OXFORD

Briefings in Bioinformatics, 2025, **26(1)**, bbae637

https://doi.org/10.1093/bib/bbae637

Problem Solving Protocol

STRsensor: a computationally efficient method for STR allele-typing from massively parallel sequencing data

Xiaolong Zhang[1,2,‡], Xianchao Ji[1,2,‡], Lingxiang Wang[3,‡], Lianjiang Chi[1], Chengtao Li[4,*], Shaoqing Wen[3,*], Hua Chen[12,5,*]

[1]Beijing Institute of Genomics, Chinese Academy of Sciences and China National Center for Bioinformation, Beijing 100101, China
[2]School of Future Technology, University of Chinese Academy of Sciences, Beijing 100049, China
[3]Institute of Archaeological Science, Fudan University, Shanghai 200032, China
[4]Shanghai Medical College, Fudan University, Shanghai 200032, China
[5]CAS Center for Excellence in Animal Evolution and Genetics, Chinese Academy of Sciences, Kunming 650023, China

*Corresponding authors. Hua Chen, E-mail: chenh@big.ac.cn; Shaoqing Wen, E-mail: wenshaoqing@fudan.edu.cn; Chengtao Li, E-mail: lichengtao@fudan.edu.cn
‡Xiaolong Zhang, Xianchao Ji and Lingxiang Wang contributed equally to the work.

Abstract

Short tandem repeats (STRs) represent one of the most polymorphic variations in the human genome, finding extensive applications in forensics, population genetics and medical genetics. In contrast to the traditional capillary electrophoresis (CE) method, genotyping STRs using massive parallel sequencing technology offers enhanced sensitivity and accuracy. However, current methods are mainly designed for target sequencing with higher coverage for a specific STR locus, thereby constraining the utility of STRs in low- and medium-coverage whole genome sequencing (WGS) data. Here, we introduce STRsensor, a method designed to type STR alleles in low-coverage WGS data and target sequencing data, achieving a significant high detection ratio and accuracy. STRsensor employs two methods for STR allele-typing: the Kmers-based method and the CIGAR-based method. Furthermore, by incorporating a model for PCR stutters, STRsensor greatly enhances the accuracy of STR allele typing. With simulation data, we demonstrate that STRsensor achieves a detection ratio of 100% and an accuracy of 99.37% for a 30× WGS data, outperforming the existing methods, such as STRait Razor, STRinNGS, and HipSTR. When applied to real target sequencing data from 687 individuals, STRsensor achieves a detection ratio of 99.64% and an accuracy of 99.99%. Moreover, STRsensor is a computationally efficient method that runs 79 times faster than HipSTR and 10 000 times faster than STRinNGS. STRsensor is freely available on GitHub: https://github.com/ChenHuaLab/STRsensor.

Keywords: STR; allele-typing; NGS; software

Introduction

Short tandem repeats (STRs), known as microsatellites, are DNA loci consisting of multiple repeat units with a length of 2-6 nucleotides [1]. Unlike single nucleotide variations with only two allele types, STRs are highly polymorphic in the human genome. As a major category of biomarkers, STR has extensive applications in medical genetics, population genetics, conservation biology, and forensics [2–4]. STR has been shown to play a key role in more than 65 Mendelian disorders and has the potential to be involved in gene regulation and the development of complex traits [5]. About 17% of human genes contain STR in their open reading frames [6], and many have been reported to affect protein function [7]. In forensic science, STR is used for personal identification [8], paternity testing [9], surname inference [10], etc. The predominant method for STR genotyping is capillary electrophoresis (CE), which has been the 'gold standard' in forensic DNA laboratories for decades [11–13]. The CE-based STR allele typing method is straightforward, cost-effective, and court-accredited. However, its limitation lies in its ability to type only a restricted number of STR loci due to current spectral resolution, making it inefficient for genotyping STRs in mixed and degraded DNA samples [14, 15].

With the development of massively parallel sequencing (MPS) technology, next-generation sequencing (NGS) data can be obtained at lower price and higher efficiency, which is expected to be an alternative to CE [9]. MPS can simultaneously amplify hundreds of thousands of STR loci, largely eliminating the limitation of CE technology on the number of STR loci [16]. Furthermore, the MPS method can detect STR alleles with genomic DNA mass greater than 62pg, which is particularly useful for trace and degraded DNA in forensics [13]. In addition to fully solving different STR alleles, MPS can also detect single nucleotide polymorphisms (SNPs), insertions, and deletions, greatly increasing the available genetic information on STR [17, 18]. However, NGS-based STR typing also has some challenges in that its sensitivity and accuracy can be influenced by various factors, such as sequencing coverage, PCR stutters, flanking sequence homology, etc. Sequencing coverage directly influences STR genotyping methods based on spanning reads which necessitate complete coverage of the STR core region for accurate

Received: April 5, 2024. **Revised:** August 16, 2024. **Accepted:** November 22, 2024

allele extraction and calculation [3, 19, 20]. Another critical factor is PCR stutter noise, which can constitute up to 20% of STR spanning reads (e.g. DYS481) [21], significantly impacting STR allele size determination. Consequently, it is a common goal to address among NGS-based STR genotyping methods [3, 5, 19, 20, 22]. Flanking sequences adjacent to the STR region often mirror the STR allele sequences due to their inclusion of multiple STR repeat units, complicating accurate sequence alignment. Furthermore, homology between flanking sequences of different STR loci further complicates precise STR typing [19]. Some methods mitigate this issue by seeking nonrepetitive sequences adjacent to the STR core region as flanking sequences [5, 23], while others opt for globally unique sequences across all STR loci [1, 19]. However, these strategies can place flanking sequences distant from the STR core, reducing the number of spanning reads and compromising STR typing sensitivity. Therefore, in medical and forensic research, there is a growing demand for STR typing software capable of achieving high detection rates and accuracy, suitable for both low-coverage WGS data and high-coverage target sequencing data.

In recent years, several tools have been developed for STR genotyping from NGS data, including HipSTR [5], STRait Razor [1, 19], STRinNGS v2.0 [20], STRling [24], and ExpansionHunter [25]. STRling can call alleles at both known and novel STR loci by counting k-mers, however, it is not feasible for genotyping known STR loci from single-end NGS data, and erroneously obtains only one allele for the loci. ExpansionHunter is a STR typing method based on sequence graphs, it uses pair-end reads from sequencing data to genotype long STR alleles, which makes it unsuitable for typing STR alleles from single-end sequencing data too. HipSTR adopts a hidden Markov model to realign reads to candidate alleles, facilitating the inference of STR alleles for an individual. However, HipSTR encounters challenges in typing certain STR loci (i.e. DYS19, DYS557) due to repetitive sequences in the flanking regions of those loci. Moreover, for some specific STR loci, such as D2S1338, there will be some base differences between the HipSTR typed allele and the true CE allele. This phenomenon is mainly attributed to the high sequence similarity between the adjacent regions of the flanking region and the STR core region, leading to incorrect matches during HipSTR realignment of the reads to the candidate alleles. STRait Razor [19] is a length-based tool specifically designed to type forensic STR loci. STRait Razor takes a FASTQ file as input, which does not contain genomic positions on the reference genome. Since some STR loci have similar or identical flanking sequences (such as D7S820 and D6S1043), STRait Razor addresses the challenge by designing a genome-wide unique flanking sequence for each target STR locus, making the flanking sequence obtained farther from the STR region. As a result, for short-read WGS data, numerous reads are directly discarded as they are unable to match the relatively far flanking sequence, leading to typing failure. Moreover, STRait Razor solely outputs a set of allele sizes extracted from the read sequence, requiring users to determine the alleles empirically. This manual operation is prone to errors, particularly in low-coverage WGS data. STRinNGS is designed to identify STR loci of the combined DNA index system (CODIS) [20]. It takes the FASTQ or BAM file as input and uses the regular expression-based algorithm 'agrep (TRE agrep) 0.8.0' to extract the flanking sequences adjacent to the STR loci. However, STRinNGS imposes stringent filtering criteria on sequence reads, so that only a few reads can be used for STR typing in low-coverage WGS data, resulting in a low detection ratio. Furthermore, for some STR loci with repetitive sequences in the flanking region, such as DYS456, STRinNGS may cause incorrect regular expression matching, resulting in STR typing errors. Additionally, although STRinNGS supports multi-threaded mode, its global sequence matching strategy makes it rather slow in practice.

Here, we developed STRsensor, a novel method for STR typing applicable to both low-coverage WGS data and target sequencing data. STRsensor is specifically designed to type both the forensic CODIS STR loci and user-specified STR loci. We demonstrate that STRsensor can achieve high detection ratio and accuracy in both low-coverage WGS data and high-coverage target sequencing data. Moreover, the C implementation of the tool significantly enhances its running speed, especially when analyzing a substantial number of samples. Consequently, STRsensor can be considered a useful tool for forensic and medical researchers.

Methods

STRsensor is designed to type STR loci with well-defined genomic positions and flanking regions, including both forensic CODIS STR loci and user-specified STR loci. As mentioned above, the influence of repetitive sequences in the flanking regions poses a significant challenge for existing software in STR typing. To address this challenge, we introduce two STR typing methods: the Compact Idiosyncratic Gapped Alignment Report (CIGAR-based method and the Kmers-based method. The CIGAR-based method can directly obtain the exact location of the STR region by correcting the CIGAR value. Meanwhile, the Kmers-based method can locate the STR region using locally unique Kmer. Our results show that the synergistic use of CIGAR- and Kmers-based methods can greatly improve the detection ratio in low-coverage sequencing data.

The procedures of STRsensor are summarized in the flowchart (Fig. 1). Briefly, the method begins by taking the k-mers of the flanking sequence as the key and the corresponding position of the k-mers in the flanking sequence as the value to construct a 'hash table' (Fig. 2). Subsequently, the STR alleles are extracted from the read sequence through two distinct approaches: the CIGAR-based method and the Kmers-based method [26]. The CIGAR-based method uses the CIGAR value in the BAM file [27] to determine the STR region; the Kmers-based method determines the STR region by remapping the up- and downstream flanking sequences to the reads. After allele extraction, a maximum likelihood method is applied to extracted alleles to assess the impact of polymerase slippage-induced PCR stutter on STR typing [28]. This process allows for the inference of PCR stutter parameters and candidate allele frequencies (Fig. 1B). Finally, the maximum A posteriori (MAP) estimate is used to determine the most likely STR allele from the pool of candidate alleles (Fig. 1B). The details of these steps are shown below.

Flanking sequence indexing

STR alleles are composed of repeating units, and their sizes vary among alleles, so it is not feasible to directly match repeat sequences at the given STR locus. To obtain accurate STR typing results, the algorithm must accurately match the 5' and 3' flanking sequences of the STR locus. To this end, STRsensor introduces a Kmers-based method to match flanking sequences of STR locus. The Kmers-based method is an alignment-free strategy [29] with a lower time complexity and faster execution speed compared to traditional dynamic programming algorithms.

STRsensor retrieves the start and end positions of STR loci from the input configuration file and then extracts the 5' and 3' flanking sequences (25 bp) adjacent to the STR region. Next,

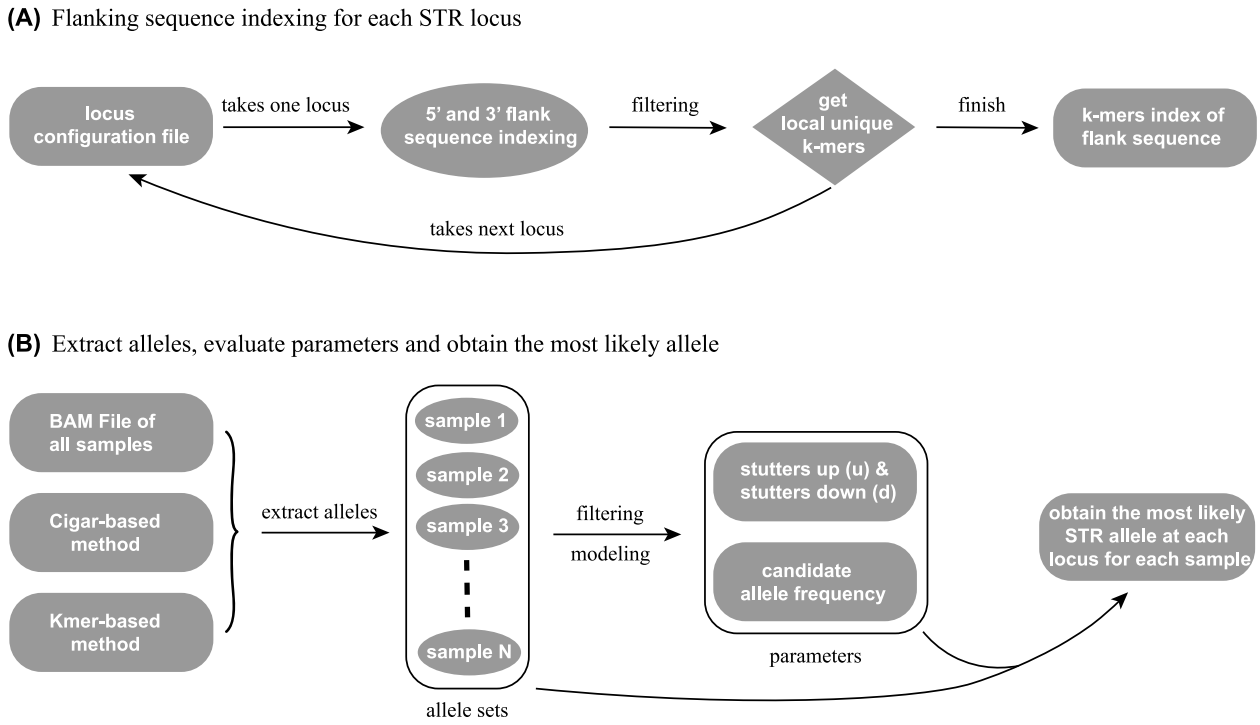

Figure 1. The workflow of STRsensor. (A) The Kmers algorithm is used in flanking sequence indexing for each STR locus. The locally unique k-mers indicate the k-mers sequence that occurred only once on up- and downstream of STR region. (B) The core algorithm of STRsensor, which includes extracting alleles from spanning reads, estimating model parameters with filtered high-quality samples, and obtaining the most likely alleles for each sample with the estimated parameters.

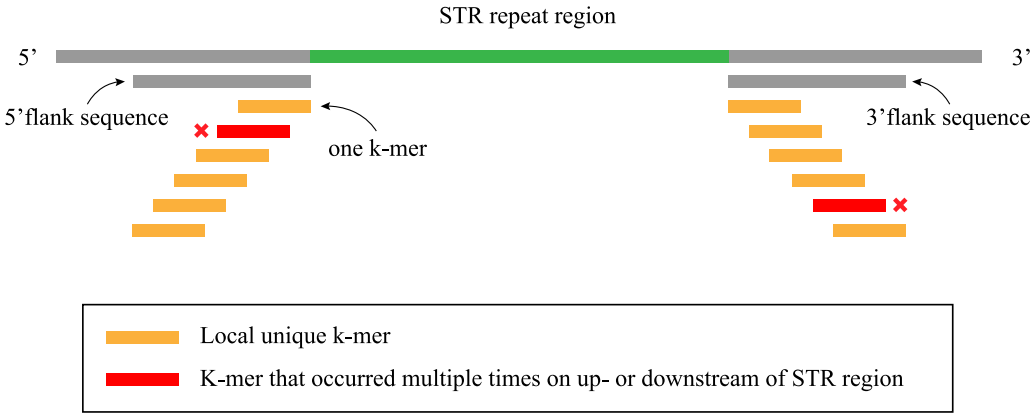

Figure 2. STR flanking sequence indexing based on Kmers method. Each k-mer subsequence differs by one base, and red indicates non-locally unique k-mers, which need to be removed from the subsequent analysis. The locally unique k-mers and theirs corresponding position on the flanking sequences will be recorded into a 'hash-table'.

the algorithm dynamically adjusts the k-mer length (starting with k=8) to determine the minimum k that ensures obtaining locally unique k-mers for STR loci. Then, STRsensor generates all potential 1-base shift k-mers sequences for the extracted flanking sequences (Fig. 2). Since most flanking sequences exhibit repetitive structures, using original k-mers with multiple positions on the flanks may lead to incorrect allele types due to misalignment. To improve the precision of flanking sequence matching, original k-mers with multiple matches within 200 bp upstream and downstream of the STR region are directly discarded (red k-mers in Fig. 2). The K-mers that are retained, termed locally unique k-mers, along with their corresponding index, are then recorded in a 'hash table' to efficiently match their flanking sequences in subsequent processing.

Extracting alleles from STR region

STRsensor takes the BAM file rather than the FASTQ file as input, because the BAM file stores the aligned position and the CIGAR value of each sequencing read, enabling more efficient and accurate allele typing. STRsensor employs two approaches, namely the CIGAR- and Kmers-based algorithms, to extract alleles from sequences between the 5' and 3' flanking regions. STRsensor prefers to extract alleles from read sequences using the CIGAR-based algorithm. However, if the CIGAR-based method fails to successfully extract alleles, the sequence will be processed again using the Kmer-based method for allele extraction. The CIGAR-based algorithm utilizes the CIGAR value of the alignments to determine the location of the STR allele, while the Kmers-based algorithm, also known as k-mer mapping and extension [26], is

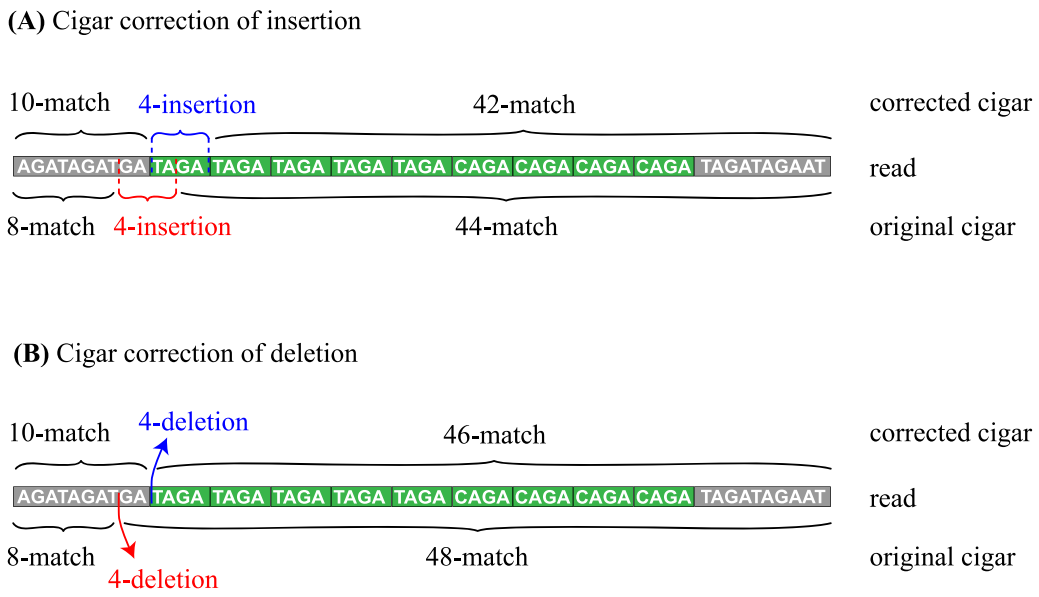

Figure 3. Wrong CIGAR value induced by insertion and deletion. The gray and green rectangles represent STR flanking sequence and allele sequence, respectively. (A) and (B) show the wrong CIGAR value induced by insertion and deletion. The red indicates the wrong insertion or deletion position, while blue indicates the corrected insertion or deletion position.

performed to realign the flanking sequence to the sequencing read to obtain the STR allele.

The CIGAR-based method commences by fetching the reads covering the given STR region. For each read, the CIGAR value and genomic position of the alignment are extracted to calculate the position of the STR region in the read sequence. The 5' and 3' flanks of the read adjacent to the STR region are then compared to the flanking sequences of STR locus defined in the configuration file to obtain the STR allele. Differences in allele size between the individual and reference genomes represent insertions and deletions introduced by alignment tools. Since the STR alleles consist of repeat sequences, an erroneous CIGAR value for insertions and deletions may be introduced when part of the flanking sequence is similar to the repeat unit sequence (Fig. 3). Therefore, prior to allele extraction, the CIGAR value needs to be corrected through the CIGAR-based method to ensure accurate allele-typing.

However, when the size of the allele is much larger than the corresponding size of the allele in the reference genome, the alignment tool will mark part of the allele sequence as a Softclip to attain a higher alignment score, leading to a entirely incorrect CIGAR value (Fig. 4). This phenomenon becomes more severe in STR loci with higher allelic polymorphisms, such as PentaE, D6S1043, DYS385ab, etc. To increase the detection ratio, the Kmers-based algorithm is adopted as a complementary method to CIGAR-based approach. Initially, the entire read sequence is sliced to acquire all 1-base shift k-mers, which are subsequently assigned to a 'hash table' for matching their flanks. Then, the Hamming distance is calculated to ensure an exact match between the read sequence and the flanking sequence. Ultimately, the STR allele is determined by dividing the STR sequence length by the repeat unit length of the STR locus.

Constructing PCR stutter model

PCR stutter artifacts arise from polymerase slippage during DNA replication [28], introducing a discrepancy between the observed and underlying alleles. PCR stutter has a significant impact on low-coverage sequencing data, as only a few available reads span the entire STR locus, making it difficult to distinguish the real allele from PCR stutter. To address this issue, STRsensor introduces a PCR stutter model for each STR locus to mitigate the effects of PCR stutter on STR typing. Since the vast majority of PCR stutters differ from the real allele by a single repeat unit, the STRsensor adopts a simplified PCR stutter model [5]. The PCR stutter model of STRsensor includes two parameters u and d, corresponding to the probability of stutter of adding and removing one repeat unit, respectively. Therefore, for a given STR locus l, x represents the number of repeat units that differ between the observed allele and the real allele, which can be divided into three cases, namely, the same as the real allele ($x = 0$), one more repeat unit than the real allele ($x = 1$) and one less repeat unit than the real allele ($x = -1$).

$$P(x|u,d) = \begin{cases} 1-u-d, & x = 0, \\ u, & x = 1, \\ d, & x = -1. \end{cases} \quad (1)$$

For a given individual i that has n^i reads spanning the STR locus l, we denote the state of the set of alleles as $X^i = \{x_1^i, x_2^i, \cdots, x_{n^i}^i\}$. Then the likelihood function for the N individuals is

$$L(X^1, X^2, ..., X^N|u,d) = \prod_{i=1}^{N}\prod_{j=1}^{n^i} P(x_j^i|u,d). \quad (2)$$

Based on equation (1), the maximum likelihood function is

$$L(X^1, X^2, ..., X^N|u,d) = u^{m_1} d^{m_2} (1-u-d)^{T-m_1-m_2}, \quad (3)$$

where

$$T = \sum_{i=1}^{N} n^i, \quad (4)$$

with m_1 being the number of alleles that have one more repeat unit than the real allele, and m_2 being the number of alleles that have one less repeat unit than the real allele. The estimates of u and d are simple:

$$\begin{cases} \hat{u} = \frac{m_1}{T} \\ \hat{d} = \frac{m_2}{T} \end{cases} \quad (5)$$

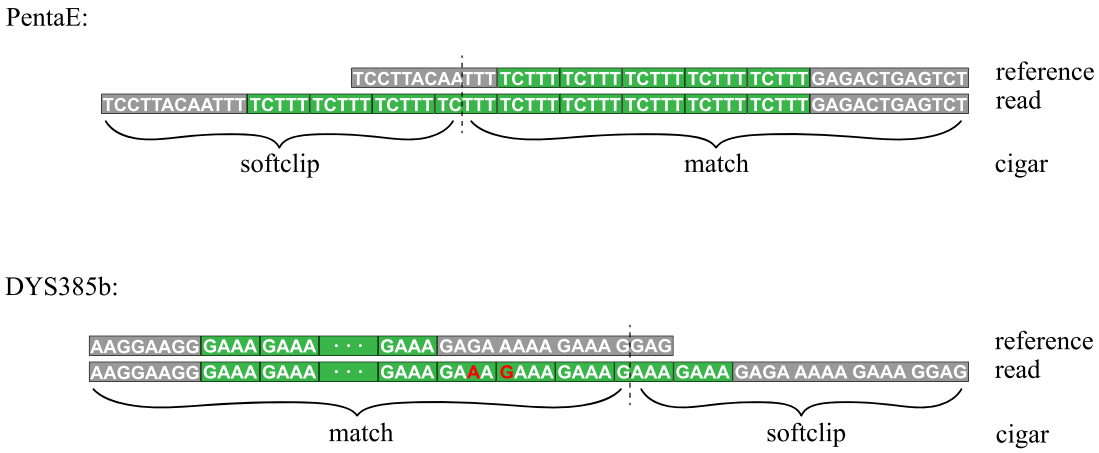

Figure 4. Wrong CIGAR value induced by error alignment. Part of the allele sequence is marked as softclip by the alignment tool, resulting in an incorrect CIGAR value. The green and gray rectangles represent STR repeat units and flanking sequences, respectively.

STRsensor sets many stringent criteria to accurately estimate model parameters for user-specified STR loci. These parameters include rate of stutter being added (u), stutter being removed (d) and the distribution of STR allele frequency (f) in the population. The estimate of parameters u and d must follow the following rules: (1) The number of reads that fully span the STR locus must exceed the given minimum threshold (default: 10). (2) The allele with the highest number of spanning reads must exceed 70% of all spanning reads at the STR locus. To estimate the allele frequency of the user-specified STR locus, STRsensor obtains an initial set of candidate alleles from the input data. The criteria to obtain candidate alleles for the user-specific STR locus are as follows: (1) at least 20 samples are available. (2) Alleles should appear in two or more samples.

Infer the most likely allele

Determining alleles for a given STR locus by simply counting the number of supporting reads for each observed allele can be prone to errors, especially in low-coverage sequencing data. To address this, STRsensor takes into account all potential allele combinations to obtain the most likely allele. With the stutter parameters and candidate allele frequency estimated from the samples, STRsensor generates all possible combinations of alleles from the candidate alleles. Then, for each individual, the method calculates the posterior probability of each allele combination by MAP. The allele with the highest posterior probability will be considered the final typing result. Users also have the option to set a minimum posterior probability threshold, such as 0.99, to improve the accuracy of STR typing. The posterior probability is estimated as follows:

For an individual with n reads spanning the locus l, the corresponding observed allele set is $A = \{a_1, a_2, \ldots, a_n\}$, then the posterior probability of each possible allele $a (a \in A_l$, the STR allele candidates on locus l) can be calculated by MAP:

$$P(allele_l = a|A, u, d) \propto f_a^l \prod_{i=1}^{n} \begin{cases} 1-u-d, & a_i = a, \\ u, & a_i = a+1, \\ d, & a_i = a-1, \\ 0, & a_i = others. \end{cases} \quad (6)$$

Here, the variable f_a^l represents the frequency of allele a at locus l. a_i denotes the i^{th} observed allele, while $a+1$ and $a-1$ denote addition and deletion of one repeat unit for the given allele a, respectively. And '*others*' denotes addition and deletion of two or more repeat units for the given allele a. There are two alleles for autosomal STR loci, therefore, we assume that the two alleles are $j (j \in A_l)$ and $k (k \in A_l)$. Then the joint probability of two alleles can also be calculated using MAP as follows:

$$P(allele_l = (j,k)|A, u, d) \propto f_j^l f_k^l \prod_{i=1}^{n} \sum_{a \in \{j,k\}} \begin{cases} 1-u-d, & a_i = a, \\ u, & a_i = a+1, \\ d, & a_i = a-1, \\ 0, & a_i = others. \end{cases} \quad (7)$$

Results

To evaluate the performance of STRsensor, we compare it with three commonly used tools, namely HipSTR, STRaitRazor and STRinNGS. The datasets used in the benchmarks include simulated WGS data generated by in-house Python script STRSimulatorV4.1.py (details in Supplementary File) and target sequencing data from Fudan University. The performance is measured based on two key criteria: detection ratio (DR) and accuracy (AC) of STR allele typing. For a given STR locus, DR is defined as the ratio of successfully typed samples to the total number of samples, while AC is the ratio of the number of samples with consistent typing alleles with CE to the total number of successfully typed samples. Further details regarding the datasets, benchmarks, and a comprehensive breakdown of detection ratio and accuracy for each STR locus are available in the Supplementary File.

Performance on simulated datasets

We simulated a total of 3000 WGS samples, 10× (1000) and 30× (1000) and 50× (1000), with single-end 150 bp. The simulated data sets consist of 43 STR loci, including 17 autosomal STR loci and 26 Y-STR loci. Then we conducted a benchmark analysis, evaluating the detection ratio and accuracy of the four tools using the simulated WGS datasets. The results (Table 1) demonstrate that STRsensor consistently achieves the highest detection ratio and accuracy across all the simulated datasets. Even in the 30× WGS dataset, STRsensor still able to achieve 100% detection ratio and 99.37% accuracy. In contrast, the accuracy of other three tools such as HipSTR, STRaitRazor and STRinNGS reaches 94.85%,

Table 1. Comparison of average detection ratio (DR) and accuracy (AC) of STRsensor, HipSTR, STRaitRazor, and STRinNGS in three simulated WGS datasets with varying coverage (10×, 30× and 50×).

	SimWGS (10×)		SimWGS (30×)		SimWGS (50×)	
	DR (%)	AC (%)	DR (%)	AC (%)	DR (%)	AC (%)
STRsensor	99.55	93.71	100.00	99.37	100.00	99.85
HipSTR	95.17	90.60	95.34	94.85	95.35	95.16
STRaitRazor	85.99	75.55	93.40	86.90	94.69	89.50
STRinNGS	91.01	69.11	92.80	77.28	92.65	80.81

Table 2. Comparison of average detection ratio (DR) and accuracy (AC) of STRsensor, HipSTR, STRaitRazor, and STRinNGS in four different proportions of reads, 1% (16.27×), 5% (78.73×), 10% (157.02×), and 100% (1,570×), randomly extracted from Fudan687 dataset.

	Fudan687 (1%)		Fudan687 (5%)		Fudan687 (10%)		Fudan687 (100%)	
	DR (%)	AC (%)	DR (%)	AC (%)	DR (%)	AC (%)	DR (%)	AC (%)
STRsensor	72.89	97.87	97.59	99.46	99.24	99.65	99.64	99.90
HipSTR	73.66	72.96	73.33	73.29	75.92	78.87	80.26	85.31
STRaitRazor	62.85	95.57	94.28	99.38	96.67	99.63	98.51	99.65
STRinNGS	60.47	83.44	81.34	86.57	88.45	93.44	97.32	96.98

86.90%, and 77.28%, respectively. Compared with STRsensor, HipSTR achieves performance in terms of detection ratio and accuracy, however, some of the STR Loci (i.e. DYS19 and DYS557) cannot be typed due to the repetitive sequence in the flanking region. Furthermore, the output of HipSTR requires additional parsing to obtain the final allele size, making it less user-friendly for researchers without bioinformatics experience. Since STRaitRazor and STRinNGS are specially designed for target sequencing data, their detection ratio is poor in simulated low-coverage WGS data. The main reason is that, unlike target sequencing data, the reads in WGS data are organized in a 'ladder' pattern on the reference genome, leading to a sharp reduction in the number of reads fully span the STR region. This phenomenon is more severe in STR loci with longer allelic repeats, such as DYS390, DYS627, DYS557, etc.

Performance in 687 target sequencing datasets

In recent years, target sequencing technology has been widely used in the field of forensics owing to its ultra-high sensitivity [13, 14]. To further evaluate the performance of the four software in the target sequencing data, we obtained 687 samples (Fudan687) from Fudan University. The sequencing type of Fudan687 is single-end 400 bp, with an average depth of at least 1570×. The 53 STR loci used in the benchmark include both NGS data and capillary electrophoresis typing results, of which 19 are autosomal STR loci and the rest are Y-STR loci.

The same benchmarking framework is performed here to compare the detection ratio and accuracy of the four software. The results (Table 2) show that the detection ratio and accuracy of STRsensor reaches 99.64% and 99.9%, respectively. Compared with STRsensor, STRaitRazor obtains a very close detection ratio of 98.51%, which is attributed to the fact that all the reads from target sequencing can fully span the target STR locus. Furthermore, owing to the high sequencing depth and low proportion of PCR stutter (approximately 8%), it is generally accurate to rely on the allele with the highest reads count. Among the four tools, HipSTR has the worst performance in accuracy (85.31%), as up to eight Y-STR loci failed to type, while the detection ratio of other three Y-STR loci are less than 50% (details in Supplementary File).

In real-case applications, there are many factors that contribute to low sequencing depth, such as highly degraded DNA, traced DNA, and low efficiency of STR amplification. To explore the performance of the four software under extreme conditions, we randomly extracted 1% (16.27×), 5% (78.73×), and 10% (157.02×) reads from each sample in Fudan687 dataset. Considering the low coverage, the minimum number of reads for the four software is set to 10, and the locus of DYS447 is set to 0 due to its ultra-low coverage. The performance of the four tools on these three datasets is evaluated using the same benchmarking method described above. When using only 1% (16.27×) reads from the dataset, STRsensor can achieve an accuracy of 97.87% (Table 2). In comparison, the accuracy of HipSTR, STRaitRazor and STRinNGS is 72.96%, 95.57%, and 83.44%, respectively. Compared to the other three software, HipSTR exhibits the poorest performance in both detection ratio and accuracy. Although the accuracy (95.57%) of STRaitRazor is comparable to that of STRsensor (97.87%) within a coverage of 16.27×, its detection ratio is only 62.85%, which is 10.04% lower than that of STRsensor. With the sequencing depth increases to 157.02×, STRsensor still able to outperform the other three software, achieving a detection ratio of 97.59% and an accuracy of 99.46%. STRinNGS demonstrates a significant improvement in both detection ratio (88.45%) and accuracy (93.44%). This improvement can be attributed to the fact that, even with its strict filtering rules, a sufficient number of reads can still be obtained for STR allele-typing.

Effectiveness of STRsensor core algorithms

To assess the effectiveness of the STRsensor core algorithms, specifically Kmer, CIGAR, CIGAR correction, and MAP, on the sensitivity and precision of STR typing, we conducted a systematic evaluation using the Fudan687 dataset and simulated low-coverage WGS data (10×). This study aims to elucidate the impact of these algorithmic components on STR typing performance. The testing was divided into five groups: complete (including all algorithm components), without MAP, without CIGAR, without Kmer, and without CIGAR correction (using the CIGAR algorithm without correction). The findings indicate that the CIGAR correction

Table 3. The average detection ratio (DR) and accuracy (AC) of STRsensor core algorithms when applied to the Fudan687 (1570×) and simulated low-coverage WGS (10×) datasets.

	Fudan687 (1,570×)		SimWGS (10×)	
	DR (%)	AC (%)	DR (%)	AC (%)
Complete	99.64	99.90	99.55	93.71
Without MAP	95.45	99.70	99.37	86.42
Without CIGAR	97.61	96.00	99.25	92.85
Without Kmer	98.64	97.74	99.05	91.23
Without CIGAR Correction	87.21	76.45	74.33	71.64

has the most significant impact on the sensitivity and accuracy of STRsensor (Table 3). In the Fudan687 dataset, compared to the original CIGAR method (without Kmer), the mean sensitivity and accuracy decreased by 11.43% and 21.29%, respectively (Table 3). Converesely, the removal of other algorithms has minimal impact on STR typing due to sufficient spanning reads in the high-coverage Fudan687 dataset, mitigating allele typing challenges (details in Supplementary File).

Evaluation of short-read (150 bp), low-coverage WGS (10×) data revealed a substantial impact of the CIGAR correction component on STRsensor performance, resulting in a total reduction in STR typing sensitivity and accuracy by 25.22% and 22.07%, respectively (Table 3). Furthermore, removing the MAP algorithm decreased STR typing accuracy from 93.71% to 86.42% (Table 3). This decline is primarily due to the reduced number of spanning reads, making it challenging to reliably determine the STR allele with the highest supporting reads. In contrast, the MAP algorithm utilizes the Stutter model to infer the most likely STR allele, thereby enhancing the accuracy of STR typing.

Time efficiency of the methods

Time efficiency is a crucial performance metric, especially in multi-sample scenarios and ultra-high coverage sequencing data. Given that STRaitRazor operates on FASTQ files rather than BAM files, it was excluded from the time consumption assessments. To evauate the efficiency of the remaining three software, we conducted tests using 10 samples from the high-coverage Fudan687 dataset, with an average depth of 3216.02×. Due to the slow running speed of STRinNGS, it was allocated 20 threads, while the other two software operated with a single thread. The results reveal that STRsensor completes typing for all 10 high-coverage samples in just 136 seconds, which is 79 times faster than HipSTR (179 minutes). Moreover, STRsensor outpaces STRinNGS by over 10 000 times, as the latter took 1145 minutes with 20 threads. Memory consumption is not a significant concern in this benchmark since STRsensor only stores the typed alleles in memory, resulting in a minimal memory footprint that ensures its compatibility with most computing devices.

Discussion

We present STRsensor, a computationally efficient software for STR typing from next-generation sequencing data. STRsensor exhibits higher detection ratio and accuracy for NGS data from different sequencing strategies (WGS and target sequencing) as well as different sequencing depths (low-coverage and high-coverage).

STRsensor is a user-friendly software capable of analyzing both commonly-used STR loci and user-specified STR loci. With the BAM file of samples as input, STRsensor directly provides alleles and their associated probabilities at each STR locus, eliminating the need for additional user-driven extraction and analysis. In contrast, STRaitRazor outputs a series of allele sizes derived from spanning reads, without final allele determination. STRling and ExpansionHunter are specifically designed for typing disease-related STR loci. However, they have poor support for single-end sequencing data. Although HipSTR offers precise STR positons, may yield inconsistent typing results for certain STR loci compared to CE results, requiring manual correction to improve the accuracy. For example, to obtain an allele consistent with CE, HipSTR requires a reduction of three bases when determining the allele size of D2S1338. A major limitation of STRinNGS is its poor accuracy in allele-typing for WGS data. Additionally, it runs much slower than other software, significantly constraining its application to large sample datasets.

The accuracy of STRsensor typing is influenced by several factors, including the repeatability of STR flanking sequences, allele length, sequencing read length, PCR stutter distribution, and allele frequency distribution in the population. STRsensor relies on spanning reads for allele typing, limiting its capability with STR loci whose allele length exceeds the sequencing read length. While STRsensor enhances matching accuracy of STR flanking sequences by leveraging locally unique Kmers in flanking sequences, higher sequence repeatability reduces the number of such Kmers, thereby lowering STR typing sensitivity. This challenge can usually be addressed through targeted sequencing designs or increased read length.

Although allele frequency and PCR stutter distribution can affect the MAP algorithm of STRsensor, STRsensor can gradually correct these distributions during the processing of input samples, minimizing the impact of the initial distribution on STR typing accuracy. We used a uniform distribution instead of the original STR allele frequency and tested it in Fudan687 and low-coverage WGS (10×) datasets. The results showed that STR typing accuracy only dropped by about 1%. For STR loci with unknown allele frequencies, we recommend training the STRsensor stutter model on a batch of samples (e.g. 100) to refine model parameters and enhance typing accuracy tailored to the population.

The typing accuracy of STRsensor can be further improved by modeling the heterozygote balance of STR allele. This adjustment is particularly important in WGS data, where the number of spanning reads for an allele is dramatically influenced by its length. The longer the allele length, the fewer corresponding spanning reads will be obtained, in which case the heterozygous allele may be determined as a homozygous allele. However, this scenario is infrequent in practical use and typically occurs at locus with large variation in allele length, such as PentaE, where allele size range from 5 to 28. Implementing a model for heterozygote balance represents a promising avenue for future enhancements to STRsensor.

Conclusion

In this study, we developed STRsensor, a method that could achieve high detection ratio and accuracy for both low-coverage WGS data and target sequencing data. Through the integration of Kmers- and CIGAR-based methods, STRsensor surpasses the existing tools, including STRait Razor, HipSTR and STRinNGS, demonstrating a higher detection ratio. Furthermore, STRsensor models the PCR stutter to infer the most likely alleles, greatly

improving the accuracy of allele typing, particularly in the context of low-coverage WGS data. Notably, STRsensor is easy to use and computationally efficient, making it a valuable tool for the application of NGS-based STR analyses across various domains, including medical genetics, population genetics and forensics.

Key Points

- We have developed a new method called STRsensor for inferring genotypes of STR loci from NGS sequencing data. STRsensor utilizes two methods for genotyping STR loci, namely Kmers-based method and CIGAR-based method, and achieve base-pair level accuracy in genotyping.
- STRsensor constructs a PCR stutter model, which can estimate the model parameters using user-provided datasets, and provide the maximum A posteriori (MAP) estimate of the genotypes of target STR loci.
- Through benchmarks on simulated and real datasets, STRsensor demonstrates high performance in terms of detection ratio and accuracy. STRsensor achieves a detection ratio of 100% and an accuracy of 99.37% for a 30× WGS simulated data, outperforming the existing methods, such as STRait Razor, STRinNGS, and HipSTR. When applied to real target sequencing data from 687 individuals, STRsensor achieves a detection ratio of 99.64% and an accuracy of 99.99%. Moreover, it is computationally efficient, surpassing existing software by orders of magnitude.

Funding

This study was supported by the National Natural Science Foundation of China (Grant Nos. 81930056 and 32370669).

Supplementary data

Supplementary data is available at *Briefings in Bioinformatics* online.

Data availability

The script 'STRsimulator.py' that used to generate the simulated WGS data is available at GitHub. The 687 raw sequencing data (Fudan687) generated during the current study are available from the corresponding author upon requests. Additionally, The source code of STRsensor is also available at GitHub (https://github.com/ChenHuaLab/STRsensor).

References

1. Warshauer DH, Lin D, Hari K. *et al*. STRait razor: a length-based forensic STR allele-calling tool for use with second generation sequencing data. *Forensic Sci Int Genet* 2013;**7**:409–17. https://doi.org/10.1016/j.fsigen.2013.04.005.
2. Borsting C, Morling N. Next generation sequencing and its applications in forensic genetics. *Forensic Sci Int Genet* 2015;**18**:78–89. https://doi.org/10.1016/j.fsigen.2015.02.002.
3. Gymrek M, Golan D, Rosset S. *et al*. lobSTR: a short tandem repeat profiler for personal genomes. *Genome Res* 2012;**22**:1154–62. https://doi.org/10.1101/gr.135780.111.
4. Balding DJ, Bishop M, Cannings C. *Handbook of Statistical Genetics*. Chichester, UK: John Wiley & Sons, 2001.
5. Willems T, Zielinski D, Yuan J. *et al*. Genome-wide profiling of heritable and de novo STR variations. *Nat Methods* 2017;**14**:590–2. https://doi.org/10.1038/nmeth.4267.
6. Gemayel R, Vinces MD, Legendre M. *et al*. Variable tandem repeats accelerate evolution of coding and regulatory sequences. *Annu Rev Genet* 2010;**44**:445–77. https://doi.org/10.1146/annurev-genet-072610-155046.
7. Li Y-C, Korol AB, Fahima T. *et al*. Microsatellites within genes: structure, function, and evolution. *Mol Biol Evol* 2004;**21**:991–1007. https://doi.org/10.1093/molbev/msh073.
8. Butler JM. Genetics and genomics of core short tandem repeat loci used in human identity testing. *J Forensic Sci* 2006;**51**:253–65. https://doi.org/10.1111/j.1556-4029.2006.00046.x.
9. Yang Y, Xie B, Yan J. Application of next-generation sequencing technology in forensic science. *Genom Proteom Bioinform* 2014;**12**:190–7. https://doi.org/10.1016/j.gpb.2014.09.001.
10. Shi C-M, Li C, Ma L. *et al*. Inferring Chinese surnames with Y-STR profiles. *Forensic Sci Int: Genet* 2018;**33**:66–71. https://doi.org/10.1016/j.fsigen.2017.11.014.
11. Kayser M, de Knijff P. Improving human forensics through advances in genetics, genomics and molecular biology. *Nat Rev Genet* 2011;**12**:179–92. https://doi.org/10.1038/nrg2952.
12. Thompson R, Zoppis S, McCord B. An overview of DNA typing methods for human identification: past, present, and future. *Methods Mol Biol* 2012;**830**:3–16. https://doi.org/10.1007/978-1-61779-461-2_1.
13. Wang L, Chen M, Wu B. *et al*. Massively parallel sequencing of forensic STRs using the ion chef and the ion S5 XL systems. *J Forensic Sci* 2018;**63**:1692–703. https://doi.org/10.1111/1556-4029.13767.
14. Zeng X, King J, Hermanson S. *et al*. An evaluation of the PowerSeq auto system: a multiplex short tandem repeat marker kit compatible with massively parallel sequencing. *Forensic Sci Int Genet* 2015;**19**:172–9. https://doi.org/10.1016/j.fsigen.2015.07.015.
15. Van Neste C, Van Nieuwerburgh F, Van Hoofstat D. *et al*. Forensic STR analysis using massive parallel sequencing. *Forensic Sci Int Genet* 2012;**6**:810–8. https://doi.org/10.1016/j.fsigen.2012.03.004.
16. Planz JV, Sannes-Lowery KA, Duncan DD. *et al*. Automated analysis of sequence polymorphism in STR alleles by PCR and direct electrospray ionization mass spectrometry. *Forensic Sci Int Genet* 2012;**6**:594–606. https://doi.org/10.1016/j.fsigen.2012.02.002.
17. Bornman DM, Hester ME, Schuetter JM. *et al*. Short-read, high-throughput sequencing technology for STR genotyping. *Biotech Rapid Dispatches* 2012;**1–6**:2012. https://doi.org/10.2144/000113857.
18. Fordyce SL, Avila-Arcos MC, Rockenbauer E. *et al*. High-throughput sequencing of core STR loci for forensic genetic investigations using the Roche genome sequencer FLX platform. *Biotechniques* 2011;**51**:127–33. https://doi.org/10.2144/000113721.
19. Warshauer DH, King JL, Budowle B. STRait razor v2.0: the improved STR allele identification tool – Razor. *Forensic Sci Int Genet* 2015;**14**:182–6. https://doi.org/10.1016/j.fsigen.2014.10.011.
20. Jonck CG, Qian X, Simayijiang H. *et al*. STRinNGS v2.0: improved tool for analysis and reporting of STR sequencing data. *Forensic Sci Int Genet* 2020;**48**:102331. https://doi.org/10.1016/j.fsigen.2020.102331.
21. Guo F, Jiao Y, Zhang L. *et al*. Massively parallel sequencing of forensic STRs and SNPs using the Illumina® ForenSeq™M

DNA signature prep kit on the MiSeq FGx^{TM} forensic genomics system. *Forensic Sci Int: Genet* 2017;**31**:135–48. https://doi.org/10.1016/j.fsigen.2017.09.003.
22. Zhang S, Niu Y, Bian Y. *et al*. Sequence investigation of 34 forensic autosomal STRs with massively parallel sequencing. *Sci Rep* 2018;**8**:6810. https://doi.org/10.1038/s41598-018-24495-9.
23. Tang H, Nzabarushimana E. STRScan: Targeted profiling of short tandem repeats in whole-genome sequencing data. *BMC bioinformatics* 2017;**18**:31–6.
24. Dashnow H, Pedersen BS, Hiatt L. *et al*. STRling: a k-mer counting approach that detects short tandem repeat expansions at known and novel loci. *Genome Biol* 2022;**23**:257. https://doi.org/10.1186/s13059-022-02826-4.
25. Dolzhenko E, Deshpande V, Schlesinger F. *et al*. ExpansionHunter: a sequence-graph-based tool to analyze variation in short tandem repeat regions. *Bioinformatics* 2019;**35**:4754–6. https://doi.org/10.1093/bioinformatics/btz431.
26. Zhang X, Shao Y, Tian J. *et al*. pTrimmer: an efficient tool to trim primers of multiplex deep sequencing data. *BMC Bioinform* 2019;**20**:236. https://doi.org/10.1186/s12859-019-2854-x.
27. Li H, Handsaker B, Wysoker A. *et al*. The sequence alignment/map format and SAMtools. *Bioinformatics* 2009;**25**:2078–9. https://doi.org/10.1093/bioinformatics/btp352.
28. Jia P, Yang X, Guo L. *et al*. MSIsensor-pro: fast, accurate, and matched-normal-sample-free detection of microsatellite instability. *Genom Proteom Bioinform* 2020;**18**:65–71. https://doi.org/10.1016/j.gpb.2020.02.001.
29. Ying Qian Y, Zhang, and Jiongmin Zhang. Alignment-free sequence comparison with multiple k values. *IEEE/ACM Trans Comput Biol Bioinform* 2021;**18**:1841–9. https://doi.org/10.1109/TCBB.2019.2955081.

Briefings in Bioinformatics, 2025, **26(1)**, bbae637
https://doi.org/10.1093/bib/bbae637
Problem Solving Protocol