Gerência e Análises de Workflows aplicados a Redes Filogenéticas de Genomas de Dengue no Brasil

Rafael Terra[1], Micaella Coelho[1], Lucas Cruz[1,2], Marco Garcia-Zapata[3], Luiz Gadelha[1], Carla Osthoff[1], Diego Carvalho[2], Kary Ocaña[1]

[1]Laboratório Nacional de Computação Científica
(LNCC) Petrópolis – RJ – Brasil

[2] Centro Federal de Educação Tecnológica Celso Suckow da Fonseca
(CEFET/RJ) – RJ – Brasil

[3] Universidade Federal de Goiás
(UFG) – Goiânia – Brasil

{rafaelst,micaella,lucruz,lgadelha,osthoff,karyann}@lncc.br,
d.carvalho@ieee.org, mctulianglobal@gmail.com

Abstract. *Evolutionary processes and dispersion of Dengue genomes in Brazil are relevant for guiding the impact and epidemiological surveillance on emerging arboviruses. Phylogenetic trees and networks can exhibit evolutionary and reticulated events in viruses due to the high diversity, high mutation rate and frequent homologous recombination. We present a parallel scientific workflow for phylogenetic networks designed to cover resources and task heterogeneity from computational biology experiments coupled to high-performance computing environments. We present an improvement in execution time of approximately 5 times compared to a sequential execution for analyzing dengue genomes with the identification of recombination events.*

Resumo. *Processos evolutivos e dispersão de genomas de Dengue no Brasil são relevantes na direção do impacto e vigilância endemo-epidêmico e social de arboviroses emergentes. Árvores e redes filogenéticas permitem exibir eventos evolutivos e reticulados em vírus originados pela alta diversidade e taxa de mutação de recombinação homóloga frequente. Apresentamos um workflow científico paralelo e distribuído para redes filogenéticas desenhado para trabalhar com a diversidade de ferramentas e recursos em experimentos da biologia computacional e acoplados a ambientes de computação de alto desempenho. Apresentamos uma melhoria no tempo de execução de aproximadamente 5 vezes em comparação com a execução sequencial em análises de genomas de dengue e com identificação de eventos de recombinação.*

1. Introdução

A incidência global da dengue cresceu drasticamente nas últimas décadas e estima-se que ocorram entre 100 a 400 milhões de infecções por ano [OPAS 2021]. A dengue é encontrada em climas tropicais e subtropicais em todo o mundo e a sua forma grave é uma das principais causas de complicações e morte de crianças em alguns países da Ásia e da América Latina. No norte do Brasil ela é classificada como uma doença endemo-epidêmica. No sul, antes com baixo surtos de dengue, atualmente tem-se um crescimento

alarmante e, em meio à pandemia do Coronavírus, o Brasil já ultrapassa 900 mil casos de dengue [WHO 2021], sendo o DENV o vírus responsável por causar a doença. Até o momento, foram observados cinco tipos desse vírus: DENV-1, DENV-2 e DENV-3, presentes no Brasil; DENV-4 que é mais comum na Costa Rica e Venezuela; e DENV-5 identificado em 2007 na Malásia, Ásia.

As metodologias de filogenia e evolução molecular computacional possibilitam um melhor entendimento sobre o comportamento evolutivo de genomas emergentes do vírus da dengue. Investigar processos evolutivos e de dispersão endemo-epidêmicas de dengue no Brasil possibilita o apoio na assistência à saúde, através da aplicação de diagramas de controle e da análise espacial de doenças endêmicas de interesse sanitário nacional. O impacto epidemiológico e social relevante na direção de reduzir as lacunas atuais do conhecimento sobre dengue e outras arboviroses pode reorientar às práticas vigentes da vigilância epidemiológica desses agravos.

O objetivo do presente trabalho visa apresentar o desenvolvimento de *workflows* científicos para redes filogenéticas aplicadas a análises de sequências genômicas de dengue do Brasil, com o uso de ambientes de Computação de Alto Desempenho (CAD) e tecnologias de análises de dados, pois a construção de redes genômicas da dengue é um processo intensivo em tempo e gasto computacional. Com esse objetivo, o restante deste artigo segue organizado da seguinte forma: a Seção 2 apresenta o *background* sobre árvores e redes filogenéticas; a Seção 3 apresenta o *workflow* para redes filogenéticas proposto; a Seção 4 apresenta análises de desempenho do *workflow* de redes filogenéticas em genomas da dengue; e a Seção 5 traz as considerações finais e trabalhos futuros.

2. *Background* sobre Redes Filogenéticas

Análises filogenéticas representadas por árvores fornecem informações cruciais para reconstruir os parâmetros dos principais eventos evolutivos que promoveram a origem e a disseminação, por exemplo, de vírus. A evolução reticulada se refere à origem das linhagens por meio da fusão completa ou parcial de linhagens ancestrais. Redes podem ser usadas para representar eventos de independência de linhagem -incluindo recombinação, hibridização, poliploidia de entrada, fusão de genoma, endossimbiose e transferência de gene horizontal- em processos filogenéticos inconsistentes em árvores [Huson et al. 2010].

A história evolutiva pode ser ilustrada como gráficos de redes ou árvores. Uma árvore filogenética é um diagrama composto por folhas chamadas de táxons (*e.g.* espécies de interesse), ramos ou braços que representam a direção evolutiva ocorrida e ramificações entre o Último Ancestral Comum (UAC) e as espécies resultantes da ramificação [Baum et al. 2008]. Matematicamente, as árvores e redes são grafos $G = (N, A)$, onde os nós ($i \in N$) representam os objetos e são conectados por arestas ($(i, j) \in A$), que identificam alguma forma de relacionamento. As arestas podem ser não-direcionadas ou direcionadas, quando $(i, j) \neq (j, i)$. No caso de grafos direcionados, a direção indica algum tipo de relação assimétrica entre os objetos. Na análise filogenética, dois tipos de grafos são frequentes: as redes implícitas que apresentam meios para compreender sinais filogenéticos incompatíveis e as redes explícitas as quais apresentam os cenários de evolução reticulada [Huson et al. 2010].

3. *Workflow* Científico para Redes Filogenéticas

Esta seção apresenta a modelagem conceitual do *workflow* proposto para análise de redes filogenéticas, suas tarefas e as principais características da sua arquitetura. Nessa modelagem, consideramos um *pipeline* como uma sequência linear de tarefas executadas por ferramentas computacionais utilizadas na obtenção de resultados intermediários, que são apresentadas nas seções seguintes. Por outro lado, consideramos que um *workflow* é composto por um ou mais *pipelines*, onde as tarefas serão orquestradas através dos recursos computacionais sob responsabilidade de um gerenciador para o mapeamento e controle das tarefas, objetivando a produção de resultados científicos finais.

3.1. *Pipeline* para Anotação de Genomas de Flavivírus

A Figura 1 exibe a representação do *workflow* para a geração de redes filogenéticas apresentado nesse trabalho. A primeira atividade usa o *pipeline* FLAVi[1] [Schneider et al. 2020]. FLAVi foi desenvolvido para o estudo de filogenia de genomas virais de flavivírus e é capaz de suportar a análise de homologia dos genomas processados. Ele utiliza estratégias *ab initio* e baseadas em homologia, sendo testado e alcançando bons resultados na reanotação dos genes representativos da família *flaviviridae*. Esse *pipeline* recebe como entrada um arquivo no formato FASTA com um conjunto de genomas da família Flaviviridae e gera um alinhamento múltiplo dos genomas, juntamente com metadados dos intervalos dos genes anotados nas sequências. A segunda atividade executa um *script* em Python que trata o alinhamento dos genomas de saída do FLAVi e o separa em arquivos de alinhamentos agrupados por gene.

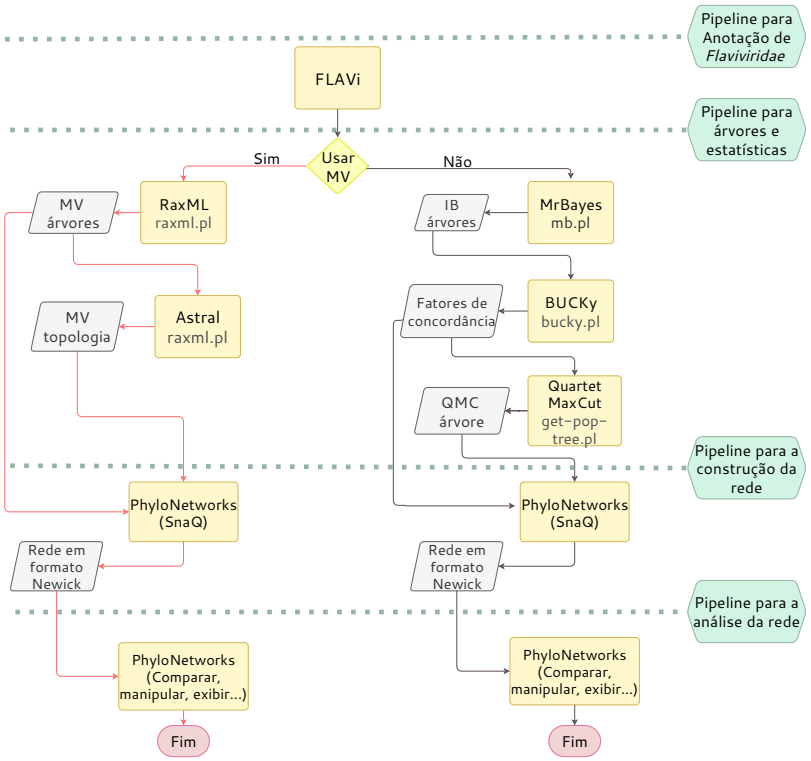

Figura 1. Modelagem Conceitual do *Workflow* de Redes Filogenéticas.

[1]Fast Loci Annotation of Viruses

3.2. *Pipeline* de Construção de Árvores Filogenéticas

A construção de árvores filogenéticas e geração de estatísticas é baseada no *pipeline* *TICR*[2]. Esse *pipeline* apresenta múltiplos fluxos por meio da execução de *scripts* em Perl e em Bash e ainda pode ser facilmente modificado, adicionando-se novos fluxos que fazem uso de outras aplicações. A Figura 1 mostra o fluxo A executando o RAxML [Stamatakis 2014] e o ASTRAL [Zhang et al. 2018] e o fluxo B executando o MrBayes [Ronquist and Huelsenbeck 2003], BUCKy [Larget et al. 2010] e Quartet MaxCut [Snir and Rao 2012]. RAxML é um *software* escrito em linguagem C que pode ser executado em um único *core*, em versão *multi-threaded*, ou usar *Message Passing Interface* (MPI) para a distribuição em diversos nós de processamento. ASTRAL é um código escrito em JAVA que utiliza um único nó de processamento. MrBayes é um *software* escrito em linguagem C com MPI e pode utilizar diversos nós para o processamento. BUCKy e Quartet MaxCut são *scripts* implementados diretamente em Perl.

O fluxo A executa RAxML que consome como entrada arquivos de alinhamento agrupados por genes gerados pelo *pipeline* FLAVi. RAxML utiliza Máxima Verossimilhança (MV) para gerar arquivos com árvores filogenéticas em formato Newick (*e.g. bootstrap* e a melhor árvore) e outros arquivos com estatísticas e metadados. Após isso, o ASTRAL consome o arquivo com a melhor árvore do RAxML e gera um outro conjunto de árvores, entre elas: uma árvore com arestas anotadas e valores de *bootstrap* e outra árvore com arestas anotadas e comprimento de braços em unidades coalescentes.

O fluxo B executa o MrBayes que consome arquivos de alinhamento agrupados por genes gerados pelo FLAVi e gera arquivos em formato NEXUS com as melhores árvores filogenéticas geradas pelo processo do cálculo de probabilidade *aposteriori* do algoritmo de Inferência Bayesiana (IB); além de outros arquivos com estatísticas e metadados. BUCKy consome as melhores árvores do MrBayes, as combina para estimar a proporção de genes e gera arquivos no formato CSV com a matriz de fatores de concordância, listando os conjuntos de 4-táxons *versus* os respectivos fatores de concordância. Quartet MaxCut recebe como entrada a matriz de fatores de concordância do BUCKy e gera uma árvore contendo o número máximo possível dos quartetos de entradas.

3.3. *Pipeline* de Construção de Rede Filogenéticas

As redes filogenéticas são construídas usando o algoritmo SNaQ do PhyloNetworks [Solís-Lemus and Ané 2016]. Ele estima as redes filogenéticas de MV usando o modelo coalescente de multi-espécies em redes. Para gerar a rede inferida em formato Newick, o SNaQ pode receber como entrada as árvores de genes geradas pelo RAxML, juntamente com as árvores de espécies geradas pelo Astral ou a tabela de fatores de concordância junto com a árvore gerada pelo Quartet MaxCut. Um último *pipeline* é executado para a análise da rede usando PhyloPlots[3] e diversas bibliotecas do PhyloNetworks que permitem a análise da rede por meio de operações de visualização, rotação e manipulação de metadados da rede inferida. Ambos SNaQ e PhyloPlots são implementados em linguagem Julia. O SNaQ pode executar as estimações em paralelo através dos pacotes próprios de computação distribuída disponíveis na linguagem de implementação. Essas execuções podem ser feitas na mesma máquina ou em nós distintos.

[2]http://crsl4.github.io/PhyloNetworks.jl/latest/man/ticr_howtogetQuartetCFs/

[3]https://github.com/cecileane/PhyloPlots.jl

3.4. Otimização do *workflow*

O *worflow* para a construção de redes filogenéticas proposto nas seções anteriores foi inspirado pela versão apresentada pelo PhyloNetworks[4] e é composto por ferramentas com características completamente diferentes e com requisitos computacionais bem distintos, que obriga a um desenho criterioso para o uso em ambientes de CAD. Para exemplificar as técnicas utilizadas, vamos analisar o primeiro ramo do *workflow* composto pelo *pipeline* formado por RAxML, ASTRAL e PhyloNetworks (destacado na Figura 1).

De maneira geral, ambos os *pipelines* apresentam padrões intercalados de tarefas que usam um *core* com outras que podem executar em paralelo usando diversos *cores* de um processador ou até diversos processadores espalhados por nós distintos como mostrado na Figura 2 (a). Essa diversidade apresentada entre cada estágio dos *pipelines* utilizados, ou seja, as diferenças entre as tarefas relativas ao número de processadores, linguagens de programação, bibliotecas, modos de processamento (sequencial, paralelo ou distribuído) obriga uma orquestração cuidadosa no uso de recursos computacionais.

Além disso, como os *pipelines* são executados para diversos sequenciamentos virais distintos, existe a oportunidade de se empacotar os *pipelines* de diversas entradas virais distintas para execução simultânea com o intuito de se utilizar todos os recursos computacionais disponíveis, com isso reduzindo o tempo total de processamento do conjunto, como mostrado na Figura 2 (b), onde pode ser observado dois *pipelines* de entradas diferentes utilizando de forma mais eficiente os recursos computacionais disponíveis.

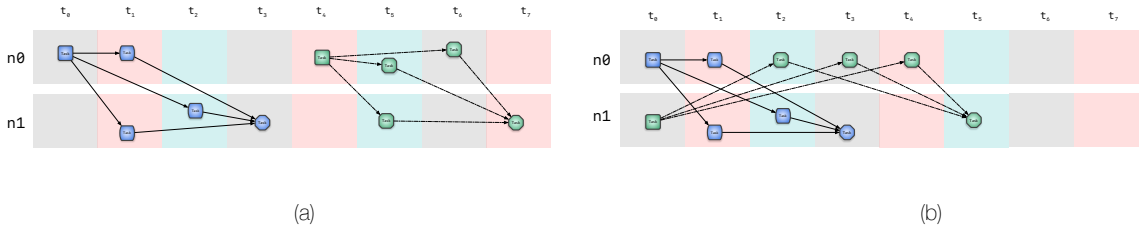

Figura 2. Exemplo de fragmentação interna.

A implementação do *workflow* em ambientes de CAD foi realizada com o uso da biblioteca Parsl [Babuji et al. 2018]. Essa biblioteca incorpora as técnicas de paralelização implícita de *workflows* científicos ao Python, apresentando a abstração de aplicações Python e Bash para a implementação das tarefas dos *pipelines*. Isso oferece a execução das tarefas de uma maneira não intrusiva, ou seja, sem a necessidade de realizar modificações nas ferramentas já existentes. Dentre os recursos do Parsl, pode-se destacar o sistema de tolerância a falhas que combina ressubmissão de aplicações que falham com um modelo de *checkpoints* que permite começar a execução de um *workflow* a partir de um estado intermediário.

O uso da linguagem Python pela biblioteca também se torna um atrativo, pois ela consegue manter os benefícios do paralelismo implícito de ferramentas já conhecidas, como o Swift [Wilde et al. 2011], e ainda ter a facilidade apresentada pela linguagem. Além disso, Parsl apresenta a ideia de executores que são os responsáveis por instanciar as aplicações no recurso computacional disponível, inclusive em partições ou filas distintas. Os executores podem gerenciar recursos heterogêneos, inclusive em infraestruturas

[4]https://github.com/crsl4/PhyloNetworks.jl/wiki

diferentes (*off site*), permitindo mapear atividades em computadores tradicionais, computadores com GPUs, *multi-cores*, aceleradores, FPGAs, etc.

Nessa implementação do *workflow* para a construção de redes filogenéticas, obtemos o empacotamento das tarefas através especificação das dependências de dados entre cada uma delas e com o uso de partições distintas, através de executores específicos. Com isso, é gerado um grafo acíclico direcionado (DAG) que permite ao Parsl a escalonar a próxima tarefa disponível para execução imediatamente que fica disponibilizado um recurso computacional.

4. Análises e resultados

4.1. Configuração do Experimento

Sequências de nucleotídeos de genomas completos de Dengue no Brasil foram obtidos do NCBI[5] com a técnica de *web scraping* e as *keywords* COMPLETE GENOME AND DENGUE VIRUS AND TYPE 1 AND BRAZIL. Foram formados dois grupos: 'Total' com 254 genomas (50 de DENV-1, 59 de DENV-2, 87 de DENV-3 e 58 de DENV-4) e 'Curado' com 230 genomas (28 de DENV-1, 57 de DENV-2, 78 de DENV-3 e 57 de DENV-4). O grupo 'Curado' foi derivado do 'Total' no qual foram verificados os metadados de anotação dos genomas como a data de coleta, o nome do genoma e o tipo de hospedeiro (humano).

4.2. Configuração de Ambiente

Todas as ferramentas de bioinformática, Parsl e dependências foram acoplados no supercomputador Santos Dumont[6] (SDumont). Do FLAVi: GeneWise (Wise v2.4.1 e TransDecoder v3.0.0), BLAST v2.4.0, HMMER v3.1b2, EMBOSS v6.6.0 e UniProt (2018). Do PhyloNetworks: RAxML v8.2.12, Astral v5.7.1, SnaQ v0.13.0, MrBayes v3.2.7a, BUCKy versão 1.4.4, Quartet MaxCut v2.10 e Parsl v1.0. As execuções foram realizadas nas unidades Bull Sequana X1120 do SDumont (doravante denominadas *sequana*), com a seguinte configuração: 2x Intel Xeon Cascade Lake Gold 6252, com 48 núcleos (24 por processador) e 384 GB de memória RAM. A parametrização dos *softwares* usados foram fixadas para questões de desenvolvimento do *workflow*.

4.3. Otimização do *workflow*

Para se demonstrar os efeitos potenciais das otimizações feitas, foram realizados dois experimentos de quinze execuções cada um, usando quatro conjuntos de dados sintéticos que são referência para os tutoriais do PhyloNetworks. O primeiro experimento executou os *pipelines* do PhyloNetworks consumindo quatro conjuntos de dados em uma única unidade Sequana. O tempo total de execução obtido foi de 23min 13s (1393s) *i.e.* que sem as otimizações se obtém uma taxa de processamento de $348,24s$, por cada conjunto de dados.

O segundo experimento executou os *pipelines* do PhyloNetworks com os mesmos conjuntos de dados, contudo, se utilizou até quatro unidades Sequana em paralelo (as unidades foram alocadas em *jobs* em partições diferentes de acordo com a necessidade), permitindo que cada tarefa fosse executada nos *cores* disponíveis existentes. Essa

[5]https://www.ncbi.nlm.nih.gov/

[6]https://sdumont.lncc.br/

execução das tarefas foi empacotada e o escalonamento foi feito baseado no DAG que representa a dependência de dados entre as tarefas. Com o empacotamento, o escalonamento do DAG permitiu obter uma taxa de processamento de 274s para todo o conjunto de dados, contra 1393s para a execução sequencial, mostrando um *speedup* potencial cinco vezes em relação ao sequencial.

4.4. Redes Filogenéticas do DENV

A infecção concomitante de duas cepas diferentes (recombinação em DENV) foi demonstrada em humanos, mosquitos e linhagens distintas dentro do mesmo hospedeiro. Primeiras evidências de recombinação entre diversas cepas de dengue surgiram em populações naturais em 1999 e foi mostrada em outros retrovírus e vírus [+] ssRNA e [-] ssRNA [Villabona-Arenas et al. 2013]. Espera-se que eventos reticulados de recombinação de RNA viral em dengue estejam presentes tal que RNAs reorganizam espontaneamente suas sequências ou que, durante a replicação, o RdRp viral interrompa a síntese de RNA e, eventualmente, salte para outro molde a fim de continuar a síntese de RNA.

A Figura 3 apresenta as redes filogenéticas não enraizadas geradas com o PhyloNetworks usando dez genomas de dengue, que foi um número reduzido para efeitos de uma melhor visualização das redes no artigo. As redes apresentam indícios de dois possíveis eventos de hibridização que podem estar relacionados à existência de táxons extintos ou não amostrados. Essas interpretações estão sujeitas a cautela, pois muitos fatores biológicos podem alterar o valor dessas proporções. Cada evento de reticulação deve, portanto, ser interpretado caso a caso. Nestes resultados foram amostrados 10 genomas de dengue. Atualmente o baixo número de genomas totalmente sequenciados e disponíveis de dengue nos bancos de dados é um fator crítico.

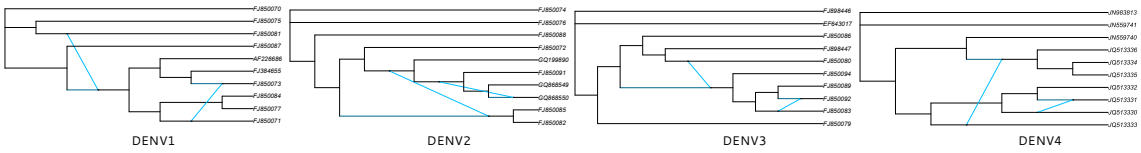

Figura 3. Redes Filogenéticas de Dengue no Brasil geradas com PhyloNetworks.

5. Conclusões e Trabalhos Futuros

Diversos fatores como algoritmos e amostragem afetam a qualidade das árvores e redes filogenéticas. Um estudo comparativo do PhyloNetworks e outras ferramentas similares de análise de redes como o Nexstrain e o StrainHub permite o suporte de estudos e análises em larga escala em genomas de famílias inteiras de Flavívirus. Adicionalmente, o processo de geração das redes é intensivo do ponto de vista computacional e de tempo, além de necessitar de uma grande diversidade de ferramentas. Resultados de desempenho obtidos com o PhyloNetworks acoplado ao Parsl apresentam uma eficiência superior em 27% por conjunto de sequências, com um *speedup* maior que cinco em comparação aos resultados de desempenho obtidos com a versão serial usando os testes sintéticos. A capacidade de gerenciar recursos computacionais e configurações distintos das ferramentas e do ambiente computacional de execução permitirá viabilizar um estudo no nível genômico mais aprofundado, com a exploração de testes em processos evolutivos e eventos reticulados tanto em genomas de dengue como de outros Flavivírus, de diversos estados brasileiros.

6. Agradecimentos

Ao LNCC/MCTI por prover recursos do supercomputador Santos Dumont e às agências de fomento brasileiras CNPq, FAPERJ (JCNE/FAPERJ nº 03/2017 processo 232985) e CAPES (bolsa de mestrado nº 001). Um agradecimento especial ao pesquisador Denis Jacob Machado (University of North Carolina at Charlotte, USA) pela colaboração e discussões científicas nas análises com o FLAVi e análises filogenéticas.

Referências

Babuji, Y. N., Chard, K., Foster, I. T., Katz, D. S., Wilde, M., Woodard, A., and Wozniak, J. M. (2018). Parsl: Scalable parallel scripting in python. In *IWSG*.

Baum, D. et al. (2008). Reading a phylogenetic tree: the meaning of monophyletic groups. *Nature Education*, 1(1):190.

Huson, D. H., Rupp, R., and Scornavacca, C. (2010). *Phylogenetic networks: concepts, algorithms and applications*. Cambridge University Press.

Larget, B. R., Kotha, S. K., Dewey, C. N., and Ané, C. (2010). Bucky: gene tree/species tree reconciliation with bayesian concordance analysis. *Bioinformatics*, 26(22):2910–2911.

OPAS, Organização Pan-Americana da Saúde, W. (2021). Dengue, organização pan-americana da saúde (em português). https://www.paho.org/pt/topicos/dengue. Acessado em 23 de março de 2021.

Ronquist, F. and Huelsenbeck, J. P. (2003). MrBayes 3: Bayesian phylogenetic inference under mixed models. 19(12):1572–1574.

Schneider, A. d. B., Jacob Machado, D., Guirales, S., and Janies, D. A. (2020). Flavi: An enhanced annotator for viral genomes of flaviviridae. *Viruses*, 12(8):892.

Snir, S. and Rao, S. (2012). Quartet MaxCut: A fast algorithm for amalgamating quartet trees. 62(1):1–8.

Solís-Lemus, C. and Ané, C. (2016). Inferring phylogenetic networks with maximum pseudolikelihood under incomplete lineage sorting. *PLoS genetics*, 12(3):e1005896.

Stamatakis, A. (2014). Raxml version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313.

Villabona-Arenas, C. J., de Brito, A. F., and de Andrade Zanotto, P. M. (2013). Genomic mosaicism in two strains of dengue virus type 3. *Infection, Genetics and Evolution*, 18:202–212.

WHO, W. (2021). Dengue and severe dengue. https://www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue. Acessado em 23 de março de 2021.

Wilde, M., Hategan, M., Wozniak, J. M., Clifford, B., Katz, D. S., and Foster, I. (2011). Swift: A language for distributed parallel scripting. *Parallel Computing*, 37(9):633–652.

Zhang, C., Rabiee, M., Sayyari, E., and Mirarab, S. (2018). ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees. 19:153.