Algoritmo Genético para Reconstrução de Árvores Filogenéticas Baseadas em Método de Distância

Hércules Cardoso da Silva[1], Luana Loubet Borges[1], and Wesley Tetsuya Schabert Takiguti Ide[1]

Universidade Estadual de Campinas, Instituto de Computação, Brasil,
l153649@dac.unicamp.br

Abstract. A área de conhecimento de ciências biológicas atua com pesquisas onde é necessário manipular uma grande quantidade de informações, sendo indispensável a utilização de tecnologias avançadas em computação. Em pesquisas que envolvem a reconstrução de árvores filogenéticas, a quantidade de dados manipuláveis é igualmente complexa. Adicionalmente, é comum que algoritmos de reconstrução não gerem a árvore filogenética mais provável. Os pesquisadores que utilizam essas árvores nos seus trabalhos podem chegar a conclusões equivocadas, já que as árvores podem não refletir a evolução dos objetos sendo estudados. Este trabalho visa contribuir com um processo cujo objetivo é utilizar algoritmo genético como heurística para melhorar o algoritmo de reconstrução de árvores utilizando o algoritmo *Neighbor Joining*. A nossa proposta no pior caso é igual ao *Neighbor Joining*, isto é, devolve a mesma árvore, nos casos em que ele não é igual é melhor que o *Neighbor Joining*.

Keywords: heurística, reconstrução filogenética, neighbor joining, algorimto genético.

1 Introdução

Retratar a evolução perfeitamente sempre foi um problema em aberto. Não existe método computacional capaz de gerar a árvore evolutiva tendo total probabilidade de estar correta. Os vários métodos existentes conseguem retornar uma boa solução para solucionar essa questão, no entanto ao combinar mais de um método ou algum método com outra técnica computacional pode ser possível melhorar a saída dos mesmos.

De acordo com [1] a evolução biológica retrata a descendência a partir de modificação de um ancestral comum. Podemos classificar a evolução em dois níveis, pequena e larga escala. A pequena escala refere-se à mudanças que ocorrem na sequência dos genes de um indivíduo de uma geração para a próxima. A evolução em larga escala consiste na descida de espécies diferentes mas que apresentam um ancestral em comum, ao decorrer de muitas gerações.

Conforme [1] e [7] é possível desenhar o processo de evolução contendo ramificações das linhagens durante a evolução, para isso é feita a reconstrução das relações evolutivas através das características de entidades, também chamadas de objetos, que possuem um ancestral comum. Essa reconstrução é chamada de filogenia, que é representada por uma árvore filogenética. Nessa árvore os ramos são interligados por nós e as entidades podem ser "espécies, genes, genomas, ou qualquer outra unidade taxonômica operacional - *Operational Taxonomic Unit* (OTU)", também conhecido como taxon. [3] definem filogenia como "o curso histórico de descendentes de seres orgânicos".

Existem diversos métodos para a reconstrução de árvores filogenéticas. Assim como há vários programas que realizam esta reconstrução com base nestes métodos. A questão é, mesmo quando estão sendo utilizados métodos que inferem a árvore através de probabilidade, os programas e métodos existentes não são capazes de gerar a árvore mais provável, na maioria dos casos também não são capazes de inferir somente uma árvore, dependendo dos dados selecionados como entrada. Os pesquisadores que necessitam da árvores filogenética para realizar a sua pesquisa, muitos deles biológos, podem utilizar uma árvore filogenética pouco provável de estar correta levando-os a uma conclusão equivocada. [10] analisam a reconstrução de uma grande árvore filogenética de *squamatas* (escamados: lagartos e cobras) construída com o intuito de observar a evolução desse grupo. [10] descobriram através da árvore, que nesta reconstrução vivíparo[1] deve ser um estado ancestral de *squamata*. Além disso, encontraram várias transições complexas entre vivíparos e ovíparos[2], tendo mais de 115 origens independentes de vivíparos. Com base na árvore filogenética usada, os autores mostraram o que se achava improvável em questão de evolução, mas isso também depende se essa árvore está refletindo de fato o que ocorre na evolução de *squamatas*.

O problema tratado nesta pesquisa refere-se a como melhorar processos de reconstrução filogenéticas utilizando uma técnica de heurística para realizar a busca por uma solução ótima, devolvendo uma árvore aproximada, já que não é possível encontrar de fato a árvore filogenética ótima. Usamos algoritmo genético para implementar a heurística e a árvore gerada pelo *Neighbor Joining* faz parte da população de entrada do mesmo

2 Revisão da Literatura

A Figura 1, retirada de [5], ilustra um recorte de uma descrição filogenética de três espécies de samambaia e como ela impacta na árvore filogenética reconstruída. A descrição é baseada em um conjunto de propriedades e seus respectivos valores atribuídos a cada espécie na forma de uma matriz. As folhas da árvore filogenética representam as espécies e os demais nós representam hipotéticos ancestrais comuns. Tais ancestrais são inferidos a partir das características comuns entre as espécies decendentes. As espécies *Marattia* e *Zygopteris* compartilham

[1] Quando filhos se desenvolvem dentro do corpo da mãe.
[2] Seres cujo o embrião se desenvolve dentro de um ovo.

a característica *Webbing of the organ*: *broad* com **Nd2**, e as três espécies compartilham uma *leaf*: *branched* com **Nd1**. O grupo de organismos contendo todos os descendentes de um ancestral comum é chamado de clade. Este exemplo foi construído com dados de fenótipo mas poderia ser igualmente construído considerando-se genes e como as espécies compartilham genes.

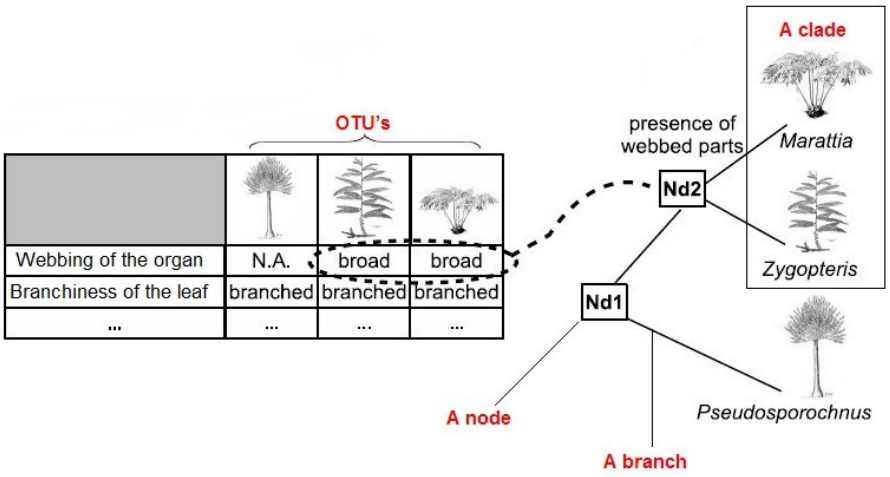

Fig. 1. Exemplo de árvore filogenética [5].

A seguir descreveremos termos importantes para o entedimento da reconstrução de árvores filogenéticas:

- Homologia: diferentes espécies possuem estruturas semelhantes, por compartilharem um ancestral em comum, essas estruturas podem ou não, desempenhar a mesma função.
- Homoplasia: características semelhantes em espécies diferentes adquiridas independentemente sem representar proximidade genética.
  - Paralelismo: em duas espécies diferentes uma mesma condição é alterada de modo idêntico, fazendo com que as espécies fiquem com uma mesma característica.
  - Convergência: duas espécies desenvolvem características similares partindo de duas características primitivas diferentes.
  - Reversão: acontece quando ocorre uma modificação em alguma características, fazendo-a semelhante à uma característica de um ancestral.

De acordo com [3], a Figura 2 ilustra casos de homologia e homoplasia. Na Figura 2 (I) podemos observar uma homologia, pois o caracter x' presente em B e em C é herdado do ancestral BC (ancestral em comum entre B e C). Na Figura

2 (II) temos uma homoplasia, pois o caracter x' presente em A e C não é herdado de um ancestral em comum, e como o caracter x' neste caso foi desenvolvido por duas espécies, temos um caso de convergência. Na Figura 2 III temos outra vez uma homoplasia, pois da mesma forma o caracter x presente em A e C não é herdado de um ancestral em comum, neste caso o caracter x é um ancestral de A e secundariamente é transformado em C, então temos um caso de reversão.

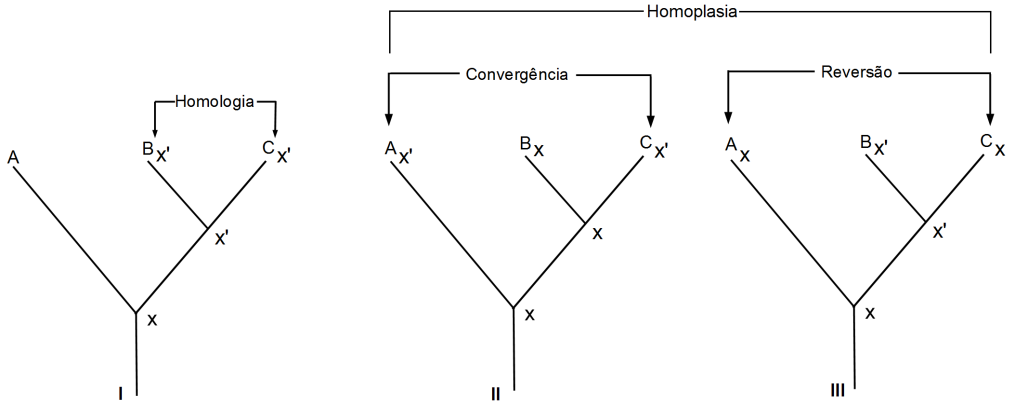

Fig. 2. Exemplo de homologia e homoplasia, retirado de [3].

Árvores filogenéticas consideram que tudo na vida está relacionado e pode ser dividido em três grandes clades, são elas: Achaea, Bacteria e Eukaryota. As árvores filogenéticas são construídas a partir de inferências. Ao realizar esta inferência alguns desafios são enfrentados, são eles: (1) linhagens distintas apresentam diferenças de forma não uniforme; (2) a noção da evolução partindo de um processo com ramificação é na verdade uma simplificação, o que pode levar a interpretações equivocadas; (3) a existência de linhagens evolutivas paralelas, criando linhagens diferentes, mas que parecem advindas do mesmo ancestral. Esse problema é chamado de homoplasia, como explicado acima. Há diferentes métodos para a inferência de árvore filogenética, que tentam tratar esses três problemas (taxas desiguais, evolução paralela, homoplasia), os principais são parcimônia, máxima verossimilhança ou máxima *likelihood* e o método de distância.

Segundo [1] e [7] o processo para gerar uma árvore filogenética a partir do genótipo consiste nos seguintes passos:

1. Escolha de uma sequência de interesse, que será utilizada na geração da árvore;
2. Identificação de dados que são homólogos;
3. Alinhamento da sequência: consiste em organizar sequências de DNA com intuito de identificar regiões de similaridade advindas de relação evolutiva

entre as sequências [9]. A similaridade entre duas sequências de genes pode significar que elas derivam do mesmo ancestral.

4. Gerar a árvore filogenética utilizando algum método, descritos a seguir:

Método de máxima parcimônia: nesse método, várias árvores são geradas e é dada uma pontuação para cada uma delas. Essa pontuação é calculada de acordo com o número de mudanças evolutivas, de forma que a árvore que tem mais mudanças tem uma pontuação maior e a que tem menos mudanças tem uma pontuação menor. Então é escolhida a árvore mais simples, isto é, a que obtiver a menor pontuação [1].

Método de distância: baseia-se no fato que organismos que compartilham um ancestral comum recente devem ser mais semelhantes que organismos que compartilham um ancestral comum mais antigo [1]. A distância entre o taxon A e o taxon B é o "número de diferenças em estados de caractéres" [12]. Sendo assim, é feita uma matriz contendo a distância evolutiva entre as OTUs, esta é chamada de matriz de distância e é utilizada para gerar a árvore.

Máxima *likelihood*: esse método utiliza probabilidade probabilidade para inferir a árvore com mais exatidão. "Máxima *likelihood* calcula a probabilidade para que um conjunto de dados se encaixe em uma árvore derivada desses dados [7]". A análise de máxima *likelihood* é iniciada com uma árvore derivada do conjunto de dados de entrada, então troca-se os ramos gerando outras árvores contendo uma pontuação, essa pontuação é a probabilidade, a árvore escolhida será a que tiver maior pontuação, isto é, a ramificação que tem maior probabilidade de similaridade [1]. Normalmente este método devolve várias aŕvores filogenéticas igualmente prováveis para o mesmo conjunto de dados.

3 Trabalhos Relacionados

Até o momento, não encontramos nenhum método que utiliza algoritmo genético juntamento com o *Neighbor Joining* para melhorar o processo de inferência de árvores filogenéticas, portanto, apresentaremos a seguir alguns trabalhos relevantes que tratam de aspectos parciais da proposta: ferramenta para visualização de árvores filogenéticas e trabalhos que propõem melhorar algoritmos existentes para a reconstrução.

Em 2006 François Chevenet, Christine Brun, Anne-Laure Bañuls, Bernard Jacq e Richard Christen [2] proporam uma ferramenta para a visualização e anotações de árvores filogenéticas, que inclui características como manipulação de árvores usando metainformação através de operadores gráficos ou scripts. O propósito é auxiliar na análise de anotações de uma única árvore ou de uma coleção de árvores. Tal ferramenta é chamada de *TreeDyn*. Essas árvores podem ser importadas de arquivos em formato nexus [8] ou newick[3]. O processo de inserir informações nas árvores é chamado de projeção, que consiste em adicionar textos, imagens ou símbolos nos elementos das árvores, sejam eles sub-árvores,

[3] http://evolution.genetics.washington.edu/phylip/newicktree.html

nós ou folhas. Esta etapa pode ser feita por três métodos: (1) o mais simples é o manual, em que o usuário pode registrar informações obtidas diretamente nos elementos da árvore; (2) um algoritmo anota explicitamente na árvore informações que já estão autocontidas na mesma, tal como o seu comprimento de ramos ou valores *bootstrap*; (3) importa anotações de arquivos externos para adicioná-las às folhas da árvore – um arquivo de anotação é um texto simples contendo um registro por linha, em que cada registro começa com o nome (rótulo) da folha, seguido das suas respectivas anotações. Os dois últimos métodos são feitos automaticamente pela ferramenta. É possível comparar as árvores através do processo chamado de identificação, em que se pode consultar entre os elementos das árvores, rótulos das folhas ou pelos arquivos de anotações.

Em 1997 Olivier Gascuel [4] propôs um método para melhorar o *Neighbor Joining* chamado *BIONJ*. Este algoritmo segue o mesmo esquema de aglomeração do *Neighbor Joining*, usando um modelo que obtém as variâncias e covariâncias das estimativas de distância evolutivas através de alinhamento de sequências. O *BIONJ* tem o resultado melhor que o *Neighbor Joining* quando estas estimativas são obtidas a partir do alinhamento das sequências.

4 Proposta

Em 1987 Naruya Saitou e Masatoshi Nei [11] propuseram um novo método de distância chamado *Neighbor Joining*. O príncipio deste método é unir as OTUs (OTU - *neighbor*) mais próximas de forma a minimizar o comprimento total de cada ramo no agrupamento das OTUs começando com uma árvore estrela [11].

O *Neighbor Joining* utiliza como entrada a matriz de distância de todas as OTUs para todas as OTUs, onde as OTUs são os nós do da árvore e as arestas são as distâncias entre as OTUs. O *Neighbor Joining* segue o princípio da mínima evolução, isto é, ele visa escolher a árvore com a menor soma do comprimento dos *branchs* [4]. Assim o *Neighbor Joining* uni os dois nós mais próximos, e.g., A e B adicionando um novo nó C entre eles, e recalcula a matriz de distância excluindo os nós que foram unidos A e B e incluindo o novo nó C. O processo é feito até unir todos os nós. *Neighbor Joining* gera uma árvore filogenética sem raiz.

Algoritmo genético foi proposto por John Henry Holland em 1975 [6]. Algoritmo genético é uma meta heurística inspirada na evolução para resolver problemas de otimização, podemos descrever o algoritmo genético da seguinte forma: Seja $f(a)$ uma função que avalia a qualidade da solução a, S uma condição de parada, $P(a_1, a_2, ...a_n)$ um conjunto de soluções para um problema:

1. Geramos uma população inicial P de soluções para o problema.
2. Selecionamos indivíduos de P para serem os pais da nova geração com base no valor de f.
3. Fizemos os cruzamentos entre os pais e os indivíduos de P, e geramos um conjunto de indivíduos G, que representa a nova geração.
4. P $\leftarrow$ G.
5. Se nenhuma condição de parada S for atingida então volte para o passo 2.

Nós propomos um método que implementa o algoritmo genético, utilizando como população de entrada várias árvores, entre elas a árvore gerada pelo *Neighbor Joining* e as outras árvores da população são implementadas a partir da matriz de distância unindo, da mesma forma que o *Neighbor Joining* mas ao invés de unirmos os nós mais próximos escolhemos aleatoriamente dois nós para serem unidos, gerando várias árvores de entrada para o algoritmo genético.

5 Metodologia

O objetivo deste trabalho é desenvolver um método para reconstrução de árvores filogenéticas utilizando algoritmo genético e *Neighbor Joining*, a fim de melhorar o processo de reconstrução de árvores filogenéticas utilizando o método de distância com heurística.

5.1 Modelagem do Processo

Esta etapa envolve a modelagem do processo proposto neste trabalho, ilustrada na figura 3 e descrito a seguir.

5.2 Visão Geral do Nossso Método

1. Implementamos o *Neighbor Joining*.
2. Geramos a população de entrada do algoritmo genético, da seguinte forma: (1) geramos várias árvores utilizando a idéia do *Neighbor Joining*, mas ao unirmos dois nós da matriz de distância escolhemos dois nós aleatoriamente para serem unidos por um novo nó; (2) a árvore gerada pelo *Neighbor Joining* no passo anterior faz parte da população.
3. Foram feitos testes para tamanhos de matrizes de entradas diversos, esse tópico será tratado na subseção 5.4.

5.3 Implementação do Protótipo

Nossa implementação do algoritmo genético para o problema de reconstrução de árvores filogenéticas pode ser descrito da seguinte forma: Seja uma matriz de distâncias M entre os objetos $X(x_1, x_2, ..., x_n)$, SM é a soma de todos os elementos da matriz M :

1. Geramos uma população inicial $P(a_1, a_2, ...a_n)$, $a_i(V, E)$ onde V é o conjunto de vértices que compõe a_i e E o conjunto de arestas que compõe a_i. Cada a_i onde $i \neq n$ é uma árvore válida gerada aleatoriamente e a_n é uma arvore gerada pelo *Neighbor Joining*.
2. Ordenamos a população P de maneira não decrescente usando o valor de F para cada elemento de P. Seja $F(a) = |SM - sum(a)|$ para todo a pertencente a P, sum(a) é a soma das distâncias entre todos os nós que representam os objetos do conjunto X na árvore.

3. Seja $P_t(a_1,a_2,...,a_t)$ o conjunto dos t primeiros elementos de P, se $F(y) > F(a_1)$ então $t \leftarrow a_1$.
4. Fizemos o cruzamento de todos os elementos de P_t com os elementos de P.
   - O cruzamento entre a_i e a_j, é feito da seguinte maneira: primeiro construímos o grafo G(A,V) sendo A o conjunto de arestas que contém todas as arestas de a_i e a_j e $V(a_i(V) \cup a_j(V))$.
   - Faça o algoritmo de $Kruskal(G(A,V))$ modificado para escolher as arestas aleatoriamente.
5. Enquanto o número de gerações não for atingido ou $F(y) = 0$ volte ao passo 2. Onde y é a melhor árvore que representa a matriz M.

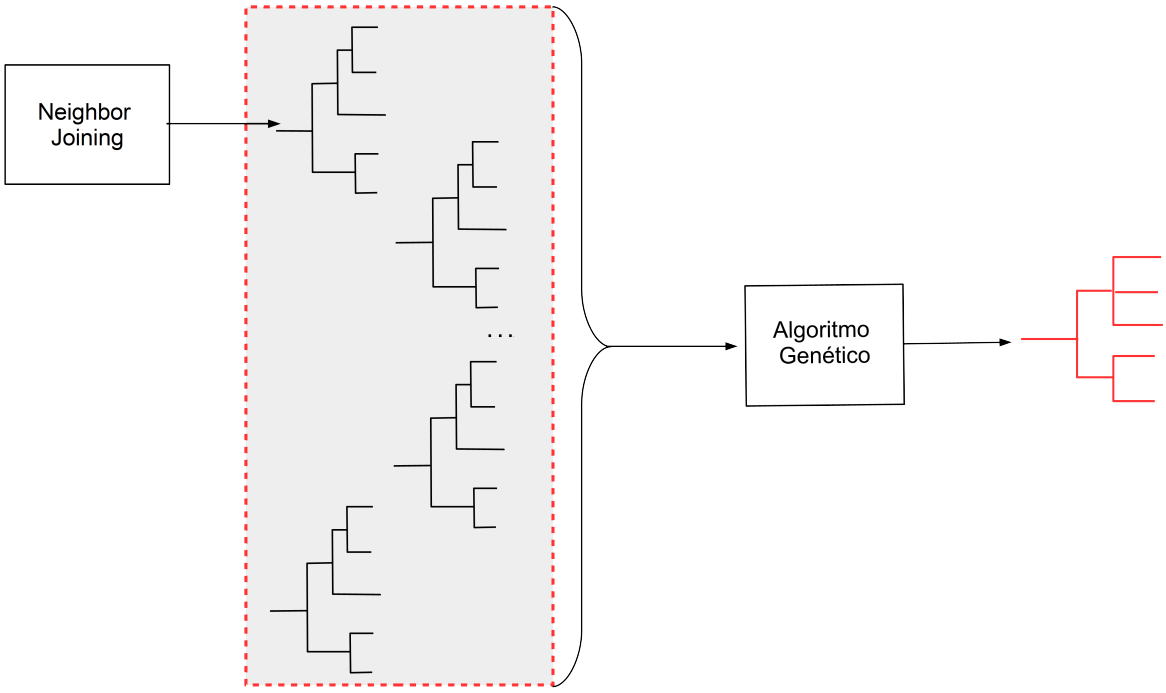

Fig. 3. Metodologia do Processo.

5.4 Validação e Teste

Os testes foram feitos da seguinte forma:

1. A partir da árvore gerada pelo *Neighbor Joining* é construída uma matriz que retrata as distâncias contidas na árvore. Em seguida essa matriz é comparada com a matriz de entrada.

2. É feito o mesmo processo citado no item anterior para a árvore gerada pelo algoritmo genético.
3. Com os passos anteriores obtemos duas matrizes de distâncias, então comparamos as duas para saber qual dos dois métodos foi melhor e o quanto foi melhor.

Como a árvore gerada pelo *Neighbor Joining* faz parte da população de entrada do algoritmo genético, temos que no pior caso o algoritmo genético devolve a mesma árvore do *Neighbor Joining*. Quando a matriz de distância de entrada é aditiva o nosso algoritmo devolve uma árvore aditiva, assim como o *Neighbor Joining*. Na maioria dos testes o nosso algoritmo foi melhor que o *Neighbor Joining*. A seguir os teste mais expressivos são ilustrados através de gráficos na seção 6.

6 Resultados

Fizemos vários testes com entradas de diferentes tamanhos. Para cada instância foram rodados 30 testes. Por exemplo, para um arquivo de entrada com uma matriz de tamanho 10x10, contendo dados hipotéticos, foram realizados 30 testes para entradas diferentes todas de tamanho 10x10. E assim sucessivamente de 10 em 10 até realizar testes de tamanho 100x100. O gráfico 4 representa a saída gerada pelo *Neighbor Joining* e pelo algoritmo genético, sendo rodada os 30 testes para cada tamanho da matriz de distância. No gráfico é comparado o tamanho das intâncias (arquivo de entrada) pela distância da árvore (gerada pelo método) à matriz de distância original. Podemos visualizar no gráfico 4 que, para todas as entradas o algoritmo genético é melhor que o *Neighbor Joining*. Concluimos que o algoritmo genético é em média 32.200% melhor que o *Neighbor Joining*. Essa porcentagem é alta, pois a nossa implementação é muito boa para conjunto de dados pequenos com tamanho até 60x60 da matriz de entrada, depois disso ela se torna em média 2 vezes melhor que o *Neighbor Joining*.

Podemos visualizar claramente os valores na tabela 1, nesta tabela temos:

- Primeira coluna é o tamanho da instância (arquivo de entrada).
- Segunda coluna é a distância média da árvore gerada pelo *Neighbor Joining* à matriz de distância dada como entrada. Essa distância é a média, pois foram testados 30 arquivos de entrada com o tamanho correspondente de cada linha.
- Terceira coluna é a distância média da árvore gerada pelo algoritmo genético (nossa proposta) à matriz de distância dada como entrada. Essa distância é a média, pois foram testados 30 arquivos de entrada com o tamanho correspondente de cada linha.

6.1 Complexidade

A complexidade da nossa implementação é descrita considerando os passos descritos na subseção 5.3:

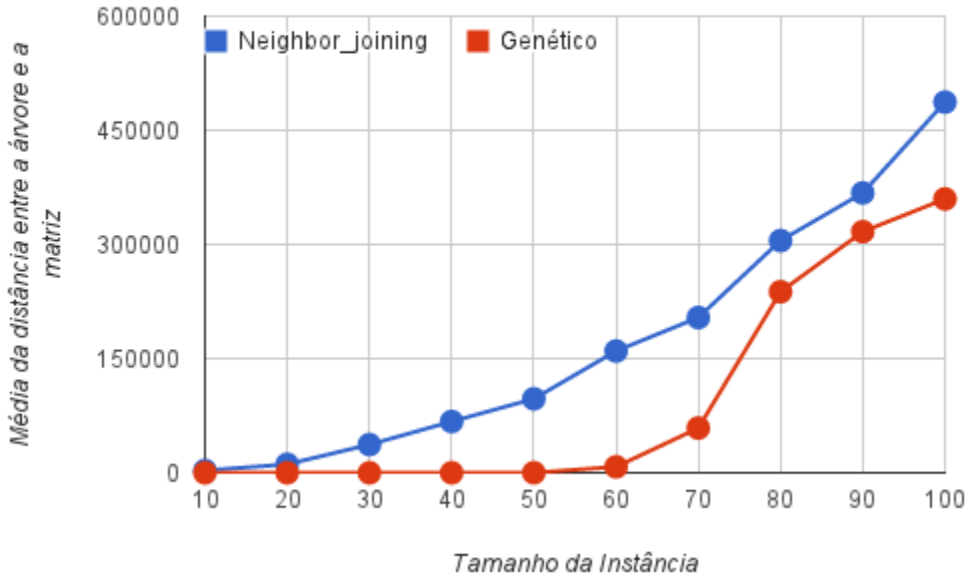

Fig. 4. Comparação do *Neighbor Joining* com o algoritmo genético em relação a matriz de distância de entrada.

1. Gerar a população de entrada: O(n * tamanho da população)
2. Ordenar a população: O(nlogn) + O(n^2) * tamanho da população
3. Escolher os pais: O(t), onde t é o número de pais
4. Crossover: O(n^2 * tamanho da população)

Conforme esses passos, concluímos que a complexidade do nosso algoritmo é:

– complexidade da população inicial + numero de geração * (crossover + encontrar pais + ordenar população). Portanto, a complexidade total do nosso algoritmo é **O(número de gerações * (n^2 * tamanho da população * t))**.

7 Conclusões

Programas de reconstrução filogenética não tem a capacidade de gerar árvores que retratam a evolução de forma perfeita. Contudo, os pesquisadores que utilizam tais árvores necessitam da árvore mais provável possível, já que as conclusões de suas pesquisas dependem das mesmas.

Table 1. Comparação dos métodos

Tamanho das Instâncias	Distância da matriz ao NJ	Distância da matriz ao AG
10	2.968	0
20	11.196	0
30	37.019	0
40	67.358	9
50	97.213	51
60	160.028	7849
70	204.002	58762
80	305.136	237903
90	367.753	316836
100	487.116	359849

Como forma de recontruir árvores filogenéticas utilizando algoritmo genético juntamente com um método de distância, propomos uma abordagem que realiza a junção destes dois conceitos na tentativa de melhorar o máximo possível a inferência de árvores filogenéticas.

A partir desta proposta, implementamos um protótipo que realiza a reconstrução de árvores filogenéticas com base no algoritmo genético e *Neighbor Joining*. Com isto, comparamos qual algoritmo apresenta a melhor solução, o *Neighbor Joining* ou o algoritmo genético. Concluímos que a nossa proposta que implementa o algoritmo genético é no pior caso igual a solução apresentada pelo *Neighbor Joining*. A nossa implementação é muito melhor que o *Neighbor Joining* para instâncias de tamanho pequenas com tamanho até 60x60 da matriz de entrada. Após esses casos ela torna-se em média 2 vezes melhor que o *Neighbor Joining*. Portanto, em 100% dos casos de testes a nossa implementação foi melhor que o *Neighbor Joining* em média 32.200%.

References

1. Barton, N. H., Briggs, D. E., Eisen, J. A., Goldstein, D. B., and Patel, N. H. *Evolution*. Cold Spring harbor Laboratory Press, 2007.
2. Chevenet, F., Brun, C., Bañuls, A.-L., Jacq, B., and Christen, R. Treedyn: towards dynamic graphics and annotations for analyses of trees. *BMC bioinformatics 7*, 1 (2006), 439.
3. Darlu, P., and Tassy, P. La reconstruction phylogénétique. *Concepts et méthodes. Paris, France* (1993).
4. Gascuel, O. Bionj: an improved version of the nj algorithm based on a simple model of sequence data. *Molecular biology and evolution 14*, 7 (1997), 685–695.
5. Grand, A., Lebbe, R. V., and Santanchè, A. From Phenotypes to Trees of Life: A Metamodel-driven approach for integration of Taxonomy models. In *Proc. 10th IEEE e-Science* (Guaruja, 2014).
6. Holland, J. H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975.

7. Jill Harrison, C., and Langdale, J. A. A step by step guide to phylogeny reconstruction. *The Plant Journal 45*, 4 (2006), 561–572.
8. Maddison, D. R., Swofford, D. L., and Maddison, W. P. Nexus: an extensible file format for systematic information. *Systematic Biology 46*, 4 (1997), 590–621.
9. Mount, D. W. Sequence and genome analysis. *Bioinformatics: Cold Spring Harbour Laboratory Press: Cold Spring Harbour 2* (2004).
10. Pyron, R. A., and Burbrink, F. T. Early origin of viviparity and multiple reversions to oviparity in squamate reptiles. *Ecology letters 17*, 1 (2014), 13–21.
11. Saitou, N., and Nei, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution 4*, 4 (1987), 406–425.
12. Stearns, S. C., and Hoekstra, R. F. *Evolução: uma introdução*, vol. 379. Atheneu, 2003.