The Scientific World Journal
Volume 2012, Article ID 450124, 4 pages
doi:10.1100/2012/450124

The ScientificWorldJOURNAL

Research Article

Sequence Comparison Alignment-Free Approach Based on Suffix Tree and *L-Words* Frequency

Inês Soares,[1,2,3] Ana Goios,[2] and António Amorim[1,2]

[1] *Faculdade de Ciências da Universidade do Porto, 4169 Porto, Portugal*
[2] *Instituto de Patologia e Imunologia Molecular da Universidade do Porto, 4200 Porto, Portugal*
[3] *Centro de Matemática da Universidade do Porto, 4169 Porto, Portugal*

Correspondence should be addressed to Inês Soares, isoares@ipatimup.pt

Received 15 June 2012; Accepted 5 August 2012

Academic Editors: J.-C. Aude, Y. Muto, and T. Roegnvaldsson

The vast majority of methods available for sequence comparison rely on a first sequence alignment step, which requires a number of assumptions on evolutionary history and is sometimes very difficult or impossible to perform due to the abundance of gaps (insertions/deletions). In such cases, an alternative alignment-free method would prove valuable. Our method starts by a computation of a generalized suffix tree of all sequences, which is completed in linear time. Using this tree, the frequency of all possible words with a preset length *L*—*L-words*—in each sequence is rapidly calculated. Based on the *L-words* frequency profile of each sequence, a pairwise standard Euclidean distance is then computed producing a symmetric genetic distance matrix, which can be used to generate a neighbor joining dendrogram or a multidimensional scaling graph. We present an improvement to word counting alignment-free approaches for sequence comparison, by determining a single optimal word length and combining suffix tree structures to the word counting tasks. Our approach is, thus, a fast and simple application that proved to be efficient and powerful when applied to mitochondrial genomes. The algorithm was implemented in Python language and is freely available on the web.

1. Introduction

During the last decades many sequence comparison methods have been developed in order to recover evolutionary and phylogenetic signals as well as for the discovery of pathogenic mutations [1, 2].

The most common approaches are based on sequence alignments [3, 4]. However, alignment quality depends on the penalties attributed to observed differences between sequences during the alignment process [5, 6]. Alternatively, many alignment-free methods have also been proposed [5, 7–9] which, being based on word frequencies or on match lengths, are algorithmically simple and computationally faster than alignment methods.

The basis of word frequency tasks is the determination of the optimal word length, L, which should be computed a priori. The *L-words* counting in a sequence is usually performed considering a one base sliding window, overlapping $L-1$ consecutive bases, that is, shifting one base each time until $m-L+1$, m being the sequence length [7, 8].

Here, we present a new approach that determines a single optimal word length, L, and generates *L-words* frequency profiles using suffix tree theory. The algorithm was applied to a variety of mtDNA sequences that are particularly difficult to handle by automated alignment methods and the performance was compared to the available word counting alignment-free methodologies.

2. Methods

2.1. Algorithm. We present here a new algorithm representing an improvement of word counting alignment free methodologies. The algorithm is described in Supplementary Material available online at doi:10.1100/2012/450124 and each step is summarized below.

2.1.1. Suffix Tree Approach. The first step of the method is the construction of a generalized suffix tree, T, of n sequences, $S_1, S_2, \ldots, S_n$, where every suffix in the data set is represented only once. Therefore, the memory requirements when using these structures are much more modest than when considering the original complete sequences. The construction of a generalized suffix tree is based on Ukkonen's algorithm, described with detail by Gusfield [10]. Function *GST* in the Supplementary Algorithm 1 automates the construction of this structure.

Generalized suffix trees are potent structures, having the useful property that each prefix of paths leading from the root to any internal node points to all occurrences of this prefix in the data set [10]. Thus, when aiming to determine the number of times that a word w occurs in each sequence, we only need to traverse the generalized suffix tree leading from the root in the direction of the branch labeled by a prefix of $w - w[1, \ldots, j], 1 \leq j \leq L$. If such branch does not exist, we conclude that w does not occur in the data set. Otherwise, we must always skip from a node to its descendant until the end of w. The indexes of all descendant leafs from the last node reached, or from its descendant nodes, are used to determine the sequences in the data set which contain w as well as the number of occurrences of w in each sequence. Each leaf indexes the sequences and the corresponding starting positions of the associated suffixes labeled in the path that leads from the root to this leaf.

An alternative approach, using a k-truncated suffix tree deserves consideration, due to reduction in both memory requirements and running time [11].

2.1.2. L-Words Frequencies. In the next step, we determine all words in the DNA alphabet {A, C, G, T} with length L—W_L—determined a priori, following the method of Sims et al. [7]. According to these authors, there is an optimal resolution range in which any integer value should be considered as the length of L. Any value inside this interval is equally good. So, in order to increase the speed of the process we start by considering only the lower limit of resolution, which is given by the expression $\log_4(m)$, where m is the sequence length. Considering n sequences with different lengths m, the expression $\log_4(m)$ can obviously generate different values. In order to find a value applicable to all sequences under analysis, we choose m as the length of the greater sequence and L as the smaller integer greater than $\log_4(m)$. Thus, in the present study, we work with the following values:

$$m = \max\{\text{length}(S_i), 1 \leq i \leq n\}, \tag{1}$$

$$L = \left\lceil \log_4(m) \right\rceil, \tag{2}$$

where $\lceil x \rceil$ is the *ceiling function of x*, defined as the smallest integer is not less than x.

Notice that the total number of possible *L-words* is $t = 4^L$ and $W_L = [w_{L_1}, w_{L_2}, w_{L_3}, \ldots, w_{L_t}]$. For example, if $L = 2$ then $t = 16$ and the following result is obtained:

$$W_2 = [AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG, GT, TA, TC, TG, TT]. \tag{3}$$

Using the generalized suffix tree we can efficiently determine the number of occurrences of each $w_j \in W_L$ in each sequence S_i just by traversing the branch with path label w_j from the root towards the leafs only one time, as was thoroughly explained in the previous section: $O_{ij} = \#\{w_j \text{ in } S_i\}$.

Finally, we can determine the relative frequency of each word w_j in each sequence $S_i - f_{ij}$ as the following:

$$f_{ij} = \frac{O_{ij}}{\sum_{j=1}^{t} O_{ij}} \in [0, 1]. \tag{4}$$

The resulting matrix F_L with dimension $n \times t$ and entries f_{ij} represents a global profile of *L-words* frequencies of all input sequences. The determination of each element f_{ij} is automated with function *LwF* in the Supplementary Algorithm 1.

2.1.3. Genetic Distance. The generated frequencies matrix may then be used to assign a pairwise correlation or a metric distance between each pair or sequences. In this work we calculate the pairwise standard Euclidean distance, which is defined as

$$SED(X, Y) = \sqrt{\sum_{w \in W_L} \left(f_{X_w} - f_{Y_w}\right)^2} \in [0, 1], \tag{5}$$

where w represents the *L-words* and f_{Z_w} means the relative frequency of w in the sequence Z.

Function *Distance* described in Supplementary Material automates this procedure.

2.2. Software. The algorithm was written in Python, version 2.5.2, and tested on a Windows 7 x32 system and on a Linux platform with a processor Intel (R) Pentium (R) Dual CPU, T3400 @ 2.16 GHz and 4 Gb of RAM. It is freely available on the web at http://www.portugene.com/SupMat/SuffixTree&Lwords.rar.

3. Results and Discussion

3.1. Phylogenetic Reconstructions. The developed algorithm was tested in different datasets of mtDNA sequences, proving to be a simple and fast way to identify phylogenetic relationships in the different sets of mitochondrial genomes.

The algorithm was first tested in a dataset composed of 29 complete primate mtDNA sequences representing genomes of different families, ranging from 15467 bp to 17036 bp long. Taking into account these lengths, we determined $L = 8$, as explained in the Methods Section. This value has proven to be a good choice, allowing the program to run quickly, while still producing a genetic distance matrix that,

when used to construct a dendrogram, exhibits a clustering that is in agreement with consensus primate phylogeny (http://tolweb.org/Primates/15963).

In order to confirm that the algorithm was also able to produce a correct phylogeny with closely related sequences we tested it with mtDNA sequences from the same species, in which the sequence length is more homogeneous. The observed clusterings are in general agreement with those published in the literature, grouping mtDNA genomes in the same clades previously published methodologies (Supplementary Material).

Aiming to check the performance of our algorithm as well as to compare the quality of the results obtained by our approach and Costa's methodology, we compare the topology of the resulting phylogenies. The dendrograms constructed using the genetic distance matrixes generated by our algorithm are consistent with consensus phylogenies (Supplementary Material), in contrast with the results obtained by Costa et al. [8] methodology, which show some discrepancies, namely, in the clade Platyrrhini, which is clustered with Tarsii and Strepsirrhini ((http://tolweb.org/Primates/15963) and [12]).

3.2. Running Time. Our algorithm takes a linear execution time to determine the words frequencies and a quadratic time to compute the pairwise distances, an improvement to previous word counting alignment-free methodologies.

Our approach was compared to the method developed by Costa et al. [8] in what concerns the running time (the word counting alignment-free methodology proposed by Sims et al. [7] could not be tested because it has not been made available). While our approach computes the optimal word length to determine the word frequency profiles and generates a genetic distance matrix just by inputting a *fasta* file with mtDNA sequences, the methodology proposed by Costa et al. [8] involves four steps/algorithms: (1) converting a *fasta* file containing n mtDNA sequences into n *fasta* files with a single sequence; (2) converting each file into a *fa* file, a simplified version of *fasta* files; running two additional algorithms to (3) generate the histograms files and (4) create a correlation similarity matrix. These last two algorithms must be tested in increasingly longer windows until a conserved correlation matrix is obtained.

Our approach was designed to be run in Windows x32 operative system but it was also tested in a Linux platform in order to be compared to the alternative methodology under the same operative system. We thus could demonstrate that, independently of the operating system, the use of suffix tree structures to compute the words frequency profiles enables our methodology to run in a much shorter time. Although for small sets of sequences the running time required by Costa's (2011) methodology [8] is shorter, when increasing the number of sequences to over a hundred, the performance of our method is clearly better (Table 1, Figure 1, Supplementary Table 5).

3.3. Final Remarks. The algorithm described here has demonstrated to be an improvement of word counting alignment-free methods for sequence clustering, showing to be computationally very fast, particularly with large datasets, while still producing good quality results. In fact, by combining suffix tree structures with word counting tasks, as well as automating the determination of a single optimal word length, a significant decrease in running time and memory requirements for *L-words* frequencies determination was obtained.

The method proved to be efficient and powerful when applied to complete mitochondrial genomes, either from different species or intraspecifically, being able to quickly cluster the sequences in accordance to acknowledged phylogenetic relationships.

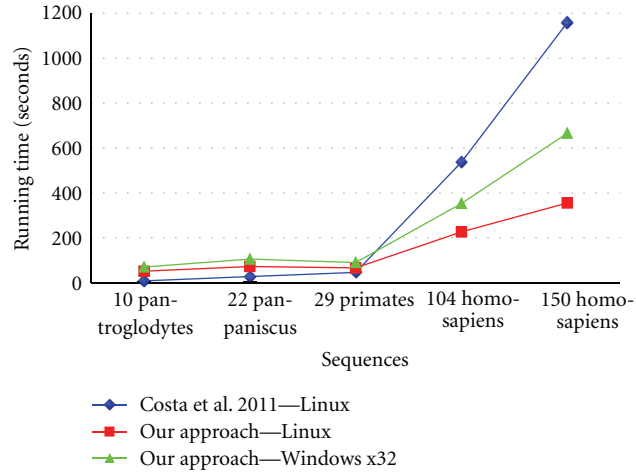

Figure 1: Differences between running times of Costa et al. 2011 [8] approach and our suggested methodology.

Table 1: Comparison of the running times between our approach (Linux and Windows x32 operative systems) and Costa et al. 2011 methodology (Linux platform) [8]. The first column lists the number of sequences and species used in each comparison; the second and third summarize the running times of each algorithm for each set of sequences, in seconds. The tabulated times correspond to the sum of running times of each step. The time spent by the user between each step, although highly time consuming, was not included.

Sequences	Running time (seconds)		
	Costa et al. 2011 [8]	our approach	
	Linux	Linux	Windows
10 Pan troglodytes	8	51	70
22 Pan paniscus	27	72	105
29 primates	46	66	90
104 Homosapiens	537	226	353
150 Homosapiens	1159	355	666

Authors' Contribution

I. Soares developed the algorithm, performed the tests, and wrote the paper. A. Goios participated in the design of the study and wrote the paper. A. Amorim conceived the study,

and participated in its design and coordination and wrote the paper. All authors read and approved the final paper.

Acknowledgments

The authors want to thank to António Guedes de Oliveira and Pedro Silva for their helpful suggestions and the three anonymous reviewers for helpful comments that greatly improved the manuscript. I. Soares has a doctoral Grant (SFRH/BD/38171/2007) and A. Goios has a postdoctoral Grant (SFRH/BPD/43646/2008) from Fundação para a Ciência e Tecnologia. IPATIMUP is an Associate Laboratory of the Portuguese Ministry of Science, Technology and Higher Education and is partly supported by Fundação para a Ciência e Technology. CMUP was funded by the European Regional Development Fund through the programme COMPETE and by the Portuguese Government through the FCT under the Project PEST-C/MAT/UI0144/2011. The funding sources had no involvement in any part of this study.

References

[1] H. J. Bandelt, V. Macaulay, and M. Richards, "Median networks: speedy construction and greedy reduction, one simulation, and two case studies from human mtDNA," *Molecular Phylogenetics and Evolution*, vol. 16, no. 1, pp. 8–28, 2000.

[2] M. Wu and J. A. Eisen, "A simple, fast, and accurate method of phylogenomic inference," *Genome Biology*, vol. 9, no. 10, article R151, 2008.

[3] N. Homer, B. Merriman, and S. F. Nelson, "BFAST: an alignment tool for large scale genome resequencing," *PLoS ONE*, vol. 4, no. 11, Article ID e7767, 2009.

[4] S. V. Angiuoli and S. L. Salzberg, "Mugsy: fast multiple alignment of closely related whole genomes," *Bioinformatics*, vol. 27, no. 3, Article ID btq665, pp. 334–342, 2011.

[5] S. Vinga and J. Almeida, "Alignment-free sequence comparison—a review," *Bioinformatics*, vol. 19, no. 4, pp. 513–523, 2003.

[6] R. C. Edgar, "MUSCLE: multiple sequence alignment with high accuracy and high throughput," *Nucleic Acids Research*, vol. 32, no. 5, pp. 1792–1797, 2004.

[7] G. E. Sims, S. R. Jun, G. A. Wu, and S. H. Kim, "Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 8, pp. 2677–2682, 2009.

[8] A. M. Costa, J. T. Machado, and M. D. Quelhas, "Histogram-based DNA analysis for the visualization of chromosome, genome and species information," *Bioinformatics*, vol. 27, no. 9, pp. 1207–1214, 2011.

[9] M. Domazet-Lošo and B. Haubold, "Efficient estimation of pairwise distances between genomes," *Bioinformatics*, vol. 25, no. 24, Article ID btp590, pp. 3221–3227, 2009.

[10] D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computer Biology*, Cambridge University Press, New York, NY, USA, 1997.

[11] M. H. Schulz, S. Bauer, and P. N. Robinson, "The generalised κ-truncated suffix tree for time-and space-efficient searches in multiple DNA or protein sequences," *International Journal of Bioinformatics Research and Applications*, vol. 4, no. 1, pp. 81–95, 2008.

[12] M. A. Carrigan, O. Uryasev, R. P. Davis, L. Zhai, T. D. Hurley, and S. A. Benner, "The natural history of class I primate alcohol dehydrogenases includes gene duplication, gene loss, and gene conversion," *PLoS ONE*, vol. 7, no. 7, Article ID e41175, 2012.