- desenvolvimento de extensões que tornem LNJ mais rápida, assim como existem extensões para NJ;
- implementação e teste do algoritmo de Bryant e Wadell [11], considerando a presença de objetos vivos como nós internos, para o caso não aditivo;
- desenvolvimento de novos algoritmos para atribuir pesos às arestas quando a topologia é dada, fazendo restrição de positividade aos pesos das arestas, considerando uma topologia de árvore viva.

Referências Bibliográficas

[1] G.S. Araújo, G.P. Telles, N.F. Almeida, e M.E.M.T. Walter. Distance-based live phylogeny. In *Proceedings of the third International Society for Computational Biology Latin America X-Meeting on Bioinformatics with BSB and SoiBio*. Belo Horizonte, Brazil, 2014. (abstract).

[2] G.S. Araújo, G.P. Telles, N.F. Almeida, e M.E.M.T. Walter. Distance-based live phylogeny. In *Proceedings of the 10th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2017)*, volume 3, pp. 196–201. Porto, Portugal, 2017. DOI:10.5220/0006224501960201.

[3] K. Atteson. The performance of the Neighbor-Joining method of phylogeny reconstruction. *Algorithmica*, 25:251–278, 1999.

[4] W. Baker, A. Broek, E. Camon, P. Hingamp, P. Sterk, G. Stoesser, e M. A. Tuli. The EMBL nucleotide sequence database. *NAR*, 32(1):19–23, 2000.

[5] H. Bandelt e A.W.M. Dress. Split decomposition: A new and useful approach to phylogenetic analysis of distance data. *Molecular Phylogenetics and Evolution*, 1(3):242 – 252, 1992. ISSN 1055-7903.

[6] D. Benson, M. Cavanaugh, K. Clark, et al. GenBank. *NAR*, 41:D36–42, 2013.

[7] W.A. Beyer, M.L. Stein, T.F. Smith, e S.M. Ulam. A molecular sequence metric and evolutionary trees. *Mathematical Biosciences*, 19(1):9 – 25, 1974. ISSN 0025-5564.

[8] W.J. Bruno, N.D. Socci, e A.L. Halpern. Weighted Neighbor-Joining: A likelihood-based approach to distance-based phylogeny reconstruction. *Mol. Biol. and Evolution*, 17(1):189–197, 2000.

[9] D. Bryant. *Building Trees, Hunting for Trees, and Comparing Trees – Theory and Methods in Phylogenetic Analysis*. Tese de Doutoramento, University of Canterbury, Christchurch, 1997.

[10] D. Bryant. On the uniqueness of the selection criterion in Neighbor-Joining. *Journal of Classification*, 22(1):3–15, Jun 2005. ISSN 1432-1343.

[11] D. Bryant e P. Waddell. Rapid evaluation of least-squares and minimum-evolution criteria on phylogenetic trees. *Mol. Biol. and Evolution*, 15(10):1346–1359, 1998.

[12] M. Bulmer. Use of the Method of Generalized Least Squares in Reconstructing Phylogenies from Sequence Data. *Mol. Biol. and Evolution*, 8(6):868–868, 11 1991. ISSN 0737-4038.

[13] S.A. Carroll, J.S. Towner, T.K. Sealy, L.K. McMullan, M.L. Khristova, F.J. Burt, R. Swanepoel, P.E. Rollin, e S.T. Nichol. Molecular evolution of viruses of the family Filoviridae based on 97 whole-genome sequences. *J. Virol.*, 87(5):2608–2616, 2013.

[14] E. Castro-Nallar, M. Perez-Losada, G.F. Burton, e K.A. Crandall. The evolution of HIV: Inferences using phylogenetics. *Molecular Phylogenetics Evol.*, 62:777–792, 2012.

[15] D. Catanzaro. The minimum evolution problem: Overview and classification. *Networks*, 53(2):112–125, 2008.

[16] L.L. Cavalli-Sforza e A.W.F. Edwards. Phylogenetic analysis: Models and estimation procedures. *Evolution*, 21(3):550–570, 1967. ISSN 00143820, 15585646.

[17] Centers for Disease Control and Prevention. About Ebola virus. https://www.cdc.gov/vhf/ebola/about.html, 2017.

[18] R. Chakraborty. Estimation of time of divergence from phylogenetic studies. *Canadian J. of Genetics and Cytology*, 19(2):217–223, 1977. PMID: 196728.

[19] M.E. Cosner, R. Jansen, B. Moret, L. Raubeson, L. Wang, e S. Wyman. A new fast heuristic for computing the breakpoint phylogeny and experimental analyses of real and synthetic data. In *Proceedings of the 8th Internatinal Conference on Intelligent Systems for Molecular Biology*, pp. 104–115. La Jolla, USA, 2000.

[20] A.M. Cuadros, F.V. Paulovich, R. Minghim, e G.P. Telles. Point placement by phylogenetic trees and its application to visual analysis of document collections. In *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology*, pp. 99–106. 2007.

[21] A.E.A. da Silva, W.J.P. Villanueva, H. Knidel, V. Bonato, S.F. dos Reis, e F.J. Von Zuben. A multi-Neighbor-Joining approach for phylogenetic tree reconstruction and visualization. *Genet Mol Res*, 4(3):525–534, 2005.

[22] W.E. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bulletin of Mathematical Biology*, 49(4):461–467, 1987.

[23] Y.V. de Peer. Phylogenetic inference based on distance methods: theory. In P. Lemey, M. Salemi, e A. Vandamme, editores, *The phylogenetic handbook : a practical approach to phylogenetic analysis and hypothesis testing*, pp. 142–160. Cambridge University Press, 2009. ISBN 9780521877107.

[24] A.L. Delcher, S. Kasif, R.D. Fleischmann, J. Peterson, O. White, e S.L. Salzberg. Alignment of whole genomes. *NAR*, 27(11):2369–2376, 1999.

[25] M. Deloger, M. El-Karoui, e M.A. Petit. A genomic distance based on MUM indicates discontinuity between most bacterial species and genera. *Journal of Bacteriology*, 191(1):91–99, 2009.

[26] R. Desper e O. Gascuel. Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. *Journal of Computational Biology*, 9(5):687–705, 2002. PMID: 12487758.

[27] R. Desper e O. Gascuel. The minimum evolution distance-based approach to phylogenetic inference. *Mathematics of evolution and phylogeny*, pp. 1–32, 2005.

[28] A. Dress, K.T. Huber, e V. Moulton. Metric spaces in pure anda applied mathematics. In *Documenta Mathematica, Special Volume Proceedings Quadratic Forms LSU*, pp. 121–139. 2001.

[29] G. Dudas e A. Rambaut. Phylogenetic analysis of Guinea 2014 EBOV Ebolavirus outbreak. *PLoS Currents: Outbreaks*, 6, 2014. ISSN 2157-3999.

[30] I. Elias e J. Lagergren. Fast Neighbor-Joining. In *Proc. of ICALP*, volume 3580, pp. 1263–1274. 2005.

[31] J. Evans, L. Sheneman, e J. Foster. Relaxed Neighbor-Joining: A fast distance-based phylogenetic tree construction method. *Journal of Molecular Evolution*, 62(6):785–792, 2006.

[32] J.S. Farris. Estimating phylogenetic trees from distance matrices. *The American Naturalist*, 106(951):645–668, 1972. ISSN 00030147, 15375323.

[33] J. Felsenstein. The Newick tree format. http://evolution.genetics.washington.edu/phylip/newicktree.html.

[34] J. Felsenstein. Distance methods for inferring phylogenies: a justification. *Evolution*, 38(1):16–24, 1984.

[35] J. Felsenstein. PHYLIP, phylogeny inference package. *Cladistics*, 5:164–66, 1989.

[36] J. Felsenstein. An alternating least squares approach to inferring phylogenies from pairwise distances. *Systematic biology*, 46(1):101–111, 1997.

[37] J. Felsenstein. *Inferring Phylogenies*. Sinauer Assoc., 2004.

[38] R.L. Fernandes, R. Güths, G.P. Telles, N.F. Almeida, e M.E.M.T. Walter. A genetic algorithm for character state live phylogeny. In R. Alves, editor, *Proc. 11th Brazilian Symposium on Bioinformatics, BSB 2018*, volume 11228 de *Lecture Notes in Bioinformatics*, pp. 114–23. 2018. ISBN 978-3-030-01722-4.

[39] W. Fitch e E. Margoliash. Construction of phylogenetic trees. *Science*, 155(3760):279–284, 1967. ISSN 0036-8075.

[40] O. Gascuel. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Mol. Biol. and Evolution*, 14(7):685–695, 1997.

[41] O. Gascuel. Concerning the nj algorithm and its unweighted version, unj. *Mathematical hierarchies and biology*, 37:149–171, 1997.

[42] O. Gascuel. On the optimization principle in phylogenetic analysis and the minimum-evolution criterion. *Mol. Biol. and Evolution*, 17(3):401–405, 2000.

[43] O. Gascuel. *Mathematics of Evolution and Phylogeny*. Oxford University Press, Inc., New York, NY, USA, 2005. ISBN 0199231346, 9780199231348.

[44] O. Gascuel e M. Steel. Neighbor-Joining revealed. *Mol. Biol. and Evolution*, 23(11):1997–2000, 2006.

[45] I. Gat-Viks, R. Shamir, e H. Wolfson. Computational genomics. Disponível em `http://www.cs.tau.ac.il/~rshamir/cg/16/`, 2016. Lecture Notes, Tel Aviv University.

[46] T. Gojobori, E.N. Moriyama, e M. Kimura. Molecular clock of viral evolution, and the neutral theory. *P. Natl. Acad. Sci.*, 87(24):10015–10018, 1990.

[47] D. Gusfield. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, 1997.

[48] R. Güths, G.P. Telles, M.E.M.T. Walter, e N.F Almeida. A heuristic for the live parsimony problem. In Nathalia Peixoto, Margarida Silveira, Hesham H. Ali, Carlos Maciel, e Egon L. van den Broek, editores, *Biomedical Engineering Systems and Technologies*, pp. 248–267. Springer International Publishing, 2018. ISBN 978-3-319-94806-5.

[49] R. Güths, G.P. Telles, M.E.M.T. Walter, e N.F. Almeida. A branch and bound for the large live parsimony problem. In *Proceedings of the 10th International Joint Conference on Biomedical Engineering Systems and Technologies*, pp. 184–189. 01 2017.

[50] M. Hasegawa, H. Kishino, e T. Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial dna. *Journal of Molecular Evolution*, 22(2):160–174, Oct 1985. ISSN 1432-1432.

[51] K. Howe, A. Bateman, e R. Durbin. Quicktree: building huge Neighbour-Joining trees of protein sequences. *Bioinformatics*, 18(11):1546–1547, 2002.

[52] N. M. Kopelman, L. Stone, O. Gascuel, e N. A. Rosenberg. The behavior of admixed populations in neighbor-joining inference of population trees. In *Biocomputing 2013*, pp. 273–284. World Scientific, 2013.

[53] M.K. Kuhner e J. Felsenstein. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol. Biol. and Evolution*, 11(3):459–468, 1994.

[54] T. Kulikova, R. Akhtar, P. Aldebert, N. Althorpe, M. Andersson, A. Baldwin, K. Bates, S. Bhattacharyya, L. Bower, P. Browne, M. Castro, G. Cochrane, K. Duggan, R. Eberhardt, N. Faruque, G. Hoad, C. Kanz, C. Lee, R. Leinonen, Q. Lin, V. Lombard, R. Lopez, D. Lorenc, H. McWilliam, G. Mukherjee,

F. Nardone, M. P. G. Pastor, S. Plaister, S. Sobhany, P. Stoehr, R. Vaughan, D. Wu, W. Zhu, e R. Apweiler. EMBL Nucleotide Sequence Database in 2006. *Nucleic Acids Research*, 35(suppl_1):D16–D20, 12 2006. ISSN 0305-1048.

[55] R.S. Lanciotti, A.J. Lambert, M. Holodniy, S. Saavedra, e L.d.C.C. Signor. Phylogeny of Zika virus in western hemisphere, 2015. *Emerging Infectious Diseases*, 22(5):933–935, 2016.

[56] J.F. Li. A fast Neighbor-Joining method. *Genetics and Molecular Research*, 14(3):8733–8743, 2015.

[57] M. Li, X. Chen, X. Li, B. Ma, e P. Vitanyi. The similarity metric. *IEEE Trans. Information Theory*, 50(12):3250–3264, 2004.

[58] J. Lu e H. Carlson. Chemtreemap: an interactive map of biochemical similarity in molecular datasets. *Bioinformatics*, 32(23):3584–3592, 2016.

[59] T. Mailund e C.N.S. Pedersen. Quickjoin–fast Neighbour-Joining tree reconstruction. *Bioinformatics*, 20(17):3261–3262, 2004.

[60] V. Makarenkov, D. Kevorkov, e P. Legendre. 3-phylogenetic network construction approaches. *Applied mycology and biotechnology*, 6:61–97, 2006.

[61] V. Makarenkov e F. Lapointe. A weighted least-squares approach for inferring phylogenies from incomplete distance matrices. *Bioinformatics*, 20 13:2113–21, 2004.

[62] V. Makarenkov e B. Leclerc. An algorithm for the fitting of a tree metric according to a weighted least-squares criterion. *Journal of Classification*, 16:3–26, 01 1999.

[63] R. Mihaescu, D. Levy, e L. Pachter. Why Neighbor-Joining works. *Algorithmica*, 54:1–24, 2009.

[64] B. Mirkin. *Mathematical classification and clustering*. Kluwer Academic Press, London, 1996.

[65] A. Moya, S. Elena, A. Bracho, R. Miralles, e E. Barrio. The evolution of RNA viruses: a population genetics view. *PNAS*, 97(13):6967–6973, 2000.

[66] D. Musso e D.J. Gubler. Zika virus. *Clin Microbiol Rev*, 29(3):487–524, 2016.

[67] L. Nakhleh, B.M. Moret, U. Roshan, K. St-John, J. Sun, e T. Warnow. The accuracy of phylogenetic methods for large datasets. In *Proceedings of Fifth Pacific Symposium of Biocomputing (PSB02)*, pp. 211–222. Hawaii, USA, 2002.

[68] M. Nunes, N. Faria, J. Vasconcelos, N. Golding, M. Kraemer, L. de Oliveira, R. Azevedo, D. da Silva, E. da Silva, S.P. da Silva, V. Carvalho, G. Coelho, A. Cruz, S. Rodrigues, J. Vianez, B. Nunes, J. Cardoso, R. Tesh, S. Hay, O. Pybis, e P. Vasconcelos. Emergence and potential for spread of Chikungunya virus in Brazil. *BMC Medicine*, 13(1):102, 2015.

[69] S. Ota e W. Li. NJML: A hybrid algorithm for the Neighbor-Joining and maximum-likelihood methods. *Mol. Biol. and Evolution*, 17(9):1401–09, 2000.

[70] J. G. Paiva, L. Florian, H. Pedrini, G. P. Telles, e R. Minghim. Improved similarity trees and their application to visual data classification. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2459–2468, 2011.

[71] Y. Pauplin. Direct calculation of a tree length using a distance matrix. *Journal of Molecular Evolution*, 51(1):41–47, Jul 2000. ISSN 1432-1432.

[72] W.R. Pearson, G. Robins, e T. Zhang. Generalized Neighbor-Joining: more reliable phylogenetic tree reconstruction. *Mol. Biol. and Evolution*, 16(6):806–816, 1999.

[73] S. Pompei, V. Loreto, e F. Tria. Phylogenetic properties of RNA viruses. *PLoS ONE*, 7(9):1–10, 2012.

[74] K.M. Potter, A.R. Campbell, S.A. Josserand, C.D. Nelson, e R.M. Jetton. Population isolation results in unexpectedly high differentiation in Carolina hemlock (Tsuga caroliniana), an imperiled southern Appalachian endemic conifer. *Tree Genetics & Genomes*, 13(5):105, Sep 2017. ISSN 1614-2950.

[75] C. Prieto e J. Lorenc-Brudecka. Description of Rhamma dawkinsi (Lepidoptera: Lycaenidae) a new mountain butterfly from Colombia. *Zootaxa*, 4338(3):587–594, 2017. ISSN 1175-5334.

[76] T. Pyrcz, C. Prieto, P. Boyer, e J. Lorenc-Brudecka. Discovery of a remarkable new species of Lymanopoda (Lepidoptera: Nymphalidae: Satyrinae) and considerations of its phylogenetic position: an integrative taxonomic approach. *EJE*, 115(1):387–399, 2018. ISSN 12105759.

[77] F. Riquet, C. Liautard-Haag, L. Woodall, C. Bouza, P. Louisy, B. Hamer, F. Otero-Ferrer, P. Aublanc, V. Beduneau, O. Briard, T. El-Ayari, S. Hochschei, K. Belkhir, S. Arnaud-Haond, P. Gagnaire, e N. Bierne. Parallel pattern of differentiation at a genomic island shared between clinal and mosaic hybrid zones in a complex of cryptic seahorse lineages. *bioRxiv*, 2018.

[78] A. Rzhetsky e M. Nei. A simple method for estimating and testing minimum-evolution trees. *Mol. Biol. and Evolution*, 9:945–967, 1992.

[79] A. Rzhetsky e M. Nei. Theoretical foundation of the minimum-evolution method of phylogenetic inference. *Mol. Biol. and Evolution*, 10(5):1073–1095, 1993.

[80] A. Rzhetsky e M. Nei. METREE: A program package for inferring and testing minimum-evolution trees. *Computer applications in the biosciences : CABIOS*, 10:409–12, 08 1994.

[81] N. Saitou e T. Imanishi. Relative efficiencies of the Fitch-Margoliash, maximum-parsimony, maximum likelihood, minimum-evolution and Neighbor-Joining methods of phylogenetic tree construction in obtaining the correct tree. *Mol. Biol. and Evolution*, 6:514–525, 1989.

[82] N. Saitou e M. Nei. The Neighbor-Joining Method: a new method for reconstructing phylogenetic trees. *Mol. Biol. and Evolution*, 4(4):406–425, 1987.

[83] J.C. Setubal, N.F. Almeida, e A.R. Wattam. Comparative genomics for prokaryotes. In J.C. Setubal, J. Stoye, e P.F. Stadler, editores, *Comparative Genomics: Methods and Protocols*, volume 1704 de *Methods in Mol. Biology*. Springer Science+Business Media LLC, Springer International Publishing AG, 2018.

[84] J.C. Setubal e J. Meidanis. *Introduction to Molecular Computational Biology*. PWS, 1997.

[85] M. Simonsen e T. Mailund C.N.S. Thomas. Rapid Neighbour-Joining. In *Proc. of WABI*, volume 5251 LNBI, pp. 113–122. 2008.

[86] P. Sneath e R. Sokal. Numerical taxonomy: the principles and practice of numerical classification, 1973.

[87] R.N. Souza. Algoritmo para matrizes aditivas. Disponível em `http://www.ime.usp.br/~rsouza`, 1999. Lecture Notes, Instituto de Matemática e Estatística da Universidade de São Paulo.

[88] A. Stamatakis. RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 2014.

[89] M. Steel. *Phylogeny: discrete and random processes in evolution*. SIAM, 2016.

[90] J.E. Studier e K.J. Keppler. A note on the Neighbor-Joining algorithm of Saitou and Nei. *Mol. Biology and Evolution*, 5(5):729–731, 1988.

[91] Y. Tateno, N. Takezaki, e M. Neiji. Relative efficiencies of the maximum-likelihood, Neighbor-Joining, and maximum-parsimony methods when substitution rate varies with site. *Mol. Biol. and Evolution*, 11(2):261–277, 1994.

[92] E. Tejada, R. Minghim, e L. G. Nonato. On improved projection techniques to support visual exploration of multidimensional data sets. *Information Visualization*, 2(4):218–231, 2003.

[93] G.P. Telles, N.F. Almeida, R. Minghim, e M.E.M.T. Walter. Live phylogeny. *Journal of Computational Biology*, 20(1):30–37, 2013.

[94] G.P. Telles, G.S. Araújo, M.E.M.T Walter, M.M. Brigido, e N.F. Almeida. Live Neighbor-Joining. *BMC Bioinformatics*, 19(1):172, May 2018. DOI:10.1186/s12859-018-2162-x.

[95] W. Vach. Least squares approximation of additive trees. In Otto Optiz, editor, *Conceptual and Numerical Analysis of Data*, pp. 230–238. Springer Berlin Heidelberg, Berlin, Heidelberg, 1989. ISBN 978-3-642-75040-3.

[96] L. Wang, R. Jansen, B. Moret, L. Raubeson, e T. Warnow. Fast phylogenetic methods for genome rearrangement evolution: an empirical study. In *Proceedings of Fifth Pacific Symposium of Biocomputing (PSB02)*, pp. 524–535. Hawaii, USA, 2002.

[97] Z. Wang, A.C. Bovik, H.R. Sheikh, e E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[98] M. Ward, G. Grinstein, e D. Keim. *Interactive Data Visualization: Foundations, Techniques, and Applications*. CRC Press, Boca Raton, FL, 2015.

[99] M. Waterman, T. Smith, e W. Beyer. Additive evolutionary trees. *Journal Theorical Biology*, 64(2):199–213, 1977.

[100] T.J. Wheeler. Large-scale Neighbor-Joining with NINJA. In *Proc. of WABI*, pp. 375–389. 2009.

[101] C. Yu, B. Baune, J. Licinio, e M. Wong. A novel strategy for clustering major depression individuals using whole-genome sequencing variant data. *Scientific Reports*, 7(1):44389, 2017. Exported from https://app.dimensions.ai on 2018/09/25.

[102] Z. Zhang, Y. Yoo, R. Kulathinal, e S. Wattal. Characterizing generative digital artifacts: The case of web mashups. Relatório técnico, Organizational Genetics Working Paper, Philadelphia, PA, USA, 2015.

[103] M. Zvelebil e J. O. Baum. *Understanding Bioinformatics*. Garland Science, 2008.