INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research

International Journal of Bioinformatics Research and Applications

ISSN online: 1744-5493 - ISSN print: 1744-5485
https://www.inderscience.com/ijbra

A novel algorithm for genomic STR mining: application to phylogeny reconstruction and taxa identification

Uddalak Mitra, Soumya Majumder, Sayantan Bhowmick

DOI: 10.1504/IJBRA.2024.10062507

Article History:

Received:	07 February 2023
Last revised:	05 June 2023
Accepted:	23 June 2023
Published online:	14 March 2024

A novel algorithm for genomic STR mining: application to phylogeny reconstruction and taxa identification

Uddalak Mitra*, Soumya Majumder and Sayantan Bhowmick

Siliguri Institute of Technology,
Hill Cart Road, Salbari, District Darjeeling,
Sukna, Siliguri, West Bengal, 734009, India
Email: uddalakmitra@gmail.com
Email: SIT000178@technoindiagroup.in
Email: soumyamajumdersm90@gmail.com
Email: sayantanbhowmick27523@gmail.com
*Corresponding author

Abstract: With vast collection of whole genome data, analysts require faster and more scalable bioinformatics tools to compare those abundant sequences for knowledge discovery. Despite of their availability, utilising the larger whole genomes for phylogeny reconstruction and taxa identification is still a challenging task. In complex organisms, a substantial portion of genome is made up of repetitive DNA. The short tandem repeat (STR) is one of the most crucial repeats. We develop an efficient and scalable algorithm called STR seed selection (3S), which mines STRs in whole genomes using k-mer comparison. The analysis of short tandem repeats has revealed species-specific variations that serve as crucial indicators of their genetic relatedness. When it comes to reconstructing the phylogeny and identifying taxa within eukaryotic species, the utilisation of short tandem repeat variants consistently matches with the established taxonomy by NCBI. With its remarkable attributes of minimal memory usage, rapid processing capabilities, and exceptional scalability, 3S emerges as a cutting-edge approach for biosequence analysis based on short tandem repeats.

Keywords: short tandem repeat; STR; short tandem repeat mining; short tandem repeat-based phylogeny; eukaryotic taxa identification using STR.

Reference to this paper should be made as follows: Mitra, U., Majumder, S. and Bhowmick, S. (2024) 'A novel algorithm for genomic STR mining: application to phylogeny reconstruction and taxa identification', *Int. J. Bioinformatics Research and Applications*, Vol. 20, No. 1, pp.21–41.

Biographical notes: Uddalak Mitra is an Assistant Professor, Department of MCA, Siliguri Institute of Technology (SIT).

Soumya Majumder is a BTech (CSE) student of Siliguri Institute of Technology.

Sayantan Bhowmick is a BTech (CSE) student of Siliguri Institute of Technology.

1 Introduction

With advent of new generation sequencing technologies, a large collection of eukaryotic genomes are now freely available in public repositories. Analysis of those data has confirmed the hypothesis that eukaryotic genomes are rich in repetitions, in particular, tandem repeats (TRs) whose functional role needs to be elucidated. A TR is certain number of juxtaposed repetitions of a group of k nucleotides, termed as k-mer or motif. Depending on the motif length, TRs are classified into three categories: microsatellites or short tandem repeats (STRs) (1–6 nt), minisatellites (7–60 nt) and satellites (larger than 60 nt). Highly poly-morphic nature of the STRs is extremely useful for linkage analysis (Ott et al., 2015), genotyping (Kashi et al., 1997) and DNA fingerprinting (Zietkiewicz et al., 1994). Variation in the repeat copy number of STRs at coding regions is linked to several neurodegenerative diseases in human such as Huntington's disease and Spinocerebellar Ataxia (Usdin, 2008). STRs have also been observed to play crucial roles in epigenetic regulation of gene expression (Greene et al., 2007) and genome organisation (Kumar et al., 2013).

One of the primary techniques of finding STRs in genetic material like DNA is wetlab methodologies, such as the polymerase chain reaction (PCR), gel electrophoresis, and Southern blot techniques. Although they have long been employed for the detection of STRs, these traditional approaches, however, suffer from protracted turnaround times, often spanning weeks, and in some cases even stretching into months (Ishiura et al., 2018). In a quest for expedited and cost-effective alternatives, computational methods have emerged as beacons of hope. These in-silico strategies offer swifter and more economical means of extracting STRs, making them increasingly attractive to the scientific community.

Contemporary computational approaches for STR extraction can be broadly classified into two categories: heuristic and combinatorial methodologies (Lim et al., 2012). The heuristic methods (Domanic and Preparata, 2007; Do et al., 2008) rely on probabilistic models to identify STRs, but their comprehensiveness and accuracy remain subject to scrutiny. In contrast, combinatorial methods (Wirawan et al., 2010; Pickett et al., 2017) adhere to stringent construction rules that facilitate the detection of STRs within DNA sequences. By initially capturing all STRs, including redundant occurrences, and subsequently subjecting them to refined filtering techniques, these approaches furnish the final set of STRs. Regrettably, the sequential nature of these combinatorial methods engenders a computational complexity that approximates quasi-linearity (O (N log$N$)), with the sequence length N-nt.

Recent advances have yielded two promising methods: Kmer-SSR (Pickett et al., 2017) and PERF (Avvaru et al., 2017). These innovative approaches integrate the detection and filtering stages, thereby endowing them with linear time complexity. Nonetheless, when confronted with the formidable challenge of processing vast genomic landscapes, both methods encounter obstacles. Kmer-SSR, in its pursuit of comprehensiveness, endeavours to identify all subsequences of a given length (k) and dutifully preserves the precise locations of each k-mer motif along the sequence. While the detection of tandem occurrences of a k-mer proves straightforward using the stored positional information, but the associated memory and computational demands become increasingly burdensome as the sequence length escalates. Furthermore, due to its dependence on multiple sequential scans of the sequence for each motif size, Kmer-SSR confronts scalability limitations [Figure 2(b)].

In contrast, *PERF* initiates its journey by artificially generating motifs of varying lengths, spanning from a minimum (m) to a maximum (M). Each motif then begets a repeat string through the concatenation of multiple tandem repetitions, ceasing only when the cumulative length reaches a user-specified cut-off (L nt). Consequently, for a motif length reaching the zenith (M), *PERF* engenders an astronomical number of distinct repeat strings-$4*(4^M-1)/(4-1)$, to be precise-each measuring L nt in length. As the computational burden exponentially burgeons with longer cutoffs and larger M values, storage and retrieval of these gargantuan repeat strings within a dictionary give rise to formidable space-time complexity [Figure 3(b)]. Furthermore, the subsequent string matching endeavour ensues quasi-linear STR extraction ($O(N L)$), a predicament further amplified when higher cut-off values are employed. Pertinently, PERF's inability to discriminate between atomic and non-atomic k-mers necessitates additional filtering to extricate STRs of non-atomic motifs. Consequently, both methodologies grapple with the vexing challenge of scalability [refer to Figures 2(a) to 3(b) for illustrative depictions]. Against this backdrop, the present study embarks on an innovative exploration of the STR mining conundrum, adroitly adopting a fresh perspective as described in the following Section 2.

Traditionally, the frequencies of STR alleles at particular loci relative to a reference genome are used to establish differentiation between the genomes of different organisms and their groupings (Li et al., 2018; Guo et al., 2018, Gou et al., 2020; Chen et al., 2022; Lewis et al., 2020). As a result, these methods heavily rely on the positional data of STR loci and the reference genome. Such methods are only applicable to closely related species with clearly characterised reference genomes. The current method deviates from the conventional one in that it harvests STRs from a single genome and concatenates them to create a repeat sequence made from the mined STRs. The real genomes are compared based on the variations in the STR made repeat sequences. Interestingly, we find that these variations can be used to correctly distinguish several taxonomic groups of eukaryotic species and hence grouping and classification of their taxa becomes plausible. Experiments on phylogeny tree reconstruction and taxa identification are 100% congruent with the data provided in NCBI. The results indicate that STR variation is a fundamental characteristic of complex organisms and can be used to infer details about their individuality and group identity. Due to its linear time computational complexity, the method is also a sophisticated tool to mine STRs from larger genomes of evolutionary higher eukaryotic species.

The major contributions of the research can be listed as follows:

1 Use of STRs for phylogenetic analysis: the research demonstrates the effectiveness of utilising STRs for phylogenetic analysis. By analysing the variations in STRs across different taxa of complex organisms, the research provides a valuable tool for inferring genetic relationships and reconstructing phylogenies. More importantly, it can be considered as a tool for phylogeny reconstruction of gigantic genomes.

2 Taxa-specific variation: the study highlights the taxa-specific nature of STR variation. By identifying and analysing STRs in different taxa, the research enables the identification and differentiation of genomic taxa based on their unique STR profiles.

3 Reference-free genetic relationship inference: a significant contribution of the research is the development of a reference-free approach for inferring genetic

relationships. By relying solely on the variations in STRs without the need for a reference genome or specific STR loci, the research offers a practical solution for genetic relationship analysis in scenarios where such references are not available.

4 Characterisation of STR origins: the research provides insights into the origins of STRs by identifying their atomic motifs. By exploring the characteristics of atomic motifs that do not generate cyclic-redundant and enclosing TRs, the research contributes to a better understanding of the mechanisms underlying STR formation.

5 Development of the 3S algorithm: the research introduces the 3S algorithm as a powerful tool for mining STRs in whole genomes. The algorithm is highly scalable and offers a linear order computational complexity, making it suitable for analysing longer whole genomes efficiently.

These contributions collectively advance the field of genetic analysis by leveraging STRs to infer genetic relationships, overcome the limitations of reference genomes, and gain insights into the origins of STRs.

The rest of the article is organised as follows: Section 2: materials and methods. This section provides a detailed description of the methods and algorithms employed in the research. It outlines the approach taken to mine STRs and explains the underlying principles and techniques utilised. Section 3: experiments. The experiments are conducted to evaluate the proposed method presented in this section. Section 3.1: comparative performance in STR mining: this subsection focuses on comparing the performance of the proposed method against existing approaches for STR mining. Various metrics, such as accuracy, efficiency, and scalability, are used to evaluate and compare the results. Section 3.2: STR mining from whole genomes: here, the experiments specifically target the mining of STRs from entire genomes. Section 3.3: phylogeny reconstruction: in this subsection, experiments are conducted to reconstruct phylogenies using the extracted STR data. Section 3.4: taxa identification: experiments related to the identification of genomic taxa using the extracted STRs are presented in this subsection. Section 4: conclusion: the article concludes in this section by summarising the key findings, contributions, and implications of the research. It also highlights potential future directions and areas for further investigation or improvement in the field of STR mining and genetic relationship inference. By organising the article into these sections, the research presents a comprehensive and logical flow of information, starting from the methods and algorithms employed, followed by the experimental results, and concluding with the overall findings and implications.

2 Materials and methods

2.1 Concept of proposed STR mining and formation of repeat sequence

TRs are formed from specific atomic motifs, which we refer to as ‘seeds’. In this context, a motif can have a variable length L(–1)1, ranging from L (the maximum motif length) to 1 nucleotide. The algorithm systematically examines each motif, starting from the longest length (L) and moving towards the shortest (1 nt), to determine if it meets certain criteria. First, it checks if the motif is atomic, and if so, it verifies whether there is an adjacent repetition of the motif in a tandem fashion. Additionally, it ensures that the two

repetitions are not cyclically redundant or enclosed within TRs of other longer motifs. Once a valid motif is identified, it becomes the seed for an ongoing STR pattern. The algorithm keeps track of the sustainability of this tandem pattern as it progresses through the sequence. This process continues until the entire sequence is traversed. Seeds of lengths ranging from L to 1 nt are successively chosen and followed to extract the STRs. These extracted STRs are then concatenated to create a repeat sequence, which can be used for comparing actual genomes.

2.2 Atomicity, cyclic-redundancy and enclosing properties

Consider a DNA sequence S of length N. A motif is a subsequence of length k, where $k = L$ (–1)1. The number of motifs of length k, we denote by M_k, is equal to 4^k. For STR, L is 6 and the total number of STR motifs is

$$M = \sum_{k=1}^{6} M_k = 4*(4^6 - 1)/(4-1) = 5,460 \tag{1}$$

A motif of length k is termed as atomic if no other motif of length $k' \leq k$ repeats in tandem within it. For instance, the DNA fragment TACACACACCCTACGTACGTACG TACATCAATATCAATATCAATG comprises two TRs of the motif ACAC (k = 4) beginning at position 2, but the motif itself consists of two repetitions of the motif AC (k = 2), hence the sequence ACAC is not atomic. In this situation, we just need to report the STR of the motif AC with a repeat count of 4. Additionally, a STR may have cyclic duplicate STRs at various points. As an illustration, ACGT (k = 4) has three TRs beginning at position 13 at the same example sequence. It should be noted that successive positions 14, 15, and 16 contain three cyclic duplicate STRs, namely CGTACGTACGTA, GTACGTACGTAC, and TACGTACG. Cyclic duplication is unnecessary and should be removed. Furthermore, if tandems of shorter motifs occur completely within a longer motif or its tandem, then the STRs of shorter motifs are termed as enclosed STRs and hence are redundant. For instance, the motif ATCAAT, which begins at position 27, appears as a STR with repeat count of 3. However, it has two STRs that start at positions 31 and 37 and have a repeat count of 2 for the motif AT; as a result, they are entirely enveloped by the STR of ATCAAT and should be ignored. In the sections that follow, we formally explain the method for removing the redundant STRs mentioned above through tests for atomicity, cyclic redundancy, and enclosing properties.

2.2.1 Formal description

- *Test of atomicity:* let ω_j^k be the j^{th} motif of length k, where $0 \leq j \leq 4^k - 1$. It is atomic if not totally constituted by TRs of any other motif of length $k' < k$. An atomic motif ω_j^k is in tandem if

  $$i_c - i_p = k, \tag{2}$$

  where i_c and i_p are locations of the first nucleotides of its two consecutive instances. TRs of a non-atomic motif do not satisfy equation (2), hence used by 3S to identify TRs of an atomic motif.

An STR, $\mathcal{R}_i^\tau\left(\omega_j^k\right)$ is τ ($\in Z+$) times tandem repetitions of an atomic motif ω_j^k, starting from nucleotide location i ($0 \le i < N - k$).

Its length $\lambda^k = \tau * k$. Let i_l denote the starting location of the last occurrence of ω_j^k in $\mathcal{R}_i^\tau\left(\omega_j^k\right)$. The next starting location of ω_j^k in $\mathcal{R}_i^\tau\left(\omega_j^k\right)$ is $i_n = i_l + k$, if the TR expands further. In that case, $\mathcal{R}_i^\tau\left(\omega_j^k\right)$ is under inspection from location i_l to i_n. We denote the set of motifs between il and in by Ω and the STRs formed by the motifs in Ω by Γ.

- *Test of cyclic-redundancy:* if an STR $\boldsymbol{\mathcal{R}}^\tau(\omega^k)$ is viewed from a locations $i + 1$, $i + 2, \cdots, i + r$ ($r < k$), another STRs $\mathcal{R}_{i+r}^\tau\left(\omega_j^k\right)$ would always be found, where $j' \neq j$ and τ' is either $\tau - 1$ or τ, and ω_j^k is r element right rotated version of ω_j^k. If there exists $\mathcal{R}_i^\tau\left(\omega_j^k\right) \in r$ and

$$i + r < i + \tau * (k - 1) \tag{3}$$

3S discards STRs like $\mathcal{R}_{i+\delta}^\tau\left(\omega_j^k\right)$ as a cyclic redundant STR of $\mathcal{R}_i^\tau\left(\omega_j^k\right)$.

- *Test of enclosing:* if tandems of shorter motifs occur completely within a longer motif or its tandem, then the STRs of shorter motifs are termed as enclosed STRs and hence are redundant. Analytically, if both the start and end positions of an STR of any shorter motif, viz. $\mathcal{R}_{i'}^{\tau'}\left(\omega_{j'}^{k'}\right)$, are within the respective indices of an STR of longer motif ω_j^k or the motif itself, then the former becomes redundant with respect to the later as it is enclosed. Thus,

$$\mathcal{R}_{i'}^{\tau'}\left(\omega_{j'}^{k'}\right) \subset \mathcal{R}_i^\tau\left(\omega_j^k\right) \tag{4}$$

iff

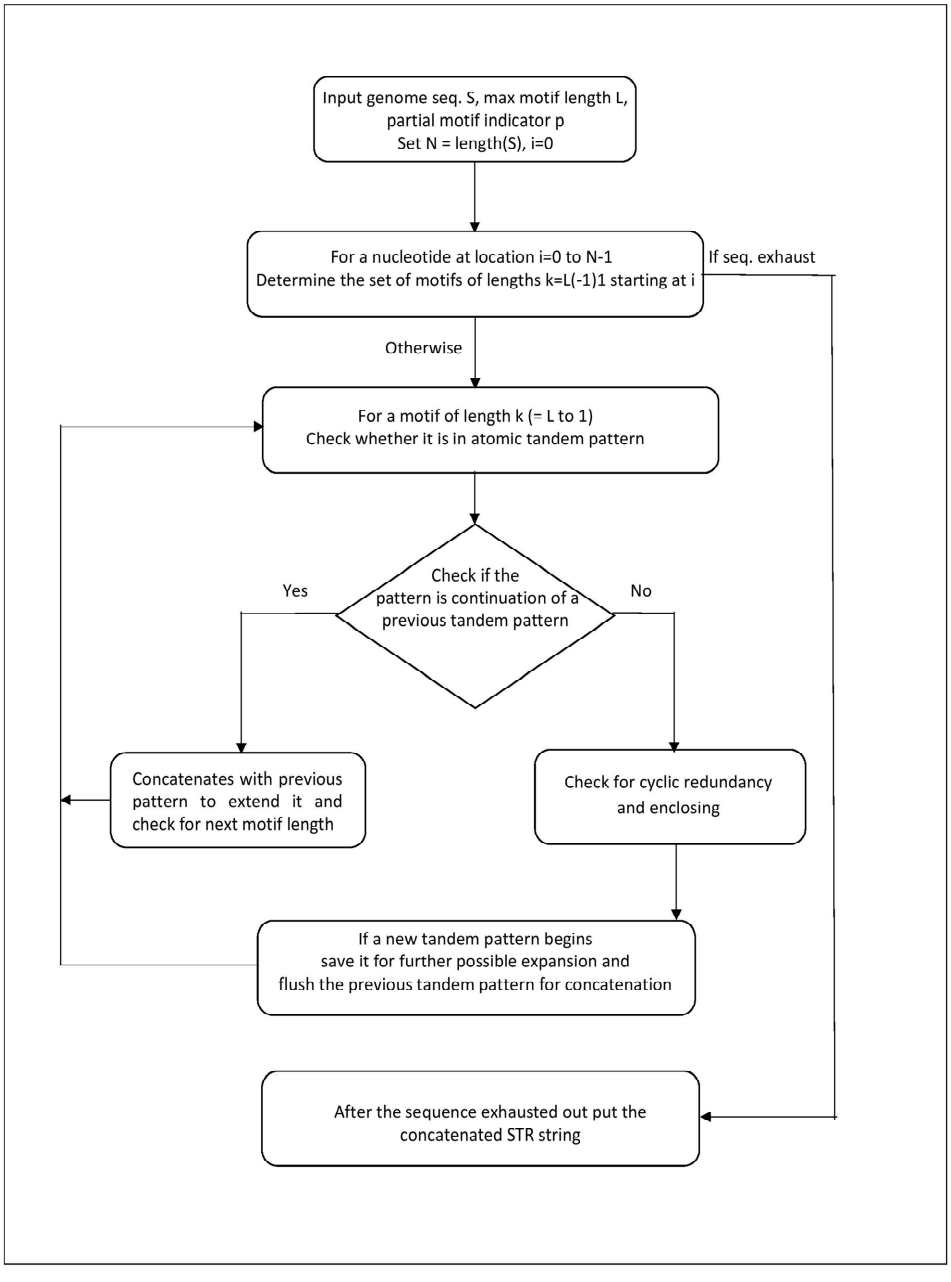
$$\left.\begin{array}{c} k' < k \\ \text{and } i' < i \\ \text{and } \left(i' + \lambda_{k'}^{j'}\right) < \left(i + \lambda_k^j\right) \end{array}\right\} \tag{5}$$

- *Partial repeats:* if any proper prefix of seed ω_j^k occurs in tandem after last occurrence of ω_j^k in $\mathcal{R}_i^\tau\left(\omega_j^k\right)$, then the prefix is also included in the STR as partial motif. We include this provision in 3S as optional. With input parameter $p = 1$, 3S mines STRs with partials, otherwise ($p = 0$, the default value) the complete STRs and no partials.

2.3 Flowchart

We present the flow chart of the method of STR mining and concatenation of the mined STRs in Figure 1.

Figure 1 Flow chart of the proposed STR mining repeat string formation algorithm



2.4 Algorithm

We describe the procedure for seed selection and finding complete structure of STRs in algorithm 3S.

Algorithm 1 STR seed selection (3*S*)

Input: DNA sequence $\mathcal{S}$, maximum motif length L and cut-off value l and value of parameter p

Output: STRs with respective start location, tandem repeat counts and the motif, and the concatenated repeat sequence $\mathcal{RS}$

$N \leftarrow$ length of $\mathcal{S}$

for each nucleotide loci $\hat{i} \in \mathcal{S}$ **do**, $(0 \le \hat{i} \le N - L)$

  for each $k = L$ to 1 **do**

    $\omega_j^k \leftarrow$ motif of length k starting at nucleotide loci $\hat{i}$

    $j \leftarrow$ a unique index of the motif $[0 \le j \le 4^k]$

    detect tandem repeat of ω_j^k using equation (2) where $i_c = \hat{i}$ and $i_p = \hat{i} - k$.

    if $\left(\exists \mathcal{R}_{i-\tau*k}^{\tau}\left(\omega_j^k\right) \in \Gamma\right)$ **them**

      $\tau \leftarrow \tau + 1$

      $\lambda_j^k \leftarrow \tau * \mathrm{k}$

    else

      if [the tandem of ω_j^k is cyclic redundant of $\mathcal{R}_i^{\tau}\left(\omega_{j'}^{k}\right) \in \Gamma$) using equation (3)]

        discard the tandem and continue for next k value

      else

        for each $k' = k + 1$ to L **do**

          if [the tandem of ω_j^k is enclosed within $\mathcal{R}_i^{\tau'}\left(\omega_{j'}^{k'}\right) \in \Gamma$) using equation (4) and equation (5)]

            discard the tandem and continue for next k value

          end if

        end for

        if (p ==1&& $\exists$ *partial motif of*

      the seed of $\mathcal{R}_i^{\tau}\left(\omega_{j'}^{k}\right) \in \Gamma$) **then**

        $\lambda_{j'}^k = \lambda_{j'}^k +$ *length of*

      partial motif

        end if

        if $\lambda_{j'}^k \ge l$ **then**

          Output STR $\mathcal{R}_i^{\tau}\left(\omega_{j'}^{k}\right) \in \Gamma$ and $\mathcal{RS} = \mathcal{RS} \bigcup \mathcal{R}_i^{\tau}\left(\omega_{j'}^{k}\right)$ where eighter $j = j'$ or $j \ne j'$

        end if

        $\Gamma = \Gamma \bigcup \mathcal{R}_{i-k}^{2}\left(\omega_j^k\right) \bigcap \mathcal{R}_i^{\tau}\left(\omega_{j'}^{k}\right)$

        $\Omega = \Omega \bigcap \omega_j^k \bigcup \omega_{j'}^{k}$

31: **end if**
32: **end if**
33: **end for**
34: **end for**
35: Report $\mathcal{RS}$ as the repeat sequence made from only STRs of the given genome $\mathcal{S}$
36: **end algorithm**

2.5 Computational complexity

No repeated searching and rehashing are involved in verifying a motif as seed STR. Checking atomicity and non-cyclic redundancy involve complexity O(1) while the test of non-enclosing involves O($L - k$). For sequence of length N and L number of motifs at each nucleotide, the total complexity becomes (O(1) + O($L - k$))*O(N*L). As $L = 6$ for STRs and $N >> 6$, the total complexity tends to O(N).

3 Experiments

We conduct our experimental studies in two stages: first, we assess the effectiveness of the proposed STR mining algorithm 3S, and then we assert the usefulness of the STR-created repeat sequences for genome comparison. Accuracy, efficiency and scalability are employed to compare the performance of 3S in STR mining tasks with Kmer-SSR and PERF using their implementations available at https://github.com/ridgelab/Kmer-SSR and https://github.com/RKM lab/perf, respectively. When compared to Kmer-SSR, PERF reports STRs along with partial motifs. We compare the outcomes in light of the fact that 3S can extract STRs in two ways, with and without partial motifs. Finally, after achieving efficient and accurate mining of STRs we proceed to conduct STR based phylogeny reconstruction and taxa identification of eukaryotic species. The ultimate goal of the second stage of experiments is to determine whether a genome's STRs can accurately represent a genomic sequence to the extent where only the STRs can be used to infer the right phylogeny and correct identification of genomic taxa. We perform the experiments on DNA sequences of several chromosomes and whole genome sequences (both reference and assembled) of different organisms, publicly available in https://www.ncbi.nlm.nih.gov/assembly/organism/. All experiments are conducted by using a modest computer with Intel(R) Core(TM) i5-5200U CPU @ 2.20 GHz with 4GB RAM and 1TB HDD.

3.1 Comparative performance of STR mining

- *Accuracy:* Table 1 shows exactly equal numbers of STRs extracted by 3S (with and without partial motifs), Kmer-SSR and PERF from human chromosome 1. For in-depth comparison on the accuracy of mined STRs, we provide start positions (Supplementary_Table_S1.xlsx), repeat counts (Supplementary_Table_S2.xlsx) and the motif combination itself (Supplementary_Table_S3.xlsx). Supplementary_Table_S4.xlsx contains the total STR counts on all other human chromosomes. The fact that human chromosome 1 has the most STRs and chromosomal Y has the fewest is an

interesting observation. Results obtained by 3S in all the experiments match exactly with the respective results by Kmer-SSR and PERF. All these confirm correctness of the proposed seed selection approach. Supplementary tables are available in https://github.com/STRHunter/3S.

Table 1 STR counts from human chromosome 1 with cut-off value 12

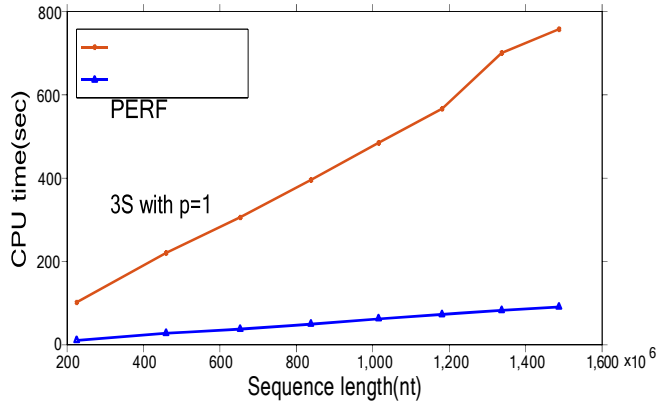
Motif size	*Methods*			
	Kmer-SSR	*3S with p = 0*	*PERF*	*3S with p = 1*
6	146,655	146,665	146,665	146,665
5	13,116	13,116	59,284	59,284
4	46,636	46,636	46,636	46,636
3	15,817	15,817	15,817	15,817
2	25,207	25,207	25,207	25,207
1	57,335	57,335	57,335	57,335
All	304,766	304,766	350,932	350,932

- *Efficiency:* we evaluate computing efficiency and memory requirement of 3S on all the human chromosomes and compare with Kmer-SSR and PERF (Supplementary Table 5). 3S is found significantly efficient over PERF and Kmer-SSR. Considering overall performance on all human chromosomes, 3S takes one fifth of CPU time by PERF and one 29th by Kmer-SSR by requiring less than 1% of their average processing memory (excluding the storage for sequence). It is significant to note that CPU time is invariant against cut-off length for both Kmer-SSR and 3S (Supplementary Table 6). Supplementary tables are available in https://github.com/STRHunter/3S.
- *Scalability:* a method may be efficient on moderately small amount of data, but its real test of proficiency is how well it performs on large volumes of data or on long sequences. We conduct a study on scalability of computation against sequence length. For this experiment, we utilised concatenated human chromosomes to generate long sequences that were arranged in increasing order. The maximum sequence length employed was 1.6 Gbp (gigabase pairs). PERF and Kmer-SSR along with 3S were applied on those set of sequences to mine STRs for comparing CPU time and processing memory. Further a separate experiment was conducted on the sequence length 1.6 Gbp but this time with increasing cut-off values [Figure 2(b)]. It is apparent in Figures 1 and 2 that 3S is exceedingly well and the best performer, both on CPU time, memory and in mining STRs of any cut-off length.

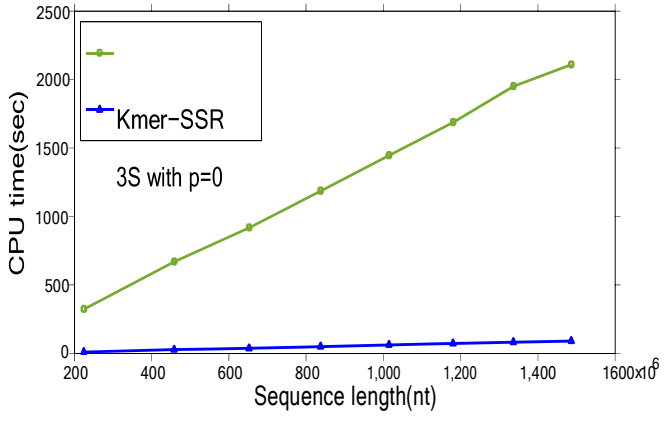
3.2 Mining STRs on whole genomes

Excellent scalability makes STR mining by 3S from whole genome sequences almost instant. We consider a total of eight whole genome reference sequences of evolutionary higher organism and five assembled human genomes for the study. We present the complete result in Table 2. In all the cases, CPU time is within 200 seconds and processing memory is limited to only 4 MB. 3S is thus very useful in extracting long STRs from whole genomes sequences.

Figure 2 Scalability on CPU time against sequence length (a) with PERF, (b) with Kmer-SSR (see online version for colours)



(a)

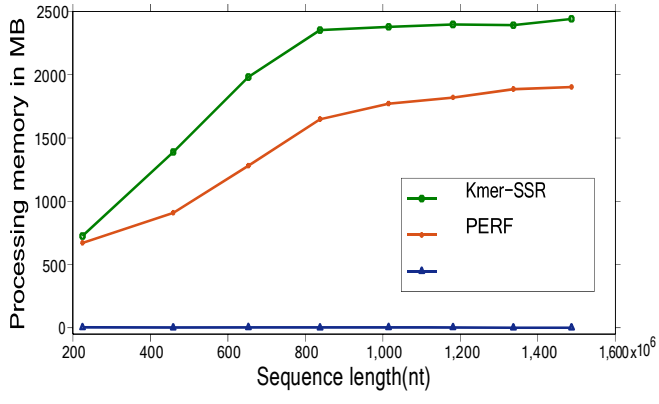


(b)

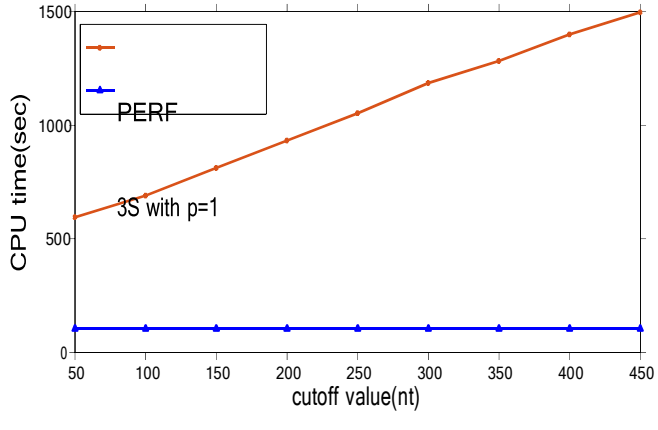
3.3 STR-based phylogeny reconstruction

The development of whole genome sequencing technology has made it possible to swiftly and affordably sequence larger genomes, but this has created a computational challenge in efficiently comparing such massive and numerous data. Sequence alignment techniques used in the past became inappropriate and impractical. It spurs the development of numerous alignment-free sequence analysis tools and techniques (Zielezinski et al., 2017, 2019). K-mer statistics is the main alternative among these techniques, but picking the best k is essential for the greatest feature extraction. Additionally, the length of optimal k is also becoming larger for complete genome sequences of evolutionary higher organisms, making it extremely difficult to compute feature vectors using k-mer frequency statistics. Thus, the need for a method that is suitable and scalable for comparing the whole genome sequence of evolutionary higher species persists.

Figure 3 (a) Scalability on processing memory against sequence length and (b) Scalability on CPU time against cut-off length (see online version for colours)



(a)



(b)

In Mitra et al. (2020), the authors attempted to devise a method, pattern extraction though entropy retrieval (PEER) that employ the entropy of successive intervals (or waits) of optimal length k-mers of the sequence for feature extraction. It transforms a sequence into a vector of wait entropies of optimal k-mers. Distance between a pair of sequences amounts to the Euclidean.

Distance between their wait vectors. It can also determine optimal value of $k(K_{opt})$ using length of the given sequence N and cardinality (β) of its alphabet(a, c, g and t), as

$$K_{opt} = \left\lceil \frac{\ln(N-1)}{\ln \beta} \right\rceil \tag{6}$$

Even if PEER proves to be more effective at reconstructing phylogeny than seven other cutting-edge alignment-free methods, K_{opt} becomes 16-nt for whole genomes of higher organisms. Due to this, a high-dimensional (4^{16}) feature vector is produced, which makes it computationally difficult for machines with limited resources and causes the feature vector to become sparse. Interestingly, 3S can mine and generate a STR repeat sequence

by concatenating all STRs from lengthy genomes of higher organism just in 200 seconds while only using 4MB of memory for STR mining calculations (Table 2). We note that the repeat sequence lengths for all the genomes examined in this analysis range from 15.6 MB to 17.7 MB of nucleotides. The repeat sequence lengths are less than 1% of the length of the whole genomes (≈ 3gbp), and as a result, the optimal k becomes 10 nt rather than 16nt for full genomes. This optimal k length reduction offers high computational preference, making quick comparison of the genomes plausible.

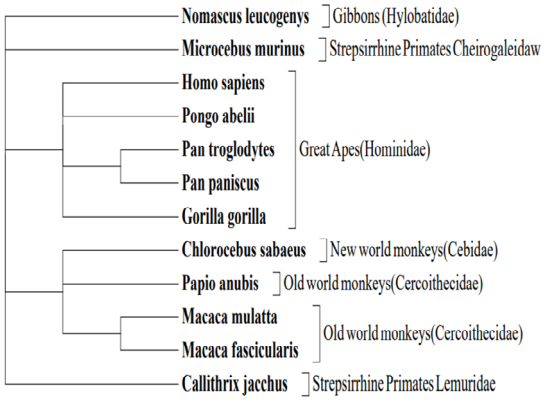
- *Dataset for phylogeny reconstruction:* in order to study precision level and effectiveness of the approach towards STR-based phylogeny reconstruction, we consider three sets of animal genomes in increasing order of diversity. The first one contains 12 genome sequences (accession numbers are in Supplementary Table 7) from a single biological order (primate) and hence they are closely related. The second one contains ten genomes (accession numbers are in Supplementary Table 8) from two biological taxa, mammal and bird. Finally, we consider a total of 100 whole genomes (accession numbers are in Supplementary Table 9) of animals from different taxa, insects, birds, fishes, amphibians and mammals.
- *Phylogeny reconstruction for 12 primates:* With the group of 12 primates, we begin our phylogeny reconstruction experiments. By using the proposed STR mining method 3S, we extract all the STRs from a genome and create a repeat sequence for that genome by concatenating them. We calculate PEER vectors for each of those primates' genomes at $k = 10$ nt from their STR-generated repeat sequences, and from the pair-wise PEER distances between the PEER vectors; we derive the PEER distance matrix. Figure 4(b) depicts the phylogenetic tree. The grouping shown in the tree closely resembles the matching taxa listed in the NCBI taxonomy [Figure 4(a)]. The ability of STR-created repeat sequences to successfully group closely related species that belong to the same biological order is demonstrated by the accuracy of phylogeny reconstruction (primate).
- *Phylogeny reconstruction for ten genomes:* we proceed to check if the same approach is well applicable in grouping of species from different biological orders.

  We take up the set of ten species of the second set for phylogeny reconstruction. It is apparent from Figures (5(a) and 5(b) that 3S and PEER can in effect is able to distinguish the biological orders, primate, rodent and ferungulates, and groups the species identical with NCBI taxonomy for the same species [Figure 5(a)].
- *Phylogeny reconstruction for 100 genomes:* we consider 100 genomes of eukaryotic species from five different animal taxa, insects, birds, fishes, amphibians and mammals. We transform the sequences into corresponding STR-created repeat sequences followed by construction of PEER vectors at $k = 10$ nt and obtain the PEER distance matrix. The phylogeny tree is shown in Figure 5. The tree shows 100% grouping with NCBI reported taxa for all the 100 genomes. This clearly indicates correctness of the proposed approach towards phylogeny reconstruction.

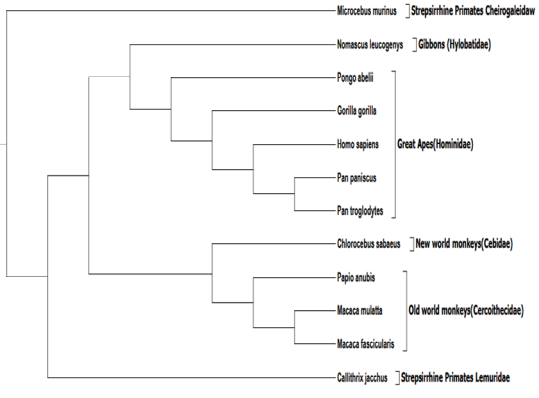
Table 2 STR counts at cut-off ranges on both reference and assembled whole genome sequences

Species (sequence length in MB)	*Large STR (cut-off>= 350 nt)*			*Medium STR (150 nt <= cut-off < 350 nt)*			*Short STR (12 nt <= cut-off< 150 nt)*			*Memory requirement*	
	No. of STR	*Avg. STR length (nt)*	*CPU time (sec)*	*No. of STR*	*Avg. STR length (nt)*	*CPU time (sec)*	*No. of STR*	*Avg. STR length (nt)*	*CPU time (sec)*	*Total memory required in MB*	*Processing memory in MB*
Reference genomes											
C elegans (100.27 MB)	2	496	5	33	187	5	94,697	12	5	104.256	3.986
Drosophila (137.05 MB)	12	619	8	26	240	8	223,123	13	8	140.675	3.625
Canis familaris (2,317.59 MB)	2	418	158	55	196	158	3,877,732	15	157	2,321.486	3.896
Bos taurus (2,640.16 MB)	2	411	179	14	197	180	2,850,953	13	178	2,644.146	3.986
Mus musculus (2,647.52 MB)	57	592	178	1058	186	177	4,062,928	17	178	2,651.416	3.896
Macaca mulata (2,763.46 MB)	2	545	187	29	205	187	3,267,234	14	187	2,767.322	3.862
Pan todro (2,803.62 MB)	1	443	180	33	205	179	3,190,500	14	181	2,807.587	3.967
Homo sapiens (2,937.63 MB)	8	558	196	41	206	199	3,211,310	14	196	2,941.564	3.934
Assembled human genomes											
NA12878 (2,798.04 MB)	2	368	191	39	195	190	3,213,613	14	193	2,801.974	3.934
HG00514 (2,805.67 MB)	8	489	187	75	210	190	3,192,268	14	195	2,809.615	3.945
NA19240 (2,815.57 MB)	6	578	188	47	201	191	3,221,358	14	195	2,819.238	3.668
CHM1.1 (2,827.65 MB)	1	465	188	30	197	191	3,191,074	14	196	2,831.584	3.934
HG00733 (2,864.15 MB)	1	552	192	72	221	190	3,240,565	14	191	2,868.117	3.967

Figure 4 Phylogeny reconstruction of 12 primates dataset, (a) NCBI taxonomy, (b) proposed tree

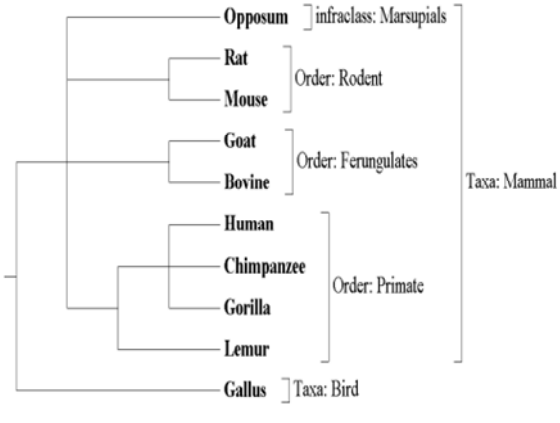


(a)



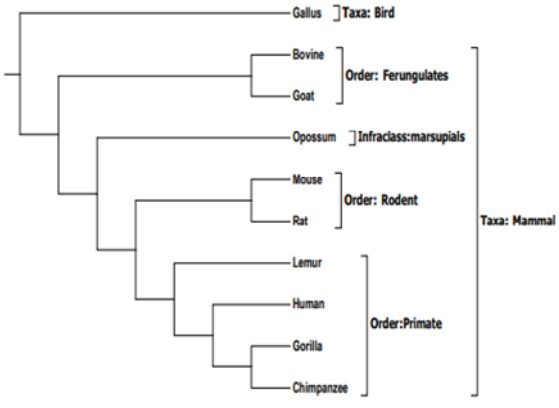
(b)

Figure 5 Phylogeny reconstruction of ten species dataset, (a) NCBI taxonomy, (b) proposed tree



(a)

Figure 5 Phylogeny reconstruction of ten species dataset, (a) NCBI taxonomy, (b) proposed tree (continued)



(b)

3.4 Taxa identification

The identification of taxa plays a crucial role in bioinformatics as it aids in annotating newly sequenced genomes. To achieve this, we utilised the repeat string created by STRs for taxa identification after successfully grouping genomes of various species. Our positive dataset consisted of 113 genomes from mammalian taxa, while our negative sample comprised 64 genomes from fish taxa and 56 genomes from bird taxa. This approach resulted in a nearly balanced dataset, allowing for effective analysis and classification.

3.4.1 One dimensional convolutional neural network classifier

As a classifier we employed 1D convolutional neural network, which can be describe as: Let X be the input sequence of length N, and let Y be the output sequence of length M. We can represent the input sequence X as a matrix of size N*C, where C is the number of input channels. For example, if X is a monochrome audio signal, then $C = 1$, whereas if X is a multichannel audio signal, then C is the number of audio channels.

Let K be the number of filters in the convolutional layer, and let F be the filter size. Each filter is represented as a weight matrix of size F*C. The output of the convolutional layer is a feature map of size

$$(N - F + 1) * K \tag{7}$$

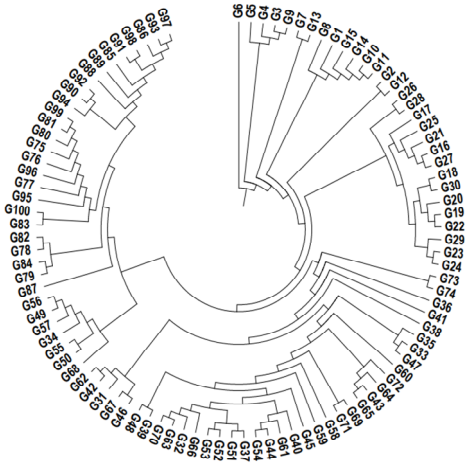
Let A be the activation function applied to the output of the convolutional layer. The output of the activation layer is a feature map of size $(N - F + 1)$*K.

Let P be the pooling operation applied to the output of the activation layer. The output of the pooling layer is a feature map of size

$$(N - F + 1) / S * K \tag{8}$$

where S is the stride of the pooling operation.

Figure 6 Phylogeny tree of 100 eukaryotic animals from complete genomes using proposed approach



Note: The NCBI taxonomic groups for the sequences are as follows: insects: G1-G15, birds: G16-G30, fishes: G31-S72, amphibians: G73-G74 and mammals: G75-G100.

Let D be the dropout fraction applied to the output of the pooling layer. Let H be the number of neurons in the fully-connected layer. The output of the fully-connected layer is a vector of size H. Let O be the output layer, which applies a softmax function to the output of the fully-connected layer to produce a probability distribution over the classes. The mathematical operations performed by a 1D CNN model can be summarised as follows: Let **x** be an input sequence of length N, represented as a 1D tensor of shape (N, 1).

The convolutional layer applies a set of K filters of length F, represented as a 2D tensor **W** of shape (F, K), to the input sequence **x** to generate a set of K feature maps $\mathbf{y}_k$ of shape ($N - F + 1$, 1) as follows:

$$y_k[i] = (\mathbf{W}[:, k] * \mathbf{x}[i : i + F - 1]) + b_k, \tag{9}$$

where * denotes the dot product operation, b_k is a scalar bias term for the k^{th} filter, and i ranges from 0 to $N - F$. This operation can be efficiently implemented using matrix multiplication as follows:

$$Y = X \cdot W + B, \tag{10}$$

where **X** is a matrix of shape ($N - F + 1$, F), representing all possible local regions of the input sequence, and **B** is a vector of shape (1, K), representing all the bias terms. The output **Y** is a matrix of shape ($N - F + 1$, K), representing the feature maps.

The feature maps are passed through an activation function, such as ReLU, to introduce nonlinearity into the model:

$$\mathbf{Y} = \text{relu}(\mathbf{Y}), \tag{11}$$

where relu(.) is the element-wise ReLU function, defined as relu(x) = max(0, x). The pooling layer downsamples the feature maps by aggregating nearby values. A common

pooling operation is max pooling, which takes the maximum value within a window of size P. Mathematically, the max pooling operation can be defined as follows:

$$Z[i, k] = \max(\mathbf{Y}[i \times P : (i+1) \times P, k]), \tag{12}$$

where Z is the output tensor of the pooling layer, and i ranges from 0 to $(N - F)/P$, representing the number of non-overlapping windows of size P in the feature maps.

The output of the pooling layer is flattened into a 1D tensor and passed through one or more fully connected layers to produce the final output of the network. Mathematically, this can be expressed as:

$$\mathbf{h} = \text{flatten}(\mathbf{Z}), \mathbf{u} = \mathbf{W}_f \cdot \mathbf{h} + \mathbf{b}_f, \mathbf{y} = \text{soft}\max(\mathbf{u}), \tag{13}$$

where flatten(.) is a function that converts a tensor into a 1D vector, $\mathbf{W}_f$ is a weight matrix, $\mathbf{b}_f$ is a bias vector, and softmax(.) is a function that normalises the output $\mathbf{u}$ into a probability distribution.

3.4.2 Performance metric

Performance measures are crucial in evaluating the effectiveness of machine learning models. There are various metrics used for this purpose, including accuracy, precision, recall, F1-score, specificity, Matthews correlation coefficient (MCC), and area under the receiver operating characteristic curve (AUC).

Accuracy refers to the proportion of correctly classified instances in a dataset, and is defined as:

$$Accuracy = (TP + TN) / (TP + FP + TN + FN) \tag{14}$$

where true positive (*TP*) is the number of instances correctly classified as positive, true negative (*TN*) is the number of instances correctly classified as negative, false positive (*FP*) is the number of instances wrongly classified as positive, and false negative (*FN*) is the number of instances wrongly classified as negative.

Precision is the proportion of true positives among all instances classified as positive, and is defined as:

$$Precision = TP / (TP + FP) \tag{15}$$

Recall is the proportion of true positives among all actual positive instances, and is defined as:

$$Recall = TP / (TP + FN) \tag{16}$$

F1-score is the harmonic mean of precision and recall, and is defined as:

$$F1\text{-}score = 2 * (Precision * Recall) / (Precision + Recall) \tag{17}$$

Specificity refers to the proportion of true negatives among all actual negative instances, and is defined as:

$$Specificity = TN / (TN + FP) \tag{18}$$

MCC is a correlation coefficient between predicted and actual classifications that ranges from –1 to 1, where 1 indicates perfect agreement and –1 indicates perfect disagreement, and is defined as:

$$MCC = (TP * TN - FP * FN) / sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)) \quad (19)$$

Recursive feature elimination (RFE) is a popular technique used in machine learning to select the most important features in a dataset. One of the main reasons to use RFE is to improve the performance of a machine learning model by reducing the number of input features. This is important because using too many features can lead to overfitting, where the model becomes too complex and fails to generalise to new, unseen data. By selecting only the most important features, RFE can improve the model's accuracy, reduce its computational cost, and make it more interpretable. Additionally, RFE can also be used to gain insights into the underlying relationships between the features and the target variable, which can be valuable for understanding the problem domain and developing better models.

Table 3 Effect of top k features selection using RFE on the taxa identification

Top k features	*Accuracy*	*Precision*	*Recall*	*F1-score*	*Specificity*	*MCC*
10,000	0.9400	0,9276	0.9501	0.9332	0.9234	0.8507
20,000	0.9407	0.9290	0.9589	0.9349	0.9278	0.8501
30,000	0.9503	0.9345	0.9600	0.9456	0.9356	0.8601
40,000	0.9584	0.9456	0.9656	0.9499	0.9399	0.8628
50,000	0.9601	0.9545	0.9667	0.9601	0.9456	0.8539
100,000	0.9600	0.9723	0.9701	0.9638	0.9567	0.8567
200,000	0.9519	0.9560	0.9677	0.9756	0.9500	0.8500
Using all 1,048,576	0.9013	0.8790	0.9129	0.9201	0.9134	0.8490

Table 3 illustrates the outcomes of the classification process. The table clearly demonstrates that not all features are essential for identifying taxa. Through the implementation of RFE, we achieved remarkable results, including an accuracy of 0.9601, precision of 0.9545, recall of 0.9667, F1-score of 0.9601, specificity of 0.9456, and MCC of 0.8539. Surprisingly, these excellent results were obtained using only 50,000 features. This is particularly intriguing considering that the proposed method managed to reduce the number of features from 416 to 410, and with RFE, the feature count further diminished to just 50,000. Indeed, this achievement enables a remarkably efficient representation of the feature space for the vast genome of mammalian taxa, spanning an impressive length of 3Gbp.

4 Conclusions

We discover that STRs originates from those atomic motifs that do not generate both cyclic-redundant and enclosing TRs. We designate such motifs as seeds of STRs and devise a novel algorithm, STR seed selection (3S), for efficient mining of STRs of any cut-off length from genome-wide sequences. The algorithm works at linear time with excellent scalability against sequence length. Moreover, the method itself makes

computation time invariant against cut-off length of STRs. Series of tests on human chromosomes to whole genomes of complex organism confirm that 3S is 100% accurate and proficient. The reasons for its exceedingly well computing efficiency are single scan of the sequence, the elegant way in identifying the STRs and no post-processing or filtering. Requirement of tiny memory, fast computing capability and high level of scalability establish that 3S has the potential to be embedded in devices to initiate new dimension in automating STR based biosequence analysis.

Furthermore, a fascinating discovery was made as a result of the phylogeny reconstruction experiments. We note that the STRs of the genome are capable of precisely representing a genomic sequence to the extent where solely the STRs may be utilised to deduce the correct phylogeny and identify the genomic taxa. Therefore, all of the tests show that STRs are useful for reconstructing the phylogeny of higher organisms, which would not have been conceivable without accurate and effective STR mining. Thus, 3S offers a cost-effective technique to compare lengthy genomes and can be a key tool in large-scale genomic studies. The utilisation of STR variation has proven to be highly effective in taxa identification, yielding notable accuracy. Furthermore, it has been observed that a mere 50,000 features possess the capability to represent the colossal genomes of mammals, measuring a staggering length of 3 Gbp. It is important to note, however, that the proposed methods solely mine exact motif STRs, with no current provision for inexact motif or motif fuzzy matching. Addressing this challenging task represents a promising avenue for future research in this field.

References

Avvaru, A.K., Sowpati, D.T. and Mishra, R.K. (2017) 'PERF: an exhaustive algorithm for ultra-fast and efficient identification of microsatellites from large DNA sequences', *Bioinformatics*, Vol. 34, No. 6, pp.943–948, DOI: 10.1093/bioinformatics/btx721.

Chen, J., Li, F., Wang, M., Li, J., Marquez-Lago, T.T., Leier, A., Revote, J., Li, S., Liu, Q. and Song, J. (2022) 'BigFiRSt: a software program using big data technique for mining simple sequence repeats from large-scale sequencing data', *Front Big Data*, 18 Jan., Vol. 4, p.727216, DOI: 10.3389/fdata.2021.727216. PMID: 35118375; PMCID: PMC8805145.

Do, H.H., Choi, K.P., Preparata, F.P. et al. (2008) 'Spectrum-based de novo repeat detection in genomic sequences', *J. Comput. Biol.*, Vol. 15, No. 5, pp.469–487, DOI: 10.1089/cmb.2008.0013.

Domanic, N.O. and Preparata, F.P. (2007) 'A novel approach to the detection of genomic approximate tandem repeats in the Levenshtein metric', *J. Comput. Biol.*, Vol. 14, No. 7, pp.873–891, https://doi.org/10.1089/cmb.2007.0018.

Gou, X., Shi, H., Yu, S., Wang, Z., Li, C., Liu, S., Ma, J., Chen, G., Liu, T. and Liu, Y. (2020) 'SSRMMD: a rapid and accurate algorithm for mining SSR feature loci and candidate polymorphic SSRs based on assembled sequences', *Front Genet.*, 27 Jul.; Vol. 11, p.706, DOI: 10.3389/fgene.2020.00706, PMID: 32849772; PMCID: PMC7398111.

Greene, E., Mahishi, L., Entezam, A., Kumari, D. and Usdin, K. (2007) 'Repeat-induced epigenetic changes in intron 1 of the frataxin gene and its consequences in Friedreich ataxia', *Nucleic Acids Research*, Vol. 35, No. 10, pp.3383–90, Epub 2007 May 3, DOI: 10.1093/nar/gkm271/.

Guo, Y., Chen, C., Xie, T., Cui, W., Meng, H., Jin, X. and Zhu, B. (2018) 'Forensic efficiency estimate and phylogenetic analysis for Chinese Kyrgyz ethnic group revealed by a panel of 21 short tandem repeats', *Royal Society Open Science*, 13 Jun., Vol. 5, No. 6, p.172089, DOI: 10.1098/rsos.172089.

Ishiura, H., Doi, K., Mitsui, J. et al. (2018) 'Expansions of intronic TTTCA and TTTTA repeats in benign adult familial myoclonic epilepsy', *Nat Genet.*, Vol. 50, No. 4, pp.581–590, DOI: 10.1038/s41588-018-0067-2.

Kashi, Y., King, D. and Soller, M. (1997) 'Simple sequence repeats as a source of quantitative genetic variation', *Trends in Genetics*, Feb., Vol. 13, No. 2, pp.74–78, DOI: 10.1016/s0168-9525(97)01008-1.

Kumar, R.P., Krishnan, J., Singh, N.P., Singh, L. and Mishra, R.K. (2013) 'GATA simple sequence repeats function as enhancer blocker boundaries', *Nature Communications*, Article No. 1844, Vol. 4, No. 1.

Lewis, D.H., Jarvis, D.E. and Maughan, P.J. (2020) 'SSRgenotyper: a simple sequence repeat genotyping application for whole-genome resequencing and reduced representational sequencing projects', *Appl Plant Sci.*, 3 Dec., Vol. 8, No. 12, p.e11402, DOI: 10.1002/aps3.11402. PMID: 33344093; PMCID: PMC7742204.

Li, Z., Zhang, J., Zhang, H., Lin, Z. and Ye, J. (2018) 'Genetic polymorphisms in 18 autosomal STR loci in the Tibetan population living in Tibet Chamdo, South-west China', *International Journal of Legal Medicine*, May, Vol. 132, No. 3, pp.733–734, DOI: 10.1007/s00414-017-1740-1.

Lim, K.G., Kwoh, C.K., Hsu, L.Y. et al. (2012) 'Review of tandem repeat search tools: a systematic approach to evaluating algorithmic performance', *Brief Bioinform.*, Vol. 14, No. 1, pp.67–81, DOI: 10.1093/bib/bbs023.

Mitra, U., Bhattacharyya, B. and Mukhopadhyay, T. (2020) 'PEER: a direct method for biosequence pattern mining through waits of optimal k-mers', *Information Sciences*, Vol. 517, No. 517, pp.393–414.

Ott, J., Wang, J. and Leal, S.M. (2015) 'Genetic linkage analysis in the age of whole-genome sequencing', *Nat. Rev. Genet.*, Vol. 16, No. 5, pp.275–284, DOI: 10.1038/nrg3908.

Pickett, B.D., Miller, J.B. and Ridge, P.G. (2017) 'Kmer-SSR: a fast and exhaustive SSR search algorithm', *Bioinformatics*, Vol. 33, No. 24, pp.3922–3928, DOI: 10.1093/bioinfor-matics/btx538.

Usdin, K. (2008) 'The biological effects of simple tandem repeats: lessons from the repeat expansion diseases', *Genome Res.*, Vol. 18, No. 7, pp.1011–1019.

Wirawan, A., Kwoh, C.K., Hsu, L.Y. and Koh, T.H. (2010) 'INVERTER: INtegrated Variable numbER tandem rEpeat findeR', in Chan, J.H., Ong, YS. and Cho, SB. (Eds.): *Computational Systems-Biology and Bioinformatics. CSBio 2010. Communications in Computer and Information Science*, Springer, Berlin, Heidelberg, Vol. 115.

Zielezinski, A., Girgis, H.Z., Bernard, G., Leimeister, C.A., Tang, K., Dencker, T. and Karlowski, W.M. (2019) 'Benchmarking of alignment-free sequence comparison methods', *Genome Biology*, Vol. 20, No. 1, pp.1–18.

Zielezinski, A., Vinga, S., Almeida, J. and Karlowski, W.M. (2017) 'Alignment-free sequence comparison: benefits, applications, and tools', *Genome Biology*, Vol. 18, No. 1, pp.1–17.

Zietkiewicz, E., Rafalski, A. and Labuda, D. (1994) 'Genome fingerprinting by simple sequence repeat (Ssr)-anchored polymerase chain-reaction amplification', *Genomics*, 15 Mar., Vol. 20, No. 2, pp.176–183, DOI: 10.1006/geno.1994.1151.