2019 International Conference on Advances in ICT for Emerging Regions (ICTer): 1-7

Alignment-free Whole Genome Comparison Using k-mer Forests

G. Gamage[#1], N. Gimhana[#2], A. Wickramarachchi[*3], V. Mallawaarachchi[*4], I. Perera[#5]

[#]*Department of Computer Science and Engineering,*
University of Moratuwa
Moratuwa, Sri Lanka
{[1]gihangamage.15, [2]gimhanadesilva.15, [5]indika}@cse.mrt.ac.lk

[*]*The Australian National University*
{[3]anuradha.wickramarachchi, [4]vijini.mallawaarachchi}@anu.edu.au

***Abstract*— In evolutionary biology, the study of phylogenetics can be considered as one of the main research disciplines. Phylogenetics is based on comparative data, which is mainly DNA sequences or raw sequencing reads. Alignment-based sequencing and alignment-free sequencing are the two main similarity computation methods, which are used to find genetic relatedness of different species. Alignment-based methods are relatively complex and computationally challenging as the genome scales when considering mammalian datasets and complex metagenomic colonies. Moreover, they show poor accuracy in certain cases in genetic comparison due to misalignments and algorithmic tolerances. Alignment-free comparison methods perform much better in genetic distance computation by addressing most of the challenges observed in alignment-based methods. In this paper, we propose a novel alignment-free, pairwise, distance calculation method based on k-mers. With this, we convert longer DNA sequences into simplified k-mer forest structures, which makes the comparison more convenient. Further, we are using a specialized tree pruning approach, which minimizes tree comparison time considerably compared to other alignment-free methods.**

Keywords—phylogenetics; genetic comparison; genetic distance; k-mer forest

I. INTRODUCTION

Genetic distance is a measure of the genetic divergence between two species or between different populations within the same species. The genetic distance can be considered as one of the major criteria used to differentiate species by considering their characteristics [1]. For example, genetic distances are used to differentiate mosquito vectors of malaria, which is significant as different mosquito types are responsible for specific subtypes of malaria that require particular treatments specific to the disease subtype. Furthermore, it is useful to refill the black holes in the history of human population evaluation in population genomics and to deduce treatments for diseases that grow drug resistance over time [2]. According to the global research "Simons Genome Diversity Project", ancestors of some pairs of present-day human populations (Africans and Non-Africans) were substantially separated 100,000 years ago [3]. In addition to that, genetic distances are used to figure out the origin of biodiversity. Different breeds of domestic animals are often examined to determine its phenotypic features as such breeds should be protected to keep the genetic diversity equilibrium in the future [4].

Phylogenetic tree (Evolutionary tree) is a branching diagram, which shows the evolutionary relationship among various organisms. It is generated by considering the similarities and the deviations of species' physical or genetic characteristics. Identifying the origin of pathogens and conservation of rare species from data observed from them are some of the emerging use-cases of phylogenetic trees. The challenging part of the construction of the phylogenetic tree is the identification of the similarities precisely and efficiently. Genetic distance calculation can be considered as the backbone of phylogenetic tree construction. Hence the method proposed intends to gain improvements in both accuracy and performance of the phylogenetic tree construction. More specifically, interspecies distances calculated from the proposed approach can be used to make the distance matrix, which is the preliminary step of phylogenetic tree construction. Moreover, the distance matrix is a key input in many biological applications, including population genetics and metagenomic binning. Based on the distances we calculate, several research avenues have been made possible including but not limited to cluster species of into groups of predominant features, and predict common ancestors.

The paper is arranged into the following sections: Section II presents the relevant literature and background for the study. Section III describes the research materials and the dataset used in the research. Section IV presents the methodology followed, and Section V presents the Evaluation of the work. Finally, Section VI concludes the paper.

978-1-7281-5156-4/19/$31.00 ©2019 IEEE

II. BACKGROUND

In the field of bioinformatics, genome sequence alignment [5] is a method used to identify regions of similarities in sequences [6]. Similarities or dissimilarities between sequences are represented as genetic distances. There are three types of alignments; global, local and glocal (global+local). Global alignment is attempting to align every residue in every sequence. This is useful when the data set has more similarities and roughly equal in length. When sequences have fewer regions of similarities, local alignment is often used. Glocal is a hybrid method. It is for searching the best possible partial alignment of two sequences.

BLAST [7], (Basic Local Alignment Search Tool) is an algorithm, which is originally developed by a group of scientists with the collaboration under NIH (National Institutes of Health), can be considered as one of the widely used local alignment-based tools in bioinformatics since its simplicity and search capabilities. Most of the researchers claimed that alignment-based algorithms such as BLAST, has performed poorly as the sequence identity has increased and because it has provided only aligning regions discarding the mismatches [8]. Moreover, another general problem with alignment-based approaches is that it requires to do many preprocessing steps in sequences to make them eligible for alignment-based comparison. Hence several practical issues are also found in such implementations.

Another dominant issue in alignment-based sequencing method is it encounter poor accuracy when getting the distance between sequence pairs where one is the repetition of the other. For example, the human genome sequence can be considered as a repetition of the mouse genome sequence, because of that alignment-based tools like BLAST gives poor similarity between those genomes, which is inaccurate. Further, these methods are highly time-consuming approaches [9]. The accuracy of sequence alignments drops off rapidly in cases where the sequence identity falls below a certain critical point. In addition, multiple-sequence alignment is an NP-hard problem; means cannot process realtime. However, more modern forms of alignment use prefix indexes [10] and compression transformations such as Borrow Wheeler Transform.

Because of these shortcomings of alignment-based genome sequencing algorithms, the alignment-free algorithms were introduced. Generally, alignment-free approaches have been used in sequence similarity searches [11], clustering and classification of sequences [12]. Alignment-free methods are identified as the most appropriate choice for most genome comparison experiments because they are computationally inexpensive. Nonetheless, since they are based on occurrences of sub-sequences they usually can be memory intensive [13]. Moreover, comparing to alignment-based methods, these are at the development level yet. Hence requires further testing for robustness and scalability when applying in phylogenetic applications.

Recently, methods such as k-mer and word frequency tend to be used in applications related to phylogenetics without doing alignment [14]. In general, comparing word frequencies is a lot easier than aligning huge genome sequences, which require heavy computation [15]. Therefore, clearly these methods outperform the alignment-based methodologies. For the comparison of multiple whole genome sequences, multiple sequence alignment of a few selected genes is not appropriate. One approach is to use an alignment-free method in which feature (or k-mer) frequency profiles (FFP) of whole genomes are used for the comparison supported by ‘a variation of a text or book comparison’ method, using word frequency profiles [16]. Composition vector (CV) is another approach, which calculates the normalized frequency of each possible kmer of the sequence.

Here genetic distance is approximated using Cosine distance function [17],[18]. Both FFP and CV is based on the frequencies of word presence and based on that they give novel interpretation to the genome sequence. Apart from that, there are some other methods such as spaced-word frequencies which match words based on a predefined pattern and does not consider their positions in the sequence [19].

Return time distribution (RTD) is another alignment-free method which is different from the above-mentioned methods. Instead of word count, it considers the amount of time required for the reappearance of k-mers [20]. As in summary, most of these methods made some sort of errors while predicting the genetic distances as they are using different method-specific variables.

In contrast when considering these alignment-free methods all of them use some kind of estimation of genome sequence to simplify the comparison such as feature frequency vectors, return time distributions etc. But ultimately with these approaches accuracy of genetic distance is decreasing as we do comparison on estimations of the sequence rather than comparing exact sequence. In order to gain higher accurate and sensitive genetic comparison we should consider what are the common and distinct k-mers/words when considering two sequences. But this is usually very much time consuming and complex to do.

However, in our approach, we build k-mer forests, which can be used as a direct representation of distinct k-mers in the a genome sequence. The set of distinct k-mers in a species' genome or a genomic region can be considered as the signature of that underlying sequence [21]. For the distance computation, we use the counts of common and distinct k-mres in comparing sequences, which give highly accurate and direct similarity value based on the Jaccard index [22]. It is expected to generate the distance with minimum error compared to existing alignment free methods such as RTD, and spaced-word frequency as this method directly considers similar and dissimilar word counts instead of estimations such as reappearance time or occurrence patterns.

Moreover with our method efficiency is also satisfied as we are building kmer forests and do comparison on them rather than scan larger genome sequences for kmer matches. Kmer forest comparison is also geared up with our pruning algorithm which we can expect up to 50% speed up in forest comparison. In nutshell it can be stated that our proposed approach can provide considerable good accuracy than existing alignment-free genetic comparison approaches with also maintain the efficiency.

III. RESEARCH MATERIALS AND DATASETS

A. Genomic Datasets

The whole-genome sequences (FASTA format - *.fna) were downloaded from the NCBI database (ftp.ncbi.nlm.nih.gov/genomes/). For the comparisons and results, we used genomes of Bolivian squirrel monkey ([23]), Honey Bee ([24]) and North American deer mouse ([25])

B. Development Tools and Environmental Configurations

For the K-mer counting, we used DSK k-mer counting tool ([26]). Python was the primary programming language used to implement algorithms.

Testing Environment: 32GB RAM, four cores of CPU, 1TB Storage, Ubuntu 16.04

After download the corresponding genome datasets from NCBI, we extracted k-mers using DSK tool. The output contains a list of k-mer strings and their frequencies. We converted that output to CSV(Comma Separated Values) to achieve the ubiquitousness and compactness of data.

C. Usage of odd k-mer sizes

One of the critical decision of k-mer listing is, using only odd values for the size of the k-mer. DSK tool considers a particular word and its reverse complement (i.e. palindrome) as the same object for enhancing the efficiency and removes redundancies. Reverse complements are known as palindromes in traditional computer science. Palindromes induce paths that fold back on themselves [27]. However, this can cause for losing data in some cases. When k size is even, there is a non zero possibility of information reduction.

For example, a k-mer like ATATATATATAT and its reverse complement is identical. Such palindromic k-mer can always be prevented by using odd k size. Reason for that is the center nucleotide will be changed in the reverse complement [28].

IV. METHODOLOGY

To solve the imperfections of previously mentioned methods, we propose an algorithm based on k-mer count (word count) as well as alignment-free. K-mers are substrings with a length of k. Most of the alignment-free tools and algorithms available today use techniques such as frequency profiles and return time distributions. [29]

With these methods, we cannot directly establish the count of unique k-mers in the sequences that we are comparing. Usually taking such a count, is complex and requires a considerable amount of time. In contrast, with our approach of k-mer forests, large genome sequence with several billions of k-mers can be converted to a simplified structure, which is straightforward to be compared.

With this algorithm, we can calculate the pairwise distance between genome sequences with high accuracy and efficiency. The proposed algorithm consists of two parts:

Part A - Creating k-mer forests for sequences

The first part of the algorithm is to construct k-mer forests for each of the genome sequences.

For that, it is required to list all distinct k-mers of the sequence. For this purpose, we have used DSK (disk streaming of k-mers) k-mer counting software, which lists k-mers with considerable low memory and disk usage [30]. Then we construct the k-mer forest using k-mer lists from each species using algorithm 1 and algorithm 2. The forest is made from iterating through all the k-mers of the sequence, which guarantees that there is a root to leaf pathway for each distinct k-mer. Each tree in the forest is k-deep. Maximum possible number of trees in k-mer forest for nucleotide sequence is 4 with possible A, C, T and G roots. If protein sequences have been used instead the number of trees became 20 as there are 20 possible roots. This leads to convert huge DNA sequence to simplified k-mer forest structure, which is more straightforward to compare. Figure 1 shows an example of constructing a k-mer forest for a given sequence.

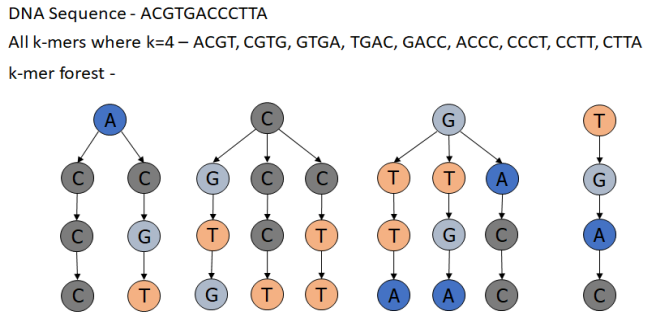

Figure 1 - Building a kmer forest for an example sequence

In this example, there are 9 k-mers with k=4 in the given DNA sequence. 2 k-mers are starting from A, 3 from starting C, G and 1 from starting T. All the forests are built as the above diagram. The algorithm is expected to perform better when there are more k-mers because the expansion of forest is minimum compared to the expansion of k-mer list.

Following algorithms show the implementation of tree construction. Algorithm 1 shows the functionality to add a new k-mer to the corresponding tree in the forest. Most importantly, it is ensured that no root to leaf path is repeated in the forest. For the forest implementation nested, dictionaries have been used here. In the algorithm 2, the tree is constructed by adding all the k-mers to the forest by using the algorithm-1.

Algorithm 1: Add a k-mer to the forest

Require: K-mer forest(F), New k-mer to be added(K)

```
1.  FUNCTION add_kmer_to_forest(F, K)
2.    IF NOT K[0] in F.keys()
3.        F[K[0]] = { }
4.    END IF
5.    Let PTR=Current Position of Forest
6.    Let ITR=Current Iteration
7.    PTR = F[K[0]]
8.    ITR = 1
9.    WHILE ITR < Length of K
10.       IF K[ITR] not in PTR.keys()
11.           PTR [ K [ ITR ] ] = { }
12.       END IF
13.       PTR = PTR [ K [ ITR ] ]
14.       ITR + = 1
15.   END WHILE
16. END FUNCTION
```

Algorithm 2: Forest construction

Require: K-mer list of specie(KL)

```
1.  FUNCTION forest_construction (KL)
2.    LET F = new forest
3.    FOR each kmer in KL
4.        add_kmer_to_forest(F,kmer)
5.    END FOR
6.  END FUNCTION
```

Part B - calculate distances using the tree comparison algorithm

After completing part A of the approach, we were able to convert huge DNA sequences into k-mer forests, which are compact and simplified. In this part, we are using a specified tree comparison algorithm based on pruning, to compare k-mer forests and calculate the genetic distances efficiently.

Following algorithms show the implementation of tree comparison mechanism.

Algorithm 3: Get child count of a given node of the tree

Require: Node of the tree (N)

```
1.  FUNCTION get_child_count (N)
2.    LET C = child count
3.    FOR each child in N
4.        IF N[child] not empty
5.            C + = get_child_count( N[child] )
6.        ELSE
7.            C + = 1
8.    END FOR
9.    Return C
10. END FUNCTION
```

Algorithm 4: Get distance between two forests

Require: Forest 1 (F1), Forest 2 (F2)

```
1.  FUNCTION get_distance (F1, F2)
2.    LET D = Number of pathways F1 has F2 doesn't
3.    LET S = Number of common pathways in forests
4.    FOR each node in F1
5.        IF node not in F2.nodes
6.            IF node is empty
7.                D + = 1
8.            ELSE
9.                D + = get_child_count(node)
10.       ELSE
11.           IF node is empty
12.               S+ = 1
13.           get_distance (F1[node], F2[node])
14.   END FOR
15.   LET U = Number of all pathways in forests
16.   LET J_I = Jaccard index
17.   U = D + F2.kmer_count
18.   J_I = S/U
19.   Return J_I
20. END FUNCTION
```

This mechanism is based on tree pruning. When comparing two forests, the genetic distance is calculated as several k-mers, which exists in the first forest but not in the second. When comparing two trees, it happens level by level from root to leaves. When any node found uncommon, all the pathways aka k-mers are counted using recursive algorithm 3 and added to the distance. With that, efficiency is drastically improved as no need for traversing children of such node. Algorithm 4, which is also a recursive algorithm, is responsible for finding uncommon nodes in trees of two forests. If those nodes are not left algorithm 3 works and pruning occurs.

Figure 2 shows an example of how pruning happens. Node A (with parent C), which is indicated in Forest I is absent in Forest II. Thus, pruning occurs, and child count,

which is equal to 5 is added to the distance without traversing in the circled subtree.

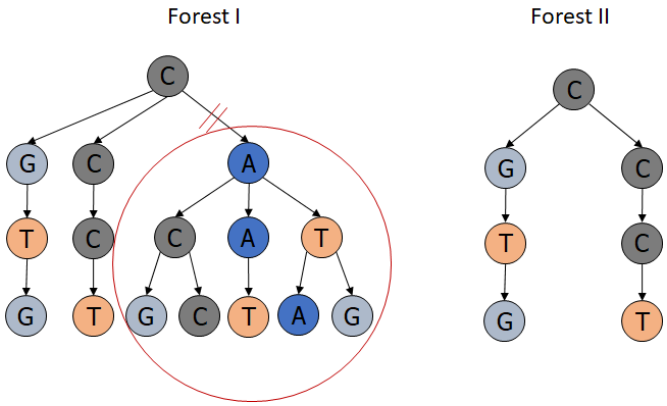

Figure 2 - k-mer forest pruning

Distance calculation happens according to the method of Jaccard index where distance is given in the following equation.

$$Distance = \frac{Number\ of\ common\ pathways\ in\ forests}{Number\ of\ all\ pathways\ in\ forests}$$

Here a pathway represents a root to leaf routine in the forest, which stands for a particular k-mer in the genome sequence. From algorithm 4, when comparing two forests (say A and B), several pathways exist only in forest A (D), and the number of common pathways in both forests (S) are counted. After that, to get a number of all pathways in forests, D is added up with k-mer count of forest B. finally Jaccard index is calculated from the collected data.

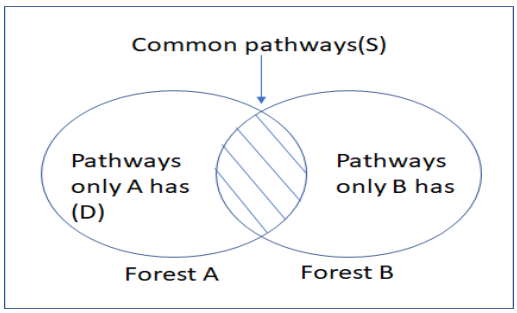

Figure 3 - Venn diagram showing the pathways of forest A and B

V. RESULTS AND EVALUATIONS

In this section, we show some of the evaluations done on our genome comparison methodology. As we have discussed above, tree construction and tree comparison are the two main operations we do in pair-wise forest comparison. We select appropriate k value for distance calculation of one of the important decision in word counting sequencing approaches.

Following graph number of distinct k-mers we get when k is varying. When moving to larger k value from smaller, it reaches a peak and gradually reducing afterward. This k value, which results in a maximum number of distinct k-mers can be considered as the optimal k value for the forest construction algorithm as more distinct k-mers describe the sequence better.

Here as the genome sequence, we have taken the whole DNA sequence of Apis mellifera (honey bee)

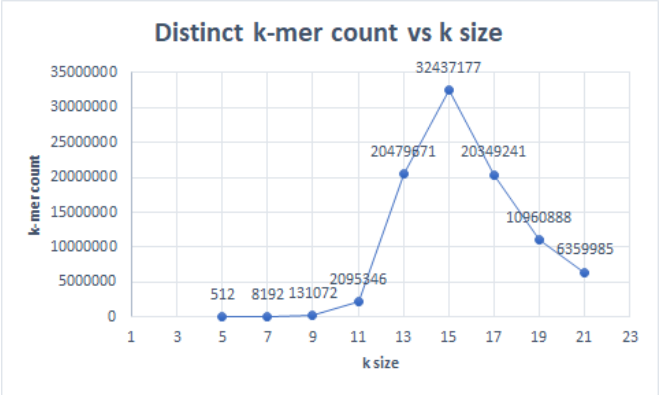

Figure 4 - Distant kmer count vs k-mer size graph

Forest construction time is also varying with k value as the number of distinct k-mers is also changing. When considering the forests for different k values, the maximum possible number of trees is 4 for genome sequences. Addition to that maximum depth of each tree is equal to the k value. Even though genome sequence is extremely large with k-mer forests, we can convert them to the more compact structure, which is convenient for further analysis and distance calculation.

Following diagram shows how forest construction time varies with k value. At the peak of k-value 15, which has 32437177 distinct k-mers it shows the maximum forest construction time, which is less than 6 minutes, a considerably fast result.

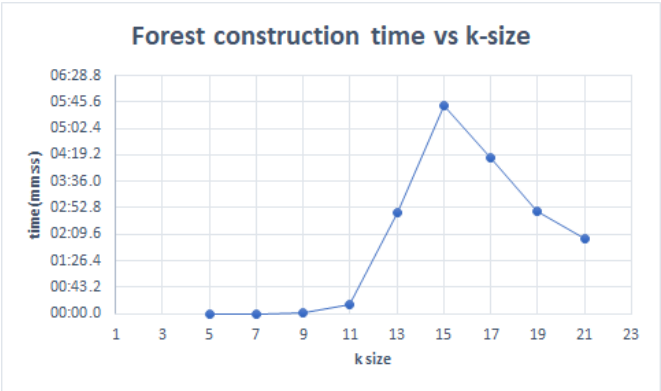

Figure 5 - Forest construction time vs k-mer size graph

As the next operation, we have evaluated how forest comparison time varies with the different k values of the genome. Here for the comparison, we have compared k-mer forest of Apis mellifera with itself. By that, we have tested the worst-case time consumption in tree comparison as no

pruning is applied. Here also the graph looks similar to the previous two graphs and peak meets at 15. As graph shows worst-case forest comparison time at the peak take only about 50 seconds to do the entire forest comparison. This is again considerably high performance compared to other alignment-free approaches.

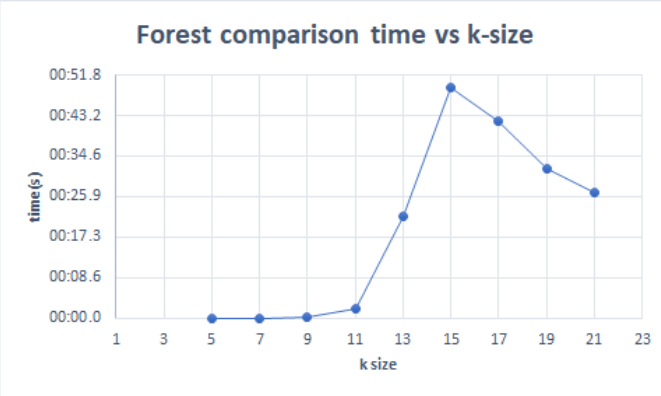

Figure 6 - Forest comparison time vs k-size graph

Performance of tree comparison algorithm

As we discussed above, the tree comparison algorithm is based on tree pruning. If a particular node does not exist in the other forest without traversing through its child nodes, all subroutines of that node will be added to the distance using algorithm 3. With this tree, comparison time can be improved by up to 50%.

Following table show percentage of speed up in comparison when pruning occurs. For this, we have taken the 13-mer forest of Peromyscus_maniculatus which consist of 67108864 distinct k-mers in 4 trees. There are 5 cases and each of those cases forest is compared with itself by removing trees. Speed up is calculated concerning the non-pruning case.

Tree count in forest A	Tree count in forest B	Pruned tree count	Comparison time(s)	Percentage speedup
4	4	0 (non pruning case)	28.927	0%
4	3	1 25%	27.413	5.25%
4	2	2 50%	24.743	14.48%
4	1	3 75%	20.326	29.73%
4	0	4(all trees are pruned)	14.408	50.19%

Table 1 - Comparison time speedup with pruning

VI. DISCUSSION AND FUTURE WORK

Here in this paper, we presented a whole genome comparison method based on k-mer forest construction. One of the important decision of the algorithm was to find the optimal k value, which gives maximum distinct k-mer set. In alignment-free word counting sequencing approaches, the genome of described better when there are more words. Similarly, in our approach best suitable k-mer forest for a particular genome is made at this optimal k value.

The outcome of the discussed approach is to generate an accurate distance matrix efficiently to construct a phylogenetic tree. Distance matrix plays a major role in phylogenetic tree construction. Since it has a linear relationship between the distance matrix and phylogenetic tree construction, the accuracy of distances directly affects the reliability of the tree and as well the evolutionary relationships of taxonomies. The proposed whole genome comparison using a k-mer forest approach guarantees the accuracy of distances while computing those amidst of optimum computational time consumption background. As our next step, we suppose to construct phylogenetic trees based on the distance matrices we are building based on this methodology. For phylogenetic tree construction, we propose a novel approach which is based on machine learning.

VII. CONCLUSION

Our method of whole genome sequence comparison can be used for genomes with diverse lengths and different amounts of similarities (closely related or not). Most of the alignment-free comparison methods give approximate distances based on techniques such as frequency profiles, reappearing time and character patterns. However, with our approach, we give the exact distance from Jaccard index with taking consideration of common and distinct k-mer count. Moreover, with our k-mer forest construction converts huge genome sequences into a more simplified and structured representation. This expedites and smoothes up genome comparison as well as other genome analytics.

Addition to that pruning based tree comparison algorithm can do considerable speed up in distance calculation by directly taking several pathways without traversing. Evaluations show that our approach guaranteed to provide result in a shorter period of time even at their optimal resolutions (at the number of maximum k-mers).

REFERENCES

[1] Contributors to Wikimedia projects, "Genetic distance - Wikipedia," *Wikimedia Foundation, Inc.*, 21-Sep-2005. [Online]. Available: https://en.wikipedia.org/wiki/Genetic_distance.[Accessed:10-May-2019].
[2] G. G. M. N. D. E Meynell, "NCBI," March 1968. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC378292/. [Accessed May 2019].
[3] S. Mallick *et al.*, "The Simons Genome Diversity Project: 300 genomes from 142 diverse populations," *Nature*, vol. 538, no. 7624, p. 201, Sep. 2016.

[4] Ruane, J. (1999). A critical review of the value of genetic distance studies in conservation of animal genetic resources. Journal of Animal Breeding and Genetics, 116(5), 317-323. Chicago.
[5] Vinga, S; Almeida, J (Mar 1, 2003). "Alignment-free sequence comparison-a review". *Bioinformatics*. 19 (4): 513–23. doi:10.1093/bioinformatics/btg005. PMID 12611807.
[6] Mount DM. (2004). *Bioinformatics: Sequence and Genome Analysis* (2nd ed.). Cold Spring Harbor Laboratory Press: Cold Spring Harbor, NY. ISBN 978-0-87969-608-5.
[7] "BLAST: Basic Local Alignment Search Tool", *Blast.ncbi.nlm.nih.gov*, 2019. [Online]. Available: https://blast.ncbi.nlm.- nih.gov/Blast.cgi. [Accessed: 08- May- 2019].
[8] "BWA," Github, [Online].Available: https://github.com/lh3/bwa. [Accessed 9 May 2019].
[9] "Website." [Online]. Available: https://genomebiology.biomedcentral-.com/articles/10.1186/s13059-017-1319-7. [Accessed: 08-May-2019].
[10] "Minimap2: pairwise alignment for nucleotide sequences. - NCBI - NIH." https://www.ncbi.nlm.nih.gov/pubmed/29750242. Accessed 20 Mar. 2019.https://github.com/lh3/bwa. Accessed 20 Mar. 2019.
[11] Hide, W; Burke, J; Davison, DB (1994). "Biological evaluation of d2, an algorithm for high-performance sequence comparison". *Journal of Computational Biology*. **1** (3): 199–215. doi:10.1089/cmb.1994.1.199. PMID 8790465.
[12] Miller, RT; Christoffels, AG; Gopalakrishnan, C; Burke, J; Ptitsyn, AA; Broveak, TR; Hide, WA (1999). "A comprehensive approach to clustering of expressed human gene sequence: the sequence tag alignment and consensus knowledge base". *Genome Research*. **9** (11): 1143–55. doi:10.1101/gr.9.11.1143. PMC 310831. PMID 10568754.
[13] M. A. R. Cheong Xin Chan, "Next-generation phylogenomics," *Biol. Direct*, vol. 8, p. 3, 2013.
[14] Chan, CX; Ragan, MA (Jan 22, 2013). "Next-generation phylogenomics". *Biology Direct*. **8**: 3. doi:10.1186/1745-6150-8-3. PMC 3564786. PMID 23339707.
[15] R. Bromberg, N. V. Grishin, and Z. Otwinowski, "Phylogeny Reconstruction with Alignment-Free Method That Corrects for Horizontal Gene Transfer," *PLoS Comput. Biol.*, vol. 12, no. 6, p. e1004985, June. 2016.
[16] S. GE, J. SR, W. GA and K. SH, "Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions.", 2009. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/19188606. [Accessed: 09- May- 2019].
[17] Apostolico, A; Denas, O (March 2008). "Fast algorithms for computing sequence distances by exhaustive substring composition". *Algorithms for Molecular Biology*. **3**.
[18] Apostolico, A; Denas, O; Dress, A (September 2010). "Efficient tools for comparative substring analysis". *Journal of Biotechnology*. **149** (3): 120–126. doi:10.1016/j.jbiotec.2010.05.006. PMID 20682467.
[19] Leimeister, CA; Boden, M; Horwege, S; Lindner, S (2014). "Fast alignment-free sequence comparison using spaced-word frequencies". *Bioinformatics*. **30** (14): 1991–1999. doi:10.1093/bioinformatics/btu177. PMC 4080745. PMID 24700317.
[20] E. al Kolekar P, "Alignment-free distance measure based on return time distribution for sequence analysis: applications to clustering, molecular phylogeny and subtype... - PubMed - NCBI." [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/22820020. [Accessed: 08-May-2019].
[21] L. H, "Minimap2: pairwise alignment for nucleotide sequences.", 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/29750242. [Accessed: 09- May- 2019].
[22] "Jaccard index", En.wikipedia.org, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Jaccard_index. [Accessed: 11- May- 2019].
[23] "Saimiri boliviensis boliviensis (ID 6907) - Genome - NCBI." [Online]. Available: https://www.ncbi.nlm.nih.gov/genome/?term=Saimiri_boliviensis_boliviensis. [Accessed: 08-May-2019].
[24] "Apis mellifera (ID 48) - Genome - NCBI." [Online]. Available: https://www.ncbi.nlm.nih.gov/genome/?term=honey+bee. [Accessed: 08-May-2019].
[25] "Peromyscus maniculatus (ID 11397) - Genome - NCBI." [Online]. Available: https://www.ncbi.nlm.nih.gov/genome/?term=Peromyscus_maniculatus. [Accessed: 08-May-2019].
[26] GATB, "GATB/dsk," GitHub. [Online]. Available: https://github.com/GATB/dsk. [Accessed: 08-May-2019].
[27] J. R. Miller, S. Koren, and G. Sutton, "Assembly Algorithms for Next-Generation Sequencing Data," *Genomics*, vol. 95, no. 6, p. 315, Jun. 2010.
[28]"Tutorials." [Online]. Available: https://homolog.us/Tutorials/book-4/p2.4.html. [Accessed: 08-May-2019].
[29] G. Rizk, D. Lavenier, and R. Chikhi, "DSK: K-mer counting with very low memory usage," *Bioinformatics*, vol. 29, no. 5, Jan. 2013.
[30] S. Mallick *et al.*, "The Simons Genome Diversity Project: 300 genomes from 142 diverse populations," *Nature*, vol. 538, no. 7624, p. 201, Sep. 2016