Reconstrução de Árvores Filogenéticas a partir de mtDNA usando o Algoritmo SEQUITUR

Andresso da Silva, Milena Marinho Arruda e Francisco Marcos de Assis

***Resumo*— A análise filogenética de sequências genômicas agrupa informações sobre a diversidade biológica e classificação genética dos organismos. Essa análise é comumente feita utilizando técnicas de alinhamento de sequências. Contudo, esses métodos têm um custo computacional alto, especialmente à medida que o comprimento das sequências genômicas crescem. Neste artigo é proposto a utilização do algoritmo SEQUITUR para definir a distância entre sequências de mtDNA de dez espécies. Os resultados obtidos a partir do método proposto foram comparados com os gerados baseado em abordagens anteriores que utilizaram o algoritmo de Lempel-Ziv. As árvores filogenéticas geradas foram semelhantes nos dois casos, produzindo os mesmos agrupamentos conforme os grandes grupos analisados. Porém, o tempo médio para a geração das árvores utilizando o SEQUITUR foi da ordem de minutos, enquanto que utilizando o Lempel-Ziv foi de horas. Desta forma, o método proposto possui características preferíveis na aplicação de reconstrução de árvores a partir de sequências de DNA.**

***Palavras-Chave*— SEQUITUR, complexidade, mtDNA, árvore filogenética.**

***Abstract*— Phylogenetic analysis of genomic sequences gathers information about the biological diversity and genetic classification of organisms. This analysis is commonly done using sequence alignment techniques. However, these methods have a high computational cost, especially as the length of genomic sequences grow. In this paper, it is proposed to use the SEQUITUR algorithm to define the distance between mtDNA sequences of ten species. The results obtained from the proposed method were compared with those generated based on previous approaches that used the Lempel-Ziv algorithm. The phylogenetic trees generated were similar in both cases, producing the same clusters according to the major groups analyzed. However, the average time for generating the trees using SEQUITUR was in the order of minutes, while using Lempel-Ziv was hours. Thus, the proposed method has preferable characteristics in the application of tree reconstruction from DNA sequences.**

***Keywords*— SEQUITUR, complexity, mtDNA, phylogenetic tree.**

I. Introdução

A análise filogenética de sequências genômicas é uma área de interesse no processamento de sinais genômicos uma vez que agrupa informações sobre a diversidade biológica e classificação genética e, portanto, explora as relações evolutivas entre os organismos. Essa análise é comumente feita utilizando técnicas de alinhamento de sequências como BLAST [1], USEARCH [2] e KCLUST [3]. Contudo, esses métodos têm um custo computacional alto, especialmente à medida que o comprimento das sequências genômicas crescem. Para superar esse problema, outras abordagens foram adotadas.

Dentre as abordagens está a utilização da complexidade de Lempel-Ziv que foi amplamente investigada como solução devido à sua propriedade de subaditividade [4], [5], [6], [7], [8]. Outros dois métodos que se destacam são o Comet [9], inspirado no algoritmo de compressão por previsão por correspondência parcial (PPM, do inglês: *Prediction by Partial Matching*) e o Kameris [10], baseado na frequência de subsequências de comprimento k.

Outra solução alternativa aos métodos de alinhamento de sequências foi proposta por Li *et al.* [11]. A partir do algoritmo de compressão de sequências *GenCompress1* foi realizada uma estimativa da complexidade de Kolmogorov e essa estimativa foi utilizada para definir as distâncias entre sequências.

Uma solução ainda não explorada para esse problema é utilizar algoritmos baseados na geração gramáticas como SEQUITUR para estimar a complexidade de Kolmogorov. O SEQUITUR é um algoritmo de compressão que identifica repetições de subsequências e que apresenta complexidades temporal e espacial lineares [12], [13] o que é uma característica desejável em aplicações envolvendo sequências DNA que podem possuir um grande número de símbolos.

Desta forma, nesse artigo é proposto o uso do algoritmo SEQUITUR para estimar a complexidade de sequências de DNA bem como a distância entre elas. Em seguida, as distâncias são aplicadas na construção de árvores filogenéticas cuja base de dados é composta por sequências genéticas mitocondriais (mtDNA) de dez espécies de mamíferos placentários. Os resultados obtidos neste artigo são comparados com as árvores filogenéticas geradas utilizando o algoritmo de Lempel-Ziv.

As Seções seguintes estão organizadas como se segue. Na Seção II são apresentados os conceitos fundamentais, definições e a descrição do algoritmo SEQUITUR. Na Seção III são apresentados os detalhes da metodologia incluindo as informações da base de dados utilizada. Na Seção IV são discutidos os resultados obtidos e, por fim, na Seção V são apresentadas as conclusões.

Andresso da Silva, Departamento de Engenharia Elétrica, UFCG, Campina Grande-PB, e-mail: andresso.silva@ee.ufcg.edu.br; Milena Marinho Arruda, Departamento de Engenharia Elétrica, UFCG, Campina Grande-PB, e-mail: milena.arruda@ee.ufcg.edu.br; Francisco Marcos de Assis, Departamento de Engenharia Elétrica, UFCG, Campina Grande-PB, e-mail: fmarcos@dee.ufcg.edu.br; Esse artigo foi parcialmente financiado pelo CNPq.

II. Conceitos Fundamentais

A. Gramática Livres de Contexto e Complexidade

Definição 1 (Gramática Livre de Contexto (GLC) [14]): Uma gramática livre de contexto é uma quádrupla $G = (N, T, P, S)$ em que

- N é o conjunto dos símbolos não-terminais;
- T é o conjunto dos símbolos terminais;
- P é o conjunto de regras de produção da forma $\sigma \rightarrow p$, em que $\sigma \in N$ e $p \in (N \cup T)^*$;
- S é a expressão inicial em que as regras de produção serão aplicadas.

Definição 2 (Complexidade de uma regra): A complexidade de uma regra $\sigma \rightarrow p$ é definida como

$$K(\sigma \rightarrow p) = l(p), \tag{1}$$

em que $l(p)$ é o número de símbolos em p.

Definição 3 (Complexidade de uma Sequência): A complexidade de uma GLC gerada a partir de uma sequência p é definida como

$$K_G(p) = l(S) + \sum_{\sigma \in N} K(\sigma \rightarrow q), \tag{2}$$

em que $l(S)$ é o número de símbolos em S.

B. SEQUITUR

O SEQUITUR [12] é um algoritmo de compressão que apresenta complexidades temporal e espacial lineares [13] e que realiza inferência da estrutura hierárquica de sequências a partir de repetições de subsequências. As repetições encontradas são utilizadas para definir uma gramática em que toda regra deve ser utilizada pelo menos duas vezes e todo par de símbolos terminais ou não-terminais consecutivos que aparecem mais de uma vez devem se tornar regras.

Um par de símbolos terminais ou não-terminais consecutivos é chamado de diagrama. Em [15] o SEQUITUR foi aplicado para a compressão de textos semi-estruturados e apresentou compressões maiores do que os outros métodos utilizados para esse fim, incluindo o PPM. O pseudo-código do SEQUITUR é apresentado no Algoritmo 1.

Na Tabela I é apresentado um exemplo de como é o processo de geração da gramática do SEQUITUR para a sequência $u = ABCABC$. Os símbolos são lidos da esquerda para a direita e a cada novo símbolo lido um diagrama é formado e armazenado. No momento em que dois diagramas iguais são observados, uma regra é formada e os diagramas são substituídos pelo símbolo não-terminal que representa a regra. Quando uma regra é utilizada somente uma vez, o símbolo não-terminal que representa a regra é substituído pelos símbolos a que ela se refere.

Desta forma, o SEQUITUR retornará (22, {2→ABC}) considerando o exemplo da Tabela I. Para obter a sequência original, basta fazer o processo de substituições sucessivas das regras. Para calcular a complexidade da gramática gerada pelo SEQUITUR se utiliza a Eq. 2, ou seja, $K_G(u) = K(22) + K(2 \rightarrow ABC) = 5$.

Considerando como outro exemplo a sequência $v = ATGGTGCACCTGACTCCTGAGG$, a gramática gerada pelo SEQUITUR é (A1G1C232TC3AGG, {1→TG, 2→AC, 3→C1}). A expressão A1G1C232TC3AGG possui 14 símbolos, dos quais são 6 não-terminais e 8 são terminais. Cada uma das 3 regras que formam as regras possui uma complexidade de 2, então, segue que a complexidade de v é $K_G(v) = 14 + 2 + 2 + 2 = 20$.

Algoritmo 1: Pseudo-código do SEQUITUR [13].

```
Entrada: Sequência u
Saída: Gramática de (S, P) gerada a partir de u
S ← string vazia;
para Cada símbolo i de u faça
    Anexar i à regra S;
    se Aparecer um digrama duplicado então
        se A outra ocorrência é uma regra então
            Substituir o novo digrama pelo não-terminal
             que encabeça o outro digrama;
        senão
            Formar uma nova regra com um novo
             não-terminal;
            Substituir os dois digramas pelo novo
             não-terminal;
        fim
        se Uma regra é utilizada apenas uma vez então
            Substituir o conteúdo da regra pelo
             não-terminal;
            Remover a regra;
        fim
    fim
fim
```

TABELA I: Geração da gramática utilizando o SEQUITUR para a sequência $u = ABCABC$.

	S	Regras	Diagramas	Comentários
A	A			Nenhuma ação.
B	AB		AB	Novo digrama: AB.
C	ABC		AB,BC	Novo digrama: BC.
A	ABCA		AB,BC,CA	Novo digrama: CA.
B	ABCAB		AB,BC,CA	Digrama AB já visto, cria regra.
	1C1	1→AB	AB	Regra criada para representar AB.
	1C1	1→AB	AB,1C,C1	Novos digramas: 1C e C1.
C	1C1C	1→AB	AB,1C,C1	Digrama 1C já visto, cria regra
	22	1→AB,2→1C	AB,1C	Regra criada para representar 1C.
	22	2→ABC	AB,BC	Regra 1 somente usada uma vez, então é removida.
	22	2→ABC	AB,BC,22	Novo digrama: 22.

O SEQUITUR produz gramáticas com complexidade menor ou igual ao comprimento da sequência de entrada. Desta forma, pode-se calcular a redução no uso de símbolos de uma sequência u de comprimento n como

$$R(u) = 1 - \frac{K_G(u)}{n} \tag{3}$$

de forma que, para $K_G(u) = 1$, a redução será de 0, sendo necessário exatamente n símbolos para reproduzir a sequência u.

Neste artigo foi utilizada a implementação do SEQUITUR disponível na biblioteca scikit-sequitur [16] escrita na linguagem Python.

III. MÉTODOS

A abordagem adotada neste artigo para gerar as árvores filogenéticas a partir do SEQUITUR se utiliza do conceito de complexidade das gramáticas livres de contexto como estimativa para a complexidade de Kolmogorov. Desta forma, a complexidade da gramática gerada pelo SEQUITUR para uma determinada sequência será utilizada como estimativa da sua complexidade Kolmogorov. Como a complexidade de Kolmogorov não é computável, então se utiliza algoritmos para se obter estimativas [17], [11].

A partir da estimativa da complexidade de Kolmogorov das sequências é possível definir uma distância entre estas. Nesse sentido, Chen e colaboradores [17] propuseram utilizar como medida de distância entre duas sequências u e v,

$$D(u,v) = 1 - \frac{K(v) - K(v|u)}{K(uv)} \tag{4}$$

em que $K(\cdot)$ é a complexidade de Kolmogorov. Posteriormente, Li *et al.* [11] apresentaram a mesma medida de distância da Eq. (4) e provaram a desigualdade triangular para pequenos valores de complexidade.

Para utilizar a distância da Eq. (4) também para o algoritmo SEQUITUR, define-se a complexidade condicional entre sequências a partir das gramáticas geradas conforme a definição a seguir.

Definição 4 (Complexidade condicional): Sejam duas sequências u, v, a complexidade condicional $K_G(u|v)$ de u dado v é definido como a complexidade $K_G(uv)$ da gramática obtida pela concatenação de u e v menos a complexidade $K_G(v)$ de v, ou ainda

$$K_G(u|v) = K_G(vu) - K_G(v). \tag{5}$$

Desta forma, a complexidade condicional será menor se as sequências forem parecidas e maior, caso contrário. Isso acontece porque se as sequências forem parecidas, as regras geradas para u serão mais reutilizadas em u, fazendo com que a gramática cresça com uma menor taxa do que se regras diferentes fossem criadas e adicionadas.

O índice G da complexidade da gramática gerada pelo SEQUITUR será suprimido por simplicidade da notação, quando não houver confusão entre as complexidades. Variações na distância da Eq. (4) foram experimentadas neste artigo (*e.g.*, usar $|u|+|v|$ ou $K(u)+K(v)$ como denominador), porém, foram obtidos resultados semelhantes ou agrupamentos inesperados. Desta forma, optou-se por usar a Eq. (4) como medida de distância.

Para comparar as árvores filogenéticas obtidas com a utilização do SEQUITUR, foi também implementada a abordagem apresentada por [18]. Em [18], é utilizada a complexidade $c(\cdot)$ de Lempel-Ziv [19] e a complexidade condicional $c(\cdot|\cdot)$ de sequências para definir a medida de distância entre duas sequências u e v como

$$D(u,v) = \max\{c(u|v), c(v|u)\} \tag{6}$$

que é simétrica, não-negativa e obedece a desigualdade triangular.

A complexidade $c(v)$ é definida como o número de frases em que a sequência v pode ser separada de forma que cada nova frase seja composta pela maior subsequência observada até o momento na sequência e um novo símbolo [20]. Como é feita a verificação de todas as subsequências anteriores, a complexidade temporal desse algoritmo é bem maior do que a do SEQUITUR. A complexidade $c(u|v)$ é o número de frases em que a sequência u pode ser separada considerando que as maiores subsequências podem pertencer a v. Desta forma, quanto maior a semelhança entre as sequências u e v, menor será a complexidade condicional $c(u|v)$. O algoritmo de Lempel-Ziv apresentado em [19] é conhecido como LZ76 e será assim referido doravante.

As sequências de DNA analisadas neste artigo estão disponíveis no banco de dados de nucleotídeos do National Center for Biotechnology Information (NCBI) que fornece acesso aberto a informações biomédicas e genômicas [21]. Os dados genômicos disponibilizados pelo NCBI são rotuladas por um identificador, o número GI, que é uma série simples de dígitos processados pelo NCBI. Portanto, a análise da reconstrução da árvore filogenética de dez espécies de mamíferos placentários dar-se-à por meio das seguintes sequências genômicas mitocondriais: X83427, Z29573, V00662, D38116, Y18001, V00711, X14848, V00654, X79547, U96639. Essa sequências foram selecionadas para contemplar primatas, roedores, espécies do clado ferungulata e um grupo externo a fim de verificar o agrupamento gerado a partir do SEQUITUR e do LZ76.

Foram calculadas as distâncias entre as sequências genômicas mitocondriais selecionadas de forma a gerar uma matriz de distância. Devido ao fato de que $K(v) - K(v|u) \approx K(u) - K(u|v)$ [22], então a matriz de distância gerada foi aproximadamente simétrica. Nos trabalhos que utilizaram essa distância [17], [11] não foram sugeridos modos de lidar com a não simetria da matriz de distância. Desta forma, foram experimentadas estratégias como usar o valor mínimo $\min\{K(v) - K(v|u), K(u) - K(u|v)\}$, o valor máximo $\max\{K(v) - K(v|u), K(u) - K(u|v)\}$ e a média aritmética $(K(v) - K(v|u) + K(u) - K(u|v))/2$ para tornar a matriz simétrica. A utilização da média aritmética apresentou os melhores resultados em termos de agrupamento e, portanto, foi a escolhida para gerar os resultados.

As matrizes de distância foram utilizadas como entrada para o método de grupo de pares não ponderados com média aritmética (UPGMA) que é um método de clusterização hierárquico aglomerativo usado para gerar dendrogramas, representando as árvores filogenéticas [23].

IV. RESULTADOS E DISCUSSÃO

Para gerar os resultados, foi utilizado um computador com um processador Intel i7, memória RAM 8 GB e relógio de 2,7 GHz.

Na Tabela II são apresentadas as complexidades calculadas com o SEQUITUR para as sequências já mencionadas. Nota-se que as complexidades K calculadas são menores do que o comprimento n das sequências e que o SEQUITUR produz uma gramática que reproduz a sequência original utilizando até 70% menos símbolos. O cálculo das reduções (R) foram realizados por meio da Eq. (3).

TABELA II: Comprimento n das sequências e complexidades K das sequências selecionadas.

Grupo	Espécie	Sequência	n	K	R
Primatas	Chimpanzé	D38116	16554	4969	0.700
	Babuíno	U20753	16521	4983	0.698
	Humano	D38115	16569	4974	0.700
Ferungulata	Cachorro	U96639	16727	4990	0.702
	Cavalo	X79547	16660	4942	0.703
	Boi	X99256	16338	4954	0.697
Roedores	Rato	Y18001	16300	4885	0.700
	Camundongo	Y10524	16295	4842	0.703
Grupo Externo	Opossum	X14848	17084	4994	0.708
	Ornitorrinco	AJ001562	17019	5043	0.704

Para analisar o comportamento da complexidade condicional do SEQUITUR, foram selecionadas sequências correspondentes a espécies evolutivamente próximas (humanos e chimpanzés), evolutivamente distantes (humanos e cachorros) e sequências aleatórias. Foram utilizadas sequências de comprimento igual a 4000 símbolos devido ao tempo para gerar as sequências aleatórias. Na Fig. 1, são apresentadas as complexidades para as sequências do humano concatenadas da seguinte maneira: humano com humano (*humano+humano*), humano com chimpanzé (*humano+chimp.*), humano com cachorro (*humano+cão*), humano com sequências aleatórias (*humano+aleat.*).

A complexidade para os primeiros 4000 símbolos é igual à complexidade do humano. Como o esperado, a menor complexidade da concatenação é obtida para o caso de humano com humano, seguido da complexidade do humano com o chimpanzé, do humano com o cachorro e, por fim, humano com uma sequência aleatória. Desta forma, a distância obtida será proporcional à diferença entre a complexidade da concatenação e a complexidade do humano e a distância entre as sequências aparenta refletir a distância evolutiva entre as espécies.

A curva correspondente a *humano+aleat.* apresenta um sombreado porque foram geradas 700 sequências aleatórias e foi tomado a média aritmética entre as complexidades. O coeficiente angular da curva da complexidade considerando a sequência aleatória é a aproximadamente igual ao coeficiente angular da curva da complexidade do humano. Isso indica que foi necessário criar novas regras e a complexidade continuou a crescer com a mesma taxa.

Quando utilizadas sequências com menos de 1000 símbolos, observou-se que as diferenças entre as complexidades das concatenações não eram bem definidas, podendo levar a cálculos incorretos de distância (*e.g.*, cachorro ser mais próximo evolutivamente do humano do que o chimpanzé do humano). O mesmo ocorreu quando se utilizou o SEQUITUR nas sequências apresentadas em [7] as quais possuem menos de 100 símbolos e a árvore filogenética produzida não apresentou os agrupamentos esperados. Desta forma, quanto maiores as sequências, melhor discerníveis são as diferenças entre as sequências. Assim, é esperado que a árvore gerada reflita mais adequadamente as distâncias evolutivas entre as

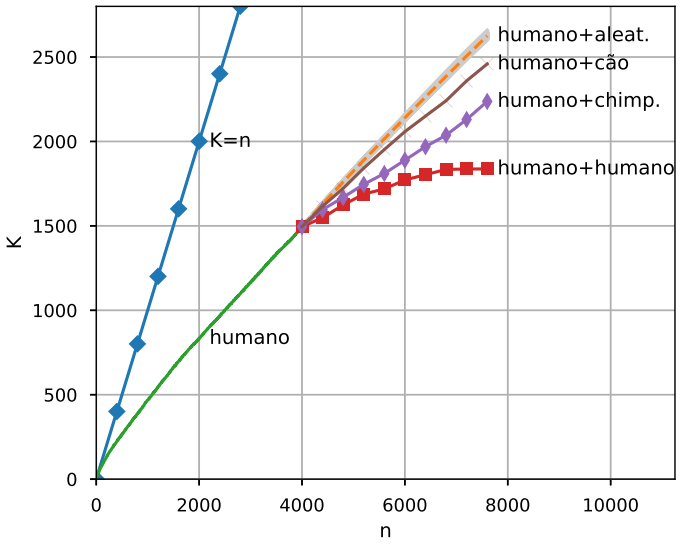
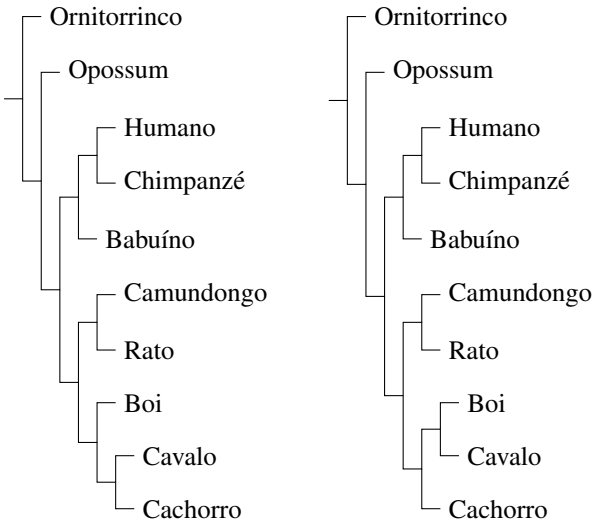


Fig. 1: Complexidades para as sequências do humano e concatenadas com aleatório, cão, chimpanzé e humano.



(a) SEQUITUR. (b) LZ76 [18].

Fig. 2: Árvores geradas utilizando SEQUITUR e LZ76.

espécies.

Utilizando as sequências selecionadas (ver Tabela II) foram geradas as árvores filogenéticas apresentadas na Fig. 2, usando o SEQUITUR (Fig. 2a) e usando o LZ76 (Fig. 2b) baseado em [18].

As espécies de primatas, roedores, ferungulata e grupo externo foram associadas corretamente nas árvores de acordo com seus grupos. Desta forma, as árvores obtidas utilizando tanto o SEQUITUR quanto o LZ76 são da forma como representado na Fig. 3. A única diferença entre as árvores geradas utilizando SEQUITUR e LZ76 é o agrupamento para o Ferungulata, em que o o cachorro e o cavalo são apresentados como mais próximos evolutivamente. Isso pode ter acontecido devido a semelhança entre as sequências do cavalo e cachorro, sendo necessárias sequências maiores para haver uma distinção maior entre as complexidades estimadas e entre as distâncias.

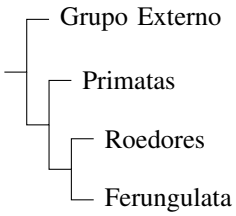


Fig. 3: Agrupamento gerado utilizando SEQUITUR e LZ76.

O tempo médio para a geração da árvore da Fig. 2a utilizando o SEQUITUR foi de 56 segundos. Enquanto que o tempo médio para a geração da árvore da Fig. 2b utilizando o LZ76 foi de 4225 segundos, o que é aproximadamente 1 hora e 10 minutos. Já ao considerar 29 espécies com comprimentos semelhantes às já utilizadas, o tempo estimado para gerar as árvores utilizando o SEQUITUR e LZ76 passa a ser 7 minutos e 8 horas, respectivamente. Como o SEQUITUR tem complexidade linear no tempo, era esperado que o tempo de execução fosse menor, o que torna possível sua utilização para um maior número de sequências e sequências com maiores comprimentos, enquanto que, sob as mesmas condições, o tempo requerido pelo LZ76 o torna inviável.

V. Conclusões

Neste artigo foi proposto a utilização do algoritmo SEQUITUR para definir a distância entre sequências de DNA mitocondrial (mtDNA) de dez espécies contemplando roedores (rato e camundongo), primatas (chimpanzé, babuíno, humano), ferungulata (cachorro, cavalo, boi) e um grupo externo (ornitorrinco e opossum).

Os resultados obtidos utilizando o SEQUITUR foram comparados com os gerados baseado em abordagens anteriores utilizando o algoritmo de Lempel-Ziv. As árvores filogenéticas geradas foram semelhantes nos dois casos, produzindo os mesmos agrupamentos conforme os grandes grupos. Porém, o tempo médio para a geração das árvores utilizando o SEQUITUR foi da ordem de minutos, enquanto que utilizando o Lempel-Ziv foi de horas.

Observou-se que a geração das árvores utilizando o SEQUITUR é mais sensível do que o Lempel-Ziv, principalmente levando em consideração o comprimento das sequências. Para a base de dados e abordagem adotadas, é preferível comparar sequências de mtDNA com mais de 4000 símbolos, para reduzir a sensibilidade ao comprimento das sequências.

Sendo assim, a aplicação do SEQUITUR para reconstrução de árvores filogenéticas produziu resultados semelhantes aos já encontrados na literatura. Além disso, o fato de as complexidades temporal e espacial do SEQUITUR serem lineares e não utilizar alinhamentos de sequências, fazem o método proposto possuir características preferíveis na aplicação de reconstrução de árvores a partir de sequências de DNA.

Como trabalhos futuros, pretende-se testar o desempenho do método proposto considerando bases de dados maiores, contendo sequências maiores. Além disso, propõe-se comparar o método proposto com mais métodos de geração de árvores filogenéticas.

Referências

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of molecular biology*, vol. 215, no. 3, pp. 403–410, 1990.

[2] R. C. Edgar, "Search and clustering orders of magnitude faster than blast," *Bioinformatics*, vol. 26, no. 19, pp. 2460–2461, 2010.

[3] M. Hauser, C. E. Mayer, and J. Söding, "kclust: fast and sensitive clustering of large protein sequence databases," *BMC bioinformatics*, vol. 14, no. 1, p. 248, 2013.

[4] H. H. Otu and K. Sayood, "A new sequence distance measure for phylogenetic tree construction," *Bioinformatics*, vol. 19, no. 16, pp. 2122–2130, 2003.

[5] M. Li, X. Chen, X. Li, B. Ma, and P. M. Vitányi, "The similarity metric," *IEEE transactions on Information Theory*, vol. 50, no. 12, pp. 3250–3264, 2004.

[6] A. Kraskov, H. Stögbauer, R. G. Andrzejak, and P. Grassberger, "Hierarchical clustering using mutual information," *EPL (Europhysics Letters)*, vol. 70, no. 2, p. 278, 2005.

[7] N. Liu and T.-m. Wang, "A relative similarity measure for the similarity analysis of dna sequences," *Chemical Physics Letters*, vol. 408, no. 4-6, pp. 307–311, 2005.

[8] L. Liu, D. Li, and F. Bai, "A relative lempel–ziv complexity: Application to comparing biological sequences," *Chemical Physics Letters*, vol. 530, pp. 107–112, 2012.

[9] D. Struck, G. Lawyer, A.-M. Ternes, J.-C. Schmit, and D. P. Bercoff, "Comet: adaptive context-based modeling for ultrafast hiv-1 subtype identification," *Nucleic acids research*, vol. 42, no. 18, pp. e144–e144, 2014.

[10] S. Solis-Reyes, M. Avino, A. Poon, and L. Kari, "An open-source k-mer based machine learning tool for fast and accurate subtyping of hiv-1 genomes," *PLoS One*, vol. 13, no. 11, p. e0206409, 2018.

[11] M. Li, J. H. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang, "An information-based sequence distance and its application to whole mitochondrial genome phylogeny," *Bioinformatics*, vol. 17, no. 2, pp. 149–154, 2001.

[12] C. Nevill-Manning, I. Witten, and D. Maulsby, "Compression by induction of hierarchical grammars," in *Proceedings of IEEE Data Compression Conference (DCC'94)*, pp. 244–253, 1994.

[13] C. G. Nevill-Manning and I. H. Witten, "Identifying hierarchical structure in sequences: A linear-time algorithm," vol. 7, p. 67–82, Sept. 1997.

[14] W. Ebeling and M. A. Jiménez-Montaño, "On grammars, complexity, and information measures of biological macromolecules," *Mathematical Biosciences*, vol. 52, no. 1, pp. 53–71, 1980.

[15] C. Nevill-Manning, I. Witten, and D. Olsen, "Compressing semi-structured text using hierarchical phrase identifications," in *Proceedings of Data Compression Conference - DCC '96*, pp. 63–72, 1996.

[16] G. Jenks, "SciKit Sequitur." https://pypi.org/project/scikit-sequitur/, 2021. [Online; Acessado 20-Fevereiro-2021].

[17] X. Chen, S. Kwong, and M. Li, "A compression algorithm for dna sequences and its applications in genome comparison," in *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology*, RECOMB '00, (New York, NY, USA), p. 107, Association for Computing Machinery, 2000.

[18] B. Li, Y.-B. Li, and H.-B. He, "Lz complexity distance of dna sequences and its application in phylogenetic tree reconstruction," *Genomics, proteomics & bioinformatics*, vol. 3, no. 4, pp. 206–212, 2005.

[19] A. Lempel and J. Ziv, "On the complexity of finite sequences," *IEEE Transactions on information theory*, vol. 22, no. 1, pp. 75–81, 1976.

[20] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[21] NCBI, "Nucleotide[internet].." Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information. 1988.

[22] M. Li and P. M. Vitnyi, *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Publishing Company, Incorporated, 3 ed., 2008.

[23] M. Weiß and M. Göker, "Chapter 12 - molecular phylogenetic reconstruction," in *The Yeasts (Fifth Edition)* (C. P. Kurtzman, J. W. Fell, and T. Boekhout, eds.), pp. 159–174, London: Elsevier, fifth edition ed., 2011.