frontiers in Genetics

ORIGINAL RESEARCH
published: 27 July 2020
doi: 10.3389/fgene.2020.00706

SSRMMD: A Rapid and Accurate Algorithm for Mining SSR Feature Loci and Candidate Polymorphic SSRs Based on Assembled Sequences

Xiangjian Gou[1,2†], Haoran Shi[1†], Shifan Yu[1], Zhiqiang Wang[1], Caixia Li[1], Shihang Liu[1], Jian Ma[1], Guangdeng Chen[3], Tao Liu[4] and Yaxi Liu[1,5]**

[1] Triticeae Research Institute, Sichuan Agricultural University, Chengdu, China, [2] Maize Research Institute, Sichuan Agricultural University, Chengdu, China, [3] College of Resources, Sichuan Agricultural University, Chengdu, China, [4] College of Information Engineering, Sichuan Agricultural University, Ya'an, China, [5] State Key Laboratory of Crop Gene Exploration and Utilization in Southwest China, Chengdu, China

OPEN ACCESS

Edited by:
Thomas Edward Gorochowski,
University of Bristol, United Kingdom

Reviewed by:
Jian-Feng Mao,
Beijing Forestry University, China
Mir Asif Iquebal,
Indian Agricultural Statistics Research Institute (ICAR), India
Suresh Babu Mudunuri,
Sagi Ramakrishnam Raju Engineering College, India

Correspondence:
Tao Liu
liutao@sicau.edu.cn
Yaxi Liu
liuyaxi@sicau.edu.cn

† These authors have contributed equally to this work

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *07 April 2020*
Accepted: *10 June 2020*
Published: *27 July 2020*

Citation:
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. 11:706. doi: 10.3389/fgene.2020.00706

Microsatellites or simple sequence repeats (SSRs) are short tandem repeats of DNA widespread in genomes and transcriptomes of diverse organisms and are used in various genetic studies. Few software programs that mine SSRs can be further used to mine polymorphic SSRs, and these programs have poor portability, have slow computational speed, are highly dependent on other programs, and have low marker development rates. In this study, we develop an algorithm named Simple Sequence Repeat Molecular Marker Developer (SSRMMD), which uses improved regular expressions to rapidly and exhaustively mine perfect SSR loci from any size of assembled sequence. To mine polymorphic SSRs, SSRMMD uses a novel three-stage method to assess the conservativeness of SSR flanking sequences and then uses the sliding window method to fragment each assembled sequence to assess its uniqueness. Furthermore, molecular biology assays support the polymorphic SSRs identified by SSRMMD. SSRMMD is implemented using the Perl programming language and can be downloaded from https://github.com/GouXiangJian/SSRMMD.

Keywords: bioinformatics, algorithm, simple sequence repeats, conservativeness, uniqueness, polymorphism

INTRODUCTION

Owing to their abundance, codominant inheritance, multi-allelic nature, transferability, and ease of analysis *via* PCR (Varshney et al., 2005; Ramu et al., 2009; Kaur et al., 2015), simple sequence repeat (SSR) markers have been successfully adopted in various genetic studies such as quantitative trait loci mapping (Qin et al., 2015; Wang et al., 2017), genotyping (Gramazio et al., 2018), genetic diversity (Nachimuthu et al., 2015; Zhou R. et al., 2015), and DNA fingerprinting

(Zhang et al., 2015). Indeed, numerous genome-wide SSR markers have been identified in plants and animals in recent years, such as those in rice (Zhang et al., 2007), maize (Xu et al., 2013), cucumber (Liu et al., 2015), bee (Liu et al., 2016), tobacco (Wang et al., 2018), and snake (Liu et al., 2019).

During the development of SSR markers, the first step is the mining of potential SSR loci from assembled sequences. Based on the repetitive architecture of their motifs, SSRs can be classified as perfect (e.g., AGAGAGAGAGAG), and imperfect (including nucleotide substitutions or indels, e.g., AGAGAGACAGAG). However, the application of perfect SSRs in genetic studies far exceeds that of imperfect SSRs because of its higher allelic variability (Zalapa et al., 2012; Xu et al., 2013). Numerous algorithms and software programs have been reported for mining perfect SSRs. For instance, SSRIT (Temnykh, 2001), MISA (Thiel et al., 2003), and GMATo (Wang et al., 2013) use regular expressions based on the greedy matching algorithm to mine SSRs. SA-SSR (Pickett et al., 2016) uses a suffix array-based algorithm to mine SSRs. Kmer-SSR (Pickett et al., 2017) uses Kmer decomposition to identify SSRs. PERF (Avvaru et al., 2017) matches each potential substring in accordance with a set of pre-computed repeat strings. Other programs including TROLL (Castelo et al., 2002), MfSAT (Chen et al., 2011), ProGeRF (Silva et al., 2015), and FullSSR (Metz et al., 2016) have also been developed. In addition, imperfect SSR detection algorithms have also been reported, such as IMEx (Mudunuri and Nagarajaram, 2007), and Krait (Du et al., 2017). However, these programs have many common undesirable features. First, they rely on additional software or modules, often with complex software configuration; second, they have poor portability and can only be run on Linux or Windows platforms; third, they have slow computational speed; and most importantly, polymorphic SSRs cannot be directly found.

With rapid advancements in genomics, software and pipelines for mining polymorphic SSRs have been reported. For instance, CandiSSR (Xia et al., 2016), a candidate polymorphic SSRs identification pipeline, is based on multiple assembly sequences. GMATA (Wang and Wang, 2016) provides a complete process for SSR markers development. IDSSR (Guang et al., 2019) has recently been reported to identify polymorphic SSRs in a single genome sequences using a similar pipeline. However, these programs or pipelines also share certain issues. First, they rely on numerous other programs, such as MISA (Thiel et al., 2003), Primer3 (Untergasser et al., 2012), BLAST (Altschul et al., 1997), and ClustalW (Thompson et al., 2002); second, they have slow computational speed for mining polymorphic SSRs; finally, they have low rates of SSR markers development.

To overcome these limitations, we developed the Simple Sequence Repeat Molecular Marker Developer (SSRMMD) program using the Perl programming language. This program rapidly and exhaustively mines perfect SSR loci through improved regular expressions. For mining polymorphic SSRs, this program uses a high-stringency sequence alignment algorithm to assess the conservativeness and uniqueness of SSR flanking sequences. Compared with other software programs, SSRMMD is more rapid, accurate, and convenient. SSRMMD can be downloaded from https://github.com/GouXiangJian/SSRMMD.

MATERIALS AND METHODS

Implemented Algorithm

The algorithm of SSRMMD involves the mining of perfect SSR loci and the discovery of polymorphic SSRs. The internal methodological details are provided in **Figure 1**, primarily including the following steps:

(1) Mining perfect SSR loci. Similar to programs such as SSRIT (Temnykh, 2001) and MISA (Thiel et al., 2003), SSRMMD uses regular expressions with the greedy matching algorithm to mine SSRs. However, to improve computational speed, SSRMMD was optimized in three aspects: (i) use of multi-threading technology. To maximize the function of each thread, we proposed a novel optimal allocation algorithm to averagely distribute assembled sequences to each thread in accordance with the length of sequences (TOS), including the following: (a) sort sequences by TOS; (b) assignment of the longest i sequences to i threads; (c) thread sorting based on the total TOS; (d) assignment of subsequent sequences to the thread with the smallest TOS; (e) thread sorting in step (d) using the insertion sorting algorithm; and (f) iterative performance of steps (d) and (e) until complete sequence allocation. (ii) Fragmented sequences. After a specific thread is assigned to store each sequence, SSRMMD fragmented each sequence into short 500-kb fragments. At this length, the computational speed was the highest. Furthermore, 5 kb was added to each fragment to prevent potential SSRs from being cut off. (iii) Improved regular expression. Ordinary regular expressions can only mine one type of motif in each match, as indicated using MISA (Thiel et al., 2003). However, by integrating all patterns, SSRMMD can mine all types of motif in each match, indicating that irrespective of the arrangement of the threshold motifs, SSRMMD will only traverse the sequence once. Notably, to completely mine compound SSRs, SSRMMD backtracks after each successful match, and the size of backtracking (B) is as follows:

$$size\left(B\right)=\begin{cases} length\left(motif_i\right)-1, & sum\left(motif_i\right)==1 \\ \max\left\{L_1,L_2,\ldots,L_n\right\}, & \\ \min\left\{S_1,S_2,\ldots,S_n\right\} & >\max\left\{L_1,\ L_2,\ \ldots,L_n\right\} \\ \min\left\{S_1,S_2,\ldots,S_n\right\}-1, & \\ \min\left\{S_1,S_2,\ldots,S_n\right\} & \leq\max\left\{L_1,\ L_2,\ \ldots,L_n\right\} \end{cases}$$

where n is the number of motif types, S_i is the length of the ith motif types of SSR, and L_i is the length of the ith type of motif.

(2) Assessment of the conservativeness of SSR flanking sequences. To develop polymorphic SSRs, we initially assessed the conservativeness of SSR flanking sequences. To maximize the computational speed, we used a novel three-stage method to align flanking sequences between two assembled sequences files, which included the following steps: (i) first, absolutely conserved flanking sequences were filtered out using HASH structure. Herein, we considered these flanking sequences in the first assembled file as a library, and then we compared

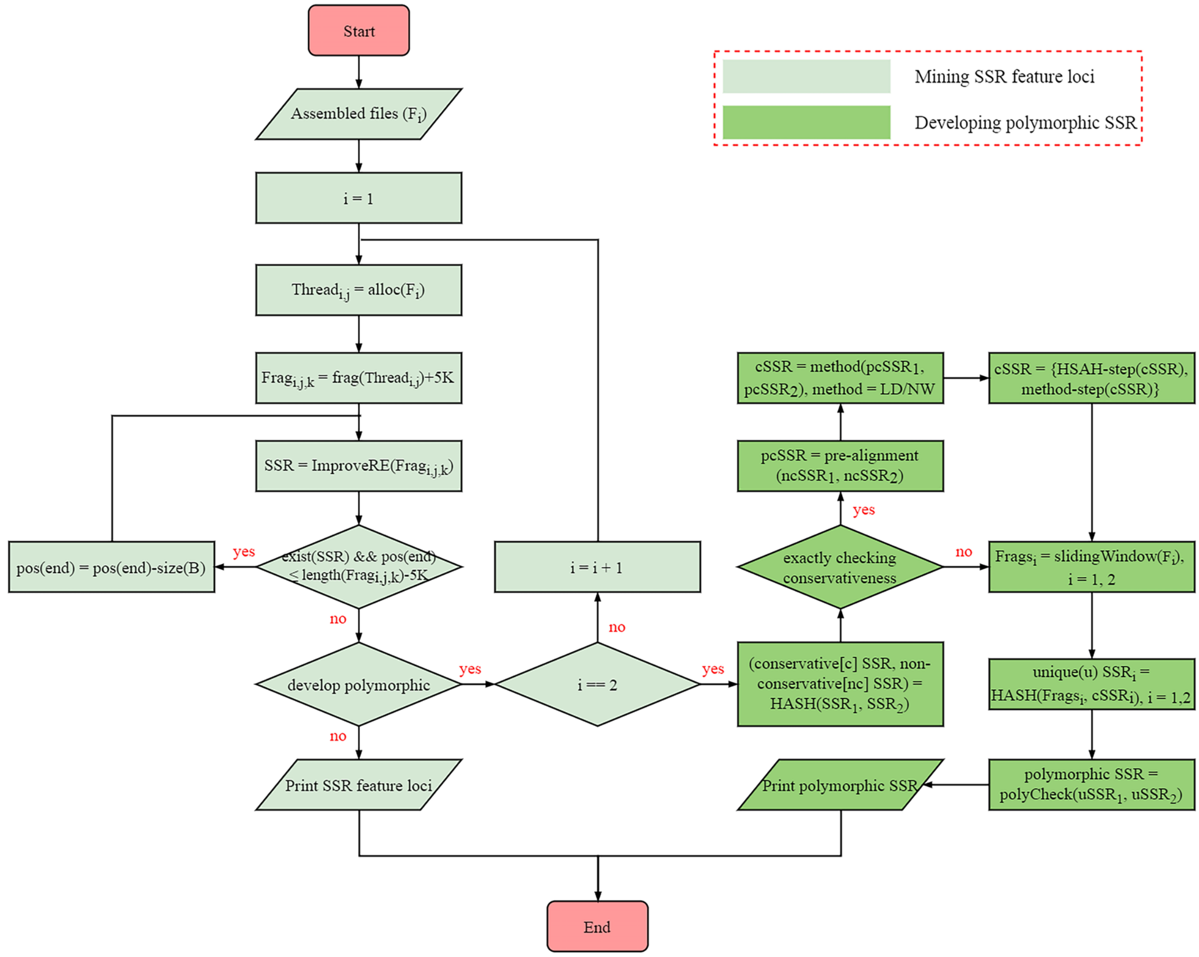
Start
Assembled files (F$_i$)
i = 1
Thread$_{i,j}$ = alloc(F$_i$)
Frag$_{i,j,k}$ = frag(Thread$_{i,j}$)+5K
SSR = ImproveRE(Frag$_{i,j,k}$)
exist(SSR) && pos(end) ≤ length(Frag$_{i,j,k}$)-5K
yes
pos(end) = pos(end)-size(B)
no
develop polymorphic
no
Print SSR feature loci
yes
i == 2
no
i = i + 1
yes
(conservative[c] SSR, non-conservative[nc] SSR) = HASH(SSR$_1$, SSR$_2$)
exactly checking conservativeness
yes
pcSSR = pre-alignment (ncSSR$_1$, ncSSR$_2$)
cSSR = method(pcSSR$_1$, pcSSR$_2$), method = LD/NW
cSSR = {HSAH-step(cSSR), method-step(cSSR)}
no
Frags$_i$ = slidingWindow(F$_i$), i = 1, 2
unique(u) SSR$_i$ = HASH(Frags$_i$, cSSR$_i$), i = 1,2
polymorphic SSR = polyCheck(uSSR$_1$, uSSR$_2$)
Print polymorphic SSR
End
Mining SSR feature loci
Developing polymorphic SSR

FIGURE 1 | Internal implementation algorithm of Simple Sequence Repeat Molecular Marker Developer (SSRMMD).

these flanking sequences in the another assembled file with the aforementioned library to rapidly identify absolutely conserved flanking sequences. (ii) Second, conservativeness pre-alignment was performed using x% [default is 5% (each side is 5 bp)] flanking sequences near SSRs. Assuming that flanking sequences near SSRs were highly conserved, SSRMMD allowed flanking sequences near SSRs to tolerate up to 2-bp mismatches. Moreover, after extensive assessments, additional mismatches (≥3 bp) did not further benefit the results, consistent with the aforementioned assumption. SSRMMD iteratively replaced mismatched bases and aligned flanking sequences between two assembled files, using a method similar to (i). (iii) Finally, SSRMMD used Levenshtein distance (LD; Levenshtein, 1966), or the Needleman–Wunsch (NW) algorithm (Needleman and Wunsch, 1970) to accurately assess the conservativeness of the flanking sequences retained through pre-alignment. LD was defined as the minimum number of edits required to convert one string to another, thus indirectly reflecting the identity of two DNA sequences. However, the NW algorithm based on dynamic programming has been extensively used for global sequence alignment, directly reflecting the identity of two DNA sequences. Compared with NW algorithm, the LD did not require backtracking; hence, it had a higher computational speed; furthermore, the NW algorithm had a more comprehensive scoring system than LD, thus facilitating more accurate elucidation of the identity of the SSR flanking sequences. The iterative formulae of the LD and NW algorithms are as follows:

$$LD_{a,b}(i,j) = \begin{cases} \max\left(i,j\right) & \min\left(i,j\right) = 0 \\ \min\begin{cases} LD_{a,b}\left(i-1,j\right)+1 \\ LD_{a,b}\left(i,j-1\right)+1 \\ LD_{a,b}\left(i-1,j-1\right)+1_{(a_i \neq b_j)} \end{cases} & otherwise \end{cases}$$

$$NW_{a,b}(i,j) = \begin{cases} 0 & \mathrm{i,j}=0 \\ max\begin{cases} NW_{a,b}\left(i-1,j\right)+ \\ S_{gap} & a_i\ aligned\ to\ a\ gap \\ NW_{a,b}\left(i,j-1\right)+ \\ S_{gap} & b_j\ aligned\ to\ a\ gap \\ NW_{a,b}\left(i-1,j-1\right)+ \\ S_{match/mismatch} & a_i\ aligned\ to\ b_j \end{cases} \end{cases}$$

where *a* and *b* are 2 strings; *i* and *j* are subscripts of *a* and *b*, respectively; S_{match} is the score of match; $S_{mismatch}$ is the score of mismatch; and S_{gap} is the score of gap.

(3) Assessment of the uniqueness of SSR flanking sequences. After conservativeness was assessed, SSRMMD further assessed the uniqueness of SSR flanking sequences. Again, assembled sequences were evenly distributed to each thread and were fragmented through the sliding window method, wherein window size was the length of flanking sequences, the step size was 1 bp, and all fragments were stored in a HASH database. Thereafter, flanking sequences with the equal sizes in the aforementioned HASH database were aligned to identify SSRs with unique flanking sequences. Finally, polymorphisms were compared in the two unique SSR sets to distinguish monomorphic and polymorphic SSRs. Notably, to meet different needs, SSRMMD used two computational methods, (i) running in a time-saving manner and (ii) running in a memory-saving manner, indicating that SSRMMD functions adequately, irrespective of the use of a personal computer, or high-performance server.

Input and Output

Assembled sequences (e.g., genome, transcriptome, or a single gene) with a standard FASTA format is required for mining SSRs; to further develop candidate polymorphic SSRs, another assembled sequence is required. Certain parameters can be set to change the SSR mining conditions, including motif threshold and the length of flanking sequences. SSRMMD is allowed to set any size motif (>6 bp), and SSRMMD would then assess the conservativeness and uniqueness of SSR flanking sequences when mining polymorphic SSRs. Notably, setting more threads would significantly enhance the computational speed.

Upon completion of the computation, SSRMMD yields three types of outputs: (i) detailed information record file of SSRs; (ii) statistical file of SSRs, which analyzes the various distribution characteristics of SSRs and helps understand the distribution pattern of the SSRs [including the following: (a) SSR number and density in each assembled sequence; (b) SSR number and proportion per unit length of the motif; and (c) SSR number among different numbers of repeats in each motif]; and (iii) detailed information record file of candidate polymorphic SSRs.

Performance Test Datasets

To assess SSRMMD, we downloaded six genomes of three plants from National Center for Biotechnology Information (NCBI)[1] and Unité de Recherche Génomique Info (URGI)[2]. Three genomes were used to assess the potential for mining SSR feature loci, including rice (Zhenshan97, ∼0.39 Gb), cotton (TM1, ∼2.29 Gb), and wheat [Chinese Spring (CS), ∼14.23 Gb]. All six genomes were used to assess the potential to mine polymorphic SSRs, including two rice genomes, two cotton genomes, and two wheat genomes. To evaluate the complexity and multi-threading of SSRMMD, we extracted 2-Gb sequences from the wheat CS and AK58 genomes, which were evenly divided into 20 sequences. The GenBank assembly accession numbers of the rice genomes were Zhenshan97 (GCA_001623345.2) and Shuhui498 (GCA_002151415.1). The GenBank assembly accession numbers of cotton genomes were TM1 (GCA_006980745.1) and ZM24 (GCA_006980775.1). The wheat CS and AK58 genomes were obtained from URGI.

Performance Test Parameters

Perfect repeats have higher allelic variability than imperfect repeats, and any SSR used to develop genetic markers should contain a perfect repeat (Xu et al., 2013). Therefore, to assess the potential of SSRMMD for mining SSR loci, we avoided imperfect repeats detection tools, and we selected six popular existing software programs including SSRIT (Temnykh, 2001), MISA (Thiel et al., 2003), GMATA (Wang and Wang, 2016), SA-SSR (Pickett et al., 2016), Kmer-SSR (Pickett et al., 2017), and PERF (Avvaru et al., 2017). In particular, SA-SSR was not included in the results owing to its markedly low computational speed. In each software program, based on previously described methods (Zhang et al., 2007; Xu et al., 2013; Liu et al., 2015), the minimum repeat times of SSR motif lengths of 1, 2, 3, 4, 5, and 6 bp were set to 10, 7, 6, 5, 4, and 4, respectively. Because Kmer-SSR can use multi-threads, we tested SSRMMD and Kmer-SSR with 1 and 12 threads, respectively, to assess its multi-thread support. However, other software programs could only use a single thread.

To assess the potential for mining polymorphic SSRs, we compared two popular existing software programs with SSRMMD, including GMATA (Wang and Wang, 2016), and CandiSSR (Xia et al., 2016). In each software program, SSR flanking sequences were set to 150 bp (Zhang et al., 2007). Because CandiSSR can use multi-threads, we assessed SSRMMD, and CandiSSR with 12 threads; however, GMATA can only use a single thread. On assessing SSRMMD, LD was used to assess the conservativeness of flanking sequences, and the threshold was set to 5% to correspond to the BLAST identity of CandiSSR, and the other parameters (not indicated herein) were retained as default setting. Similarly, parameters not included in GMATA and CandiSSR were used with default setting.

Performance Evaluation Criteria

The performance of SSRMMD and existing software programs for mining perfect SSRs was evaluated in accordance with six criteria. **Table 1** shows the portability, dependence, and function of existing software programs and SSRMMD. The computational accuracy, speed, and memory consumption were evaluated for the test datasets. We used the Linux *time* command to record the computational time and *pmap* command to record the memory peak. All tests were performed using a personal computer with an Intel® Xeonl® CPU E5-2683 v3 @ 2.00 GHz with CentOS Linux release 7.4.1708 and 64 GB RAM.

Experimental Validation

To verify the accuracy of the output through SSRMMD, 80 pairs of polymorphic SSRs were randomly selected from the computational results of wheat for molecular biology assays. These selected polymorphic SSRs were evenly distributed on each chromosome, encompassing differently sized motifs. Genomic

[1] https://www.ncbi.nlm.nih.gov/

[2] https://urgi.versailles.inra.fr/

TABLE 1 | Various features of SSRMMD and existing software programs for mining perfect SSRs.

Software	Year	Portability	Dependence[a]	Function
SSRIT	2001	Windows/Linux	No	Mining SSR feature loci
TROLL	2002	Windows/Linux	Staden	Mining SSR feature loci
MISA	2003	Windows/Linux	No	Mining SSR feature loci
MfSAT	2011	Windows	No	Mining SSR feature loci
GMATo	2013	Windows/Linux	No	Mining SSR feature loci
ProGeRF	2015	Linux	No	Mining SSR feature loci
CandiSSR	2016	Linux	MISA, BLAST, Primer3, Clustalw	Developing polymorphic SSRs
FullSSR	2016	Windows/Linux	BioPerl, Bio:Tools:Run: Primer3	Mining SSR feature loci
GMATA	2016	Windows/Linux	Primer3, e-PCR	Developing polymorphic SSRs
SA-SSR	2016	Linux	No	Mining SSR feature loci
Kmer-SSR	2017	Linux	No	Mining SSR feature loci
PERF	2017	Windows/Linux	tqdm, biopython	Mining SSR feature loci
IDSSR	2019	Linux	SSRIT, BLAST, Primer3	Developing polymorphic SSRs
SSRMMD	this	Windows/Linux	No	Developing polymorphic SSRs

Note. SSRMMD, Simple Sequence Repeat Molecular Marker Developer. [a]*Dependence on additional programs or modules.*

DNA was extracted using the cetyl trimethylammonium bromide (CTAB) method from fresh leaves of 10 wheat popularized and local cultivars of CS, AK58, CM107, CN16, MM37, ZM012542, ZM000652, ZM018703, ZM003222, and ZM003284.

Additionally, we provided a tool named connectorToPrimer3 to associate SSRMMD with Primer3 (Untergasser et al., 2012); hence, a primer design can be easily performed. The primary parameters were as follows: (1) minimum, optimal, and maximum primer sizes of 18, 20, and 27 bp, respectively; (2) minimum and maximum GC contents of 20% and 80%, respectively; (3) minimum, optimal, and maximum Tm values of 57, 60, and 63°C, respectively; and (4) product lengths of 100–300 bp. Primers were synthesized by Beijing Qingke Biotechnology Co., Ltd.

PCR was performed in 10-μl reactions containing 5 μl of mix buffer (2×), 1.0 μl of template DNA (100 ng/μl), 0.5 μl of primers, and 3 μl of ddH_2O. The PCR conditions were as follows: 1 cycle at 94°C for 5 min, 35 cycles at 94°C for 30 s, 60°C for 30 s, 72°C for 30 s, and 1 cycle at 72°C for 10 min. The PCR products were electrophoresed on a 6% denaturing polyacrylamide gel. SSR polymorphisms in different wheat genotypes were identified on the basis of differences in mobility, as revealed through the electrophoretic bands.

RESULTS

Assessment of Complexity and Threads

On the basis of the 2-Gb base sequences from wheat, we tested the time and space complexity of SSRMMD in a single thread. As shown in **Figures 2A,B**, as the amount of data increased, the time and space consumed by SSRMMD increased linearly when mining SSR feature loci. Similarly, when SSRMMD was used to mine polymorphic SSRs (assessing uniqueness in a memory-saving manner), the time and space were also linearly associated with the amount of data (**Figures 2C,D**). These results suggest that the algorithm of SSRMMD has linear time complexity $[T(n) = O(n)]$ and space complexity $[S(n) = O(n)]$.

Furthermore, we assessed the multi-threading support of SSRMMD. As shown in **Figures 2E,F**, whether mining in SSR feature loci or polymorphic SSRs, as the number of threads increased, time consumption decayed as a power function with the number of threads; however, memory consumption scaled linearly. Notably, despite using 10 threads, memory consumption of SSRMMD did not exceed the size of test data (2 Gb). In total, SSRMMD adequately supported multi-threading.

Verification of the Performance of SSRMMD to Mine Simple Sequence Repeat Feature Loci

Based on the high citation rate and novel principles, six software programs were compared with SSRMMD (SA-SSR is not indicated). As shown in **Table 2**, SSRMMD identified the most SSRs. This was larger than other regular expression-based programs including MISA and GMATA. Furthermore, SSRMMD had the highest computational speed when running on a single thread and better supported multi-threading than Kmer-SSR. Additionally, we analyzed the validity of SSRs found by SSRMMD and compared them with four other programs (PERF, Kmer-SSR, GMATA, and MISA). As shown in **Figures 3A–C**, numerous common products were identified in these software programs, accounting for 76.95% (rice), 85.96% (cotton), and 74.21% (wheat) of SSRMMD, respectively.

Verification of the Performance of SSRMMD to Mine Polymorphic Simple Sequence Repeats

CandiSSR and GMATA were compared with SSRMMD. First, compared with CandiSSR, SSRMMD mined approximately doubled the number of polymorphic SSRs, and CandiSSR discarded numerous monomorphic SSRs from among these candidate markers (**Table 3**). Second, compared with GMATA, SSRMMD mined more polymorphic SSRs in rice and cotton, but less in wheat. However, because GMATA identified polymorphisms through e-PCR amplification products, which yield two forms of false positives, (1) the target SSR did not exist in the product and (2) the target SSR in the product was the same size as the reference SSR. Hence, we generated a script[3] to rectify the output of

[3] https://github.com/GouXiangJian/CorrectGMATA

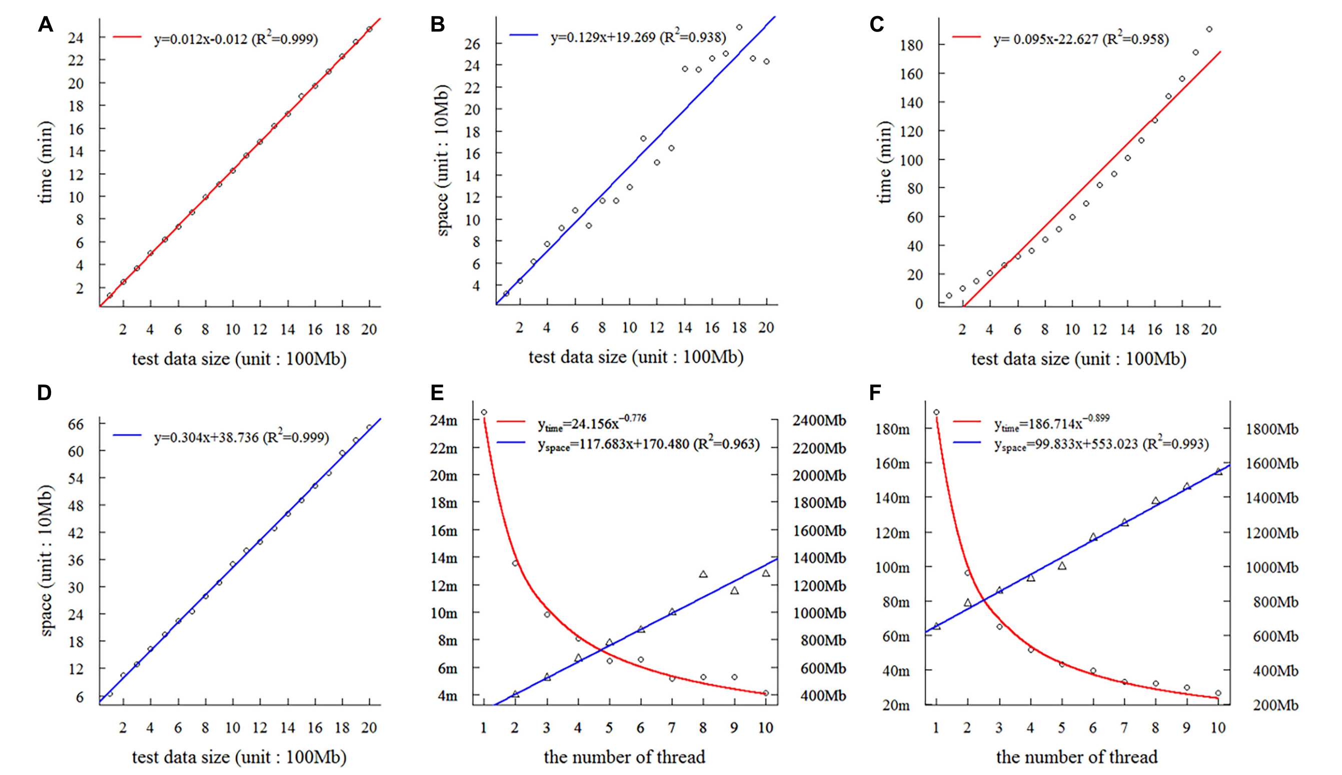

FIGURE 2 | Assessment of complexity and threads of Simple Sequence Repeat Molecular Marker Developer (SSRMMD). **(A)** The time complexity in mining SSR feature loci. **(B)** The space complexity in mining simple sequence repeat (SSR) feature loci. **(C)** Time complexity in developing polymorphic SSRs. **(D)** The space complexity in developing polymorphic SSRs. **(E)** The impact on time and space when increasing threads in mining SSR feature loci. **(F)** The impact on time and space when increasing threads in developing polymorphic SSRs.

TABLE 2 | Performance comparison between SSRMMD and other software programs for identifying genome-wide SSR feature loci.

Software	Thread	Rice (Zhenshan97, ~0.39 Gb)			Cotton (TM1, ~2.29 Gb)			Wheat (CS, ~14.23 Gb)		
		Number	Time (m:s)	Mem (Mb)	Number	Time (m:s)	Mem (Mb)	Number	Time (m:s)	Mem (Mb)
SSRIT	1	111,960	5:53	131.55	384,488	34:45	343.04	1,345,128	210:47	2,503.85
MISA	1	111,905	5:50	205.52	384,400	35:31	382.60	1,343,830	212:45	4,195.61
GMATA[a]	1	111,905	7:28	85.48	384,400	42:15	361.72	1,343,831	254:46	1,739.50
PERF	1	111,960	8:49	211.60	384,488	52:32	522.91	1,345,128	320:36	3,325.42
Kmer-SSR	1	111,960	14:08	123.05	384,488	83:14	169.00	1,345,128	516:19	1,028.44
Kmer-SSR	12	111,960	5:28	321.95	384,488	28:20	353.96	1,345,128	205:19	1,251.40
SSRMMD	1	111,960	4:49	139.38	384,488	28:36	421.95	1,345,128	175:19	2,404.68
SSRMMD	6	111,960	1:08	466.19	384,488	6:46	1,044.88	1,345,128	43:40	5,972.30
SSRMMD	12	111,960	0:49	731.02	384,488	4:28	1,717.92	1,345,128	27:05	11,248.89

Note. SSRMMD, Simple Sequence Repeat Molecular Marker Developer; SSR, simple sequence repeat. [a] *Because GMATA and Kmer-SSR could not simultaneously mine different types of motifs in one task, these two programs were multiply performed to identify SSRs, then the time was added, and memory peak was selected as the maximum among all tasks.*

GMATA, and we found that the actual polymorphic SSR numbers of GMATA were 7,452 (rice), 21,675 (cotton), and 188,342 (wheat). GMATA had a high false-positive rate, being 36.86% (4,350 of 11,802), 30.06% (9,317 of 30,992), and 25.33% (63,902 of 252,244) for rice, cotton, and wheat, respectively, thus implying a defect in the GMATA pipeline. SSRMMD required markedly less time, especially in the wheat genome (**Table 3**). Unfortunately, we could not quantify memory consumption because GMATA and CandiSSR called numerous other programs and scripts for mining polymorphic SSRs. Furthermore, we compared the output of polymorphic SSRs between SSRMMD and two other software programs. As shown in **Figures 3D–F**, approximately 70.48% (rice), 37.11% (cotton), and 49.19% (wheat) of the SSRMMD outputs were novel in comparison with other two software programs.

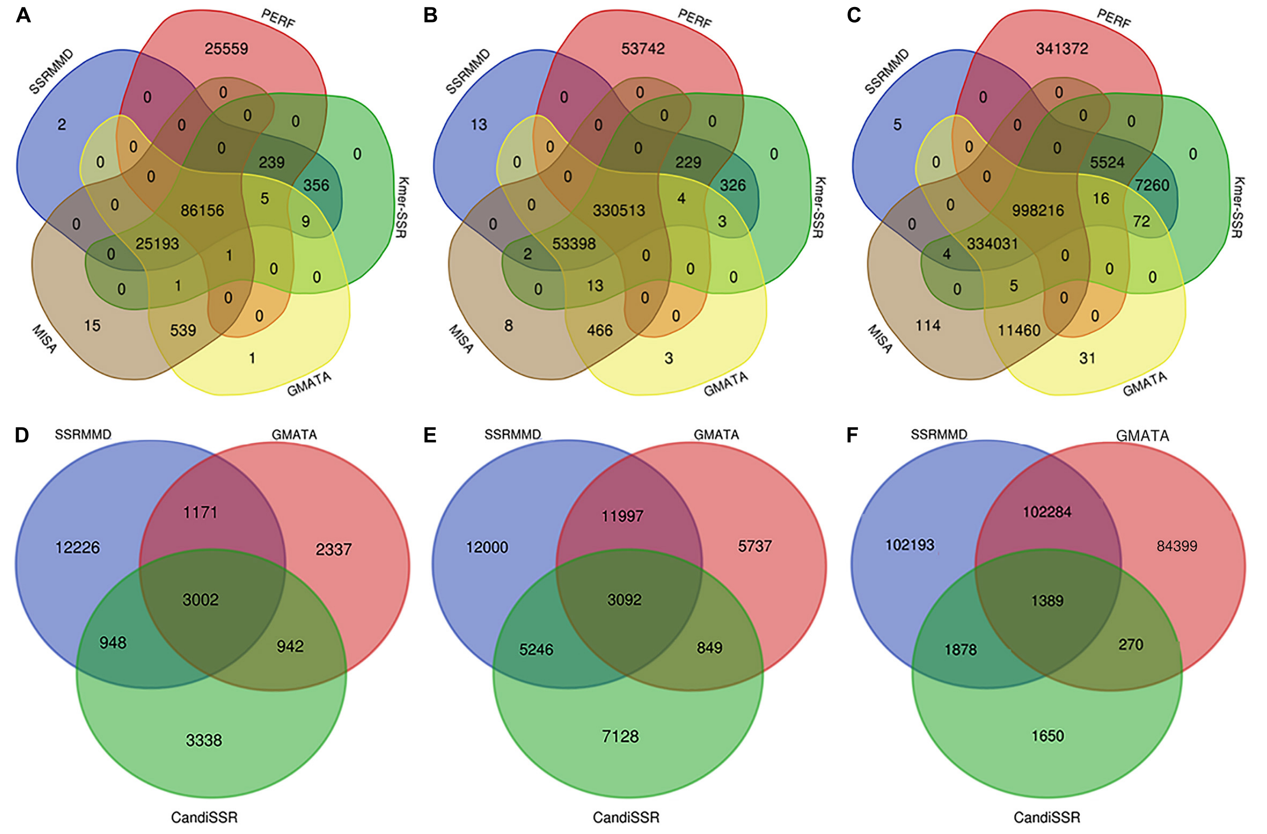

FIGURE 3 | Overlapping simple sequence repeats (SSRs) calculated by five programs [Simple Sequence Repeat Molecular Marker Developer (SSRMMD), PERF, Kmer-SSR, GMATA, and MISA], and overlapping polymorphic SSRs calculated by three programs (SSRMMD, GMATA, and CandiSSR). If the sequence name, start, and end position were the same in all programs, the SSR was considered overlapping. **(A)** Rice. **(B)** Cotton. **(C)** Wheat. **(D)** Rice. **(E)** Cotton. **(F)** Wheat. Notably, CandiSSR only mined 5,187 polymorphic SSRs of wheat chromosome 5D.

Experimental Verification of the Accuracy of Polymorphic Simple Sequence Repeats

To verify the accuracy of the output by SSRMMD, 80 pairs of polymorphic SSRs were randomly selected to identify polymorphisms in 10 wheat cultivars. As shown in **Figure 4** and **Supplementary Table 1**, 56 independent products were successfully amplified using 56 primer pairs. However, the remaining 24 primer pairs did not yield stable or clear bands, and the reasons may include the following: (1) we did not optimize the PCR amplification conditions for each primer pair and used a uniform annealing temperature for all primer amplifications; and (2) some primer designs were created in batches generated under uniform conditions, which may have defects. Forty-four (∼79%) among these 56 primer pairs revealed polymorphisms in CS and AK58, suggesting that SSRMMD had a high accuracy.

DISCUSSION

With rapid innovations in sequencing technologies, third-generation DNA markers such as single-nucleotide polymorphisms have become widely used (Zhou Z. et al., 2015; Yang et al., 2017). However, SSRs are still used in various genetic studies such as quantitative trait loci mapping, genotyping, genetic diversity, and marker-assisted selection because of their codominant inheritance, multi-allelic nature, and ease of amplification *via* PCR operation (Varshney et al., 2005; Ramu et al., 2009; Kaur et al., 2015). These features are not applicable to single-nucleotide polymorphisms. Therefore, development of SSR markers from diverse organisms still is important in biological studies.

In vitro SSR marker development based on the creation of a genomic library, screening of positive clones, and subsequent sequencing is time-consuming and expensive. Song et al. (2005) only developed 540 SSR flanking primer pairs in the wheat mapping study by *in vitro* methods. However, we easily obtained millions of SSR loci from the wheat CS genome using our *in silico* methods (**Table 1**). Undoubtedly, it is more rapid and economical to develop SSR markers by using bioinformatics tools and genotypic data, and *in silico* methods have gradually replaced *in vitro* methods.

Although numerous software programs have been developed for mining perfect SSRs from assembled sequences, the accuracy, speed, and flexibility of these programs need to be balanced

TABLE 3 | Performance comparison between SSRMMD and other two software programs for developing candidate polymorphic SSRs.

Organism	Rice (~0.39 Gb)			Cotton (~2.29 Gb)			Wheat (~14.23 Gb)		
Software	**SSRMMD**	**GMATA**	**CandiSSR[c]**	**SSRMMD**	**GMATA**	**CandiSSR**	**SSRMMD**	**GMATA**	**CandiSSR[d]**
Total number of SSR[a]	111,960	111,905	111,905	384,488	384,400	384,400	1,345,128	1,343,831	1,331,146
Number of candidate marker	68,242	34,667	8,230	292,307	166,813	16,315	572,023	477,531	129,461
Candidate marker rate (%)	60.95	30.98	7.35	76.02	43.40	4.24	42.53	35.54	9.73
Number of monomorphic SSR	50,895	22,865	0	259,972	135,821	0	364,279	225,287	0
Number of polymorphic SSR	17,347	11,802	8,230	32,335	30,992	16,315	207,744	252,244	129,461
Polymorphic rate (%)	15.49	10.55	7.35	8.41	8.06	4.24	15.44	18.77	9.73
Time (min:s)[b]	6:08	16:15	8,117:37	30:30	119:06	1,746:23	184:55	6,363:45	118,953:53

Note. SSRMMD, Simple Sequence Repeat Molecular Marker Developer; SSR, simple sequence repeat. [a]*Total number of SSR referred to Zhenshan97 (rice), TM1 (cotton), and CS (wheat).* [b]*Because GMATA could not simultaneously mine different types of motifs in one task, it was multiply performed to identify SSRs, and then the time was added.* [c]*Because CandiSSR could not normally calculate chromosome 8 of rice (the program stopped at the BLAST stage), the results of CandiSSR did not include chromosome 8.* [d]*Because CandiSSR spent numerous time to mine polymorphic SSRs of wheat, we only selected chromosome 5D (closest to the total SSR density of wheat) to estimate the number of polymorphic SSRs.*

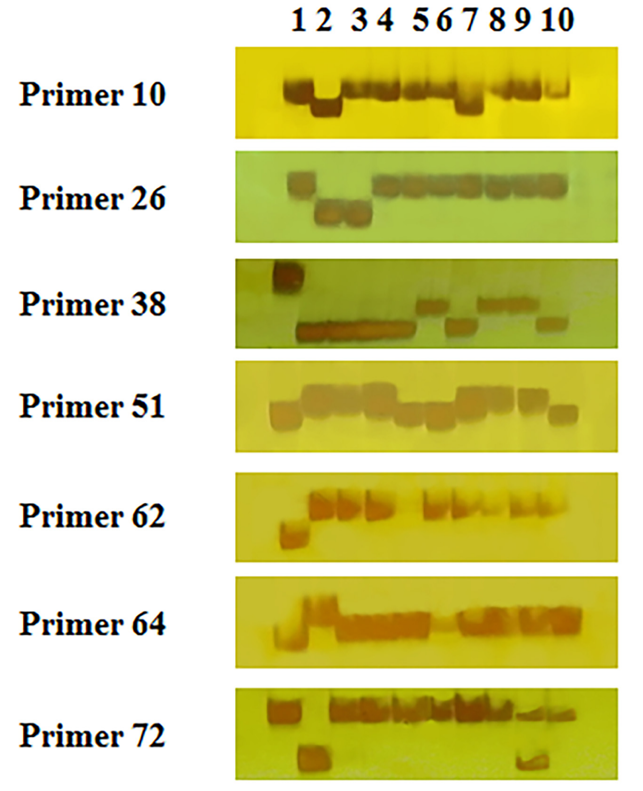

FIGURE 4 | Experimental validation of seven randomly selected polymorphic simple sequence repeats (SSRs) in 10 wheat genotypes. Lanes 1–10 were PCR products of CS, AK58, CM107, CN16, MM37, ZM012542, ZM000652, ZM018703, ZM003222, and ZM003284, respectively.

to suit the users' needs. SSRIT can completely mine SSRs (**Table 2**); however, when used only for mining a certain motif, such as tetra-nucleotide and hexa-nucleotide motifs for rice (Zhenshan97), SSRIT displayed 82.94% and 87.36% error rates, respectively (data not shown), implying a defect in the algorithm of SSRIT. In contrast, MISA and GMATA were inadequate for SSR mining. Although Kmer-SSR supported multi-threading, this support was inadequate, and this program can only be run on Linux. Furthermore, GMATA and Kmer-SSR had inflexible motif thresholds; these two programs needed to be performed in multiple tasks to identify SSRs. PERF was inflexible owing to its dependence on other modules (**Table 1**), and the computational speed was highly dependent on the motif thresholds, thus displaying a poor performance in the present tests. However, SSRMMD displayed an adequate performance in all aspects. SSRMMD completely mined credible SSRs (**Figures 3A–C**); furthermore, SSRMMD was rapid, especially for large genomes (**Table 2**); moreover, SSRMMD was flexible and did not rely on additional modules and could theoretically be run on any machine with PERL5 installed (**Table 1**).

The ever-increasing availability of plant and animal genomes and transcriptomes (Kersey et al., 2017; Marschall et al., 2018) has resulted in large data resources for developing polymorphic SSRs. In the past 3 years, certain software programs were reported for this purpose; however, they were all based on a complex pipeline and utilize numerous other software programs, increasing their dependence and decreasing their portability. For example, CandiSSR called numerous other programs during development, including MISA, Primer3, BLAST, and ClustalW (**Table 1**), among which BLAST was the most prominent reason for its low computational speed. Furthermore, the formatdb program in BLAST could not build an entire wheat genome library; hence, to complete the assessment, we artificially modified the source code of CandiSSR to enable it to normally perform computations with wheat. Similarly, GMATA depended on other programs when developing polymorphic SSRs, including e-PCR and Primer3. However, our proposed SSRMMD did not have these limitations, and SSRMMD used a stringent algorithm to assess the conservativeness and uniqueness of SSR flanking sequences. Performance assessments revealed that SSRMMD identified more novel polymorphic SSRs at extremely high speed (**Table 3** and **Figures 3D–F**).

Furthermore, we performed molecular biology assays for 80 randomly selected polymorphic SSRs of wheat to confirm the accuracy of SSRMMD, and we found that SSRMMD had an accuracy of up to 79% (**Figure 4** and **Supplementary Table 1**). We further examined 24 pairs of SSRs not yielding stable or clear bands, and we found that 19 of them were developed through GMATA. Similarly, 7 of the 12 non-polymorphic SSRs were developed using GMATA (data not shown), indicating that these inaccurate results may have been obtained from the wheat genome itself.

Nonetheless, Gao et al. (2019) recently used SSRMMD to assess the barley genome in quantitative trait loci mapping study, and they reported that SSRMMD has an excellent algorithm for mining polymorphic SSRs.

CONCLUSION

In this study, we proposed a rapid, accurate, and flexible algorithm named SSRMMD for mining perfect SSR loci and further mining candidate polymorphic SSRs in accordance with any size of assembled sequence. Our program can easily collect numerous polymorphic SSRs from genomes and transcriptomes of diverse organisms and will undoubtedly accelerate numerous types of genetic studies including those of quantitative trait loci mapping, genotyping, and genetic diversity.

DATA AVAILABILITY STATEMENT

All datasets presented in this study are included in the article/**Supplementary Material**.

ETHICS STATEMENT

The experiments comply with the ethical standards in the country in which they were performed.

AUTHOR CONTRIBUTIONS

XG and HS conducted data analysis and drafted the manuscript. SY, ZW, CL, SL, and JM performed the validation experiment and helped with data analysis. GC helped to draft the manuscript. TL participated in the design of the study and partially revised the manuscript. YL designed and coordinated this study and revised the manuscript. All authors have read and approved the final manuscript.

FUNDING

This study was supported by the National Natural Science Foundation of China (31771794 and 91731305), the National Key Research and Development Program of China (2016YFD0101004 and 2017YFD0100900), the outstanding Youth Foundation of the Department of Science and Technology of Sichuan Province (2016JQ0040), and the International Science and Technology Cooperation Program of the Bureau of Science and Technology of Chengdu China (No. 2015DFA306002015-GH03-00008-HZ).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2020.00706/full#supplementary-material

TABLE S1 | Molecular experiments to verify the accuracy of the output by SSRMMD in wheat.

REFERENCES

Altschul, S. F., Madden, T. L., Schffer, A. A., Jinghui, Z., Zheng, Z., Webb, M., et al. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein detabase search programs. *Nucleic Acids Res.* 25, 3389–3402. doi: 10.1093/nar/25.17.3389

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* 34, 943–948. doi: 10.1093/bioinformatics/btx721

Castelo, A. T., Martins, W., and Gao, G. R. (2002). TROLL–Tandem Repeat Occurrence Locator. *Bioinformatics* 18, 634–636. doi: 10.1093/bioinformatics/18.4.634

Chen, M., Tan, Z., and Zeng, G. (2011). MfSAT: detect simple sequence repeats in viral genomes. *Bioinformation* 6, 171–172. doi: 10.6026/97320630006171

Du, L. M., Zhang, C., Liu, Q., Zhang, X. Y., and Yue, B. S. (2017). Krait: an ultrafast tool for genome-wide survey of microsatellites and primer design. *Bioinformatics* 4, 681–683. doi: 10.1093/bioinformatics/btx665

Gao, S., Zheng, Z., Hu, H. Y., Shi, H. R., Ma, J., Liu, Y. X., et al. (2019). A novel QTL conferring fusarium crown rot resistance located on chromosome arm 6HL in barley. *Front. Plant Sci.* 10:1206. doi: 10.3389/fpls.2019.01206

Gramazio, P., Plesa, I. M., Truta, A. M., Sestras, A. F., and Sestras, R. E. (2018). Highly informative SSR genotyping reveals large genetic diversity and limited differentiation in European larch (*Larix decidua*) populations from Romania. *Turk. J. Agric. For.* 42, 165–175. doi: 10.3906/tar-1801-41

Guang, X. M., Xia, J. Q., Lin, J. Q., Yu, J., Wan, Q. H., and Fang, S. G. (2019). IDSSR: an efficient pipeline for identifying polymorphic microsatellites from a single genome sequence. *Int. J. Mol. Sci.* 20:3497. doi: 10.3390/ijms20143497

Kaur, S., Panesar, P. S., Bera, M. B., and Kaur, V. (2015). Simple sequence repeat markers in genetic divergence and marker-assisted selection of rice cultivars: a review. *Crit. Rev. Food Sci. Nutr.* 55, 41–49. doi: 10.1080/10408398.2011.646363

Kersey, P. J., Allen, J. E., Allot, A., Barba, M., Boddu, S., Bolt, B. J., et al. (2017). Ensembl Genomes 2018: an integrated omics infrastructure for non-vertebrate species. *Nucleic Acids Res.* 46, D802–D808. doi: 10.1093/nar/gkx1011

Levenshtein, V. (1966). Binary codes capable of correcting insertions and reversals. *Soviet Phys. Doklady* 10, 707–710.

Liu, J., Qu, J. T., Hu, K. H., Zhang, L., Li, J. W., Wu, B., et al. (2015). Development of genome wide simple sequence repeat fingerprints and highly polymorphic markers in cucumbers based on next-generation sequence data. *Plant Breed.* 134, 605–611. doi: 10.1111/pbr.12304

Liu, L., Qin, M., Yang, L., Song, Z., Luo, L., Bao, H., et al. (2016). A genome-wide analysis of simple sequence repeats in *Apis cerana* and its development as polymorphism markers. *Gene* 599, 53–59. doi: 10.1016/j.gene.2016.11.016

Liu, W. C., Xu, Y. T., Li, Z. K., Fan, J., and Yang, Y. (2019). Genome-wide mining of microsatellites in king cobra (*Ophiophagus hannah*) and cross-species development of tetranucleotide SSR markers in Chinese cobra (*Naja atra*). *Mol. Biol. Rep.* 46, 6087–6098. doi: 10.1007/s11033-019-05044-7

Marschall, T., Marz, M., Abeel, T., Dijkstra, L., Dutilh, B. E., Ghaffaari, A., et al. (2018). Computational pan-genomics: status, promises and challenges. *Brief. Bioinform.* 19, 118–135. doi: 10.1093/bib/bbw089

Metz, S., Manuel, C. J., Eva, R., Federico, G., and Patricia, A. (2016). FullSSR: microsatellite finder and primer designer. *Adv. Bioinform.* 2016, 1–4. doi: 10.1155/2016/6040124

Mudunuri, S. B., and Nagarajaram, H. A. (2007). IMEx: imperfect microsatellite extractor. *Bioinformatics* 23, 1181–1187. doi: 10.1093/bioinformatics/btm097

Nachimuthu, V. V., Muthurajan, R., Duraialaguraja, S., Sivakami, R., and Sabariappan, R. (2015). Analysis of population structure and genetic diversity in rice germplasm using SSR markers: an initiative towards association mapping of agronomic traits in *Oryza Sativa*. *Rice* 8:30. doi: 10.1186/s12284-015-0062-5

Needleman, S. B., and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequences of two proteins. *J. Mol. Biol.* 48, 443–453. doi: 10.1016/0022-2836(70)90057-4

Pickett, B. D., Karlinsey, S. M., Penrod, C. E., Cormier, M. J., Ebbert, M. T. W., Shiozawa, D. K., et al. (2016). SA-SSR: a suffix array-based algorithm for exhaustive and efficient SSR discovery in large genetic sequences. *Bioinformatics* 32, 2707–2709. doi: 10.1093/bioinformatics/btw298

Pickett, B. D., Miller, J. B., and Ridge, P. G. (2017). Kmer-SSR: a fast and exhaustive SSR search algorithm. *Bioinformatics* 33, 3922–3928. doi: 10.1093/bioinformatics/btx538

Qin, H., Chen, M., Yi, X., Bie, S., Zhang, C., Zhang, Y., et al. (2015). Identification of associated SSR markers for yield component and fiber quality traits based on frame map and upland cotton collections. *PLoS One* 10:e0118073. doi: 10.1371/journal.pone.0118073

Ramu, P., Kassahun, B., Senthilvel, S., Kumar, C. A., Jayashree, B., Folkertsma, R. T., et al. (2009). Exploiting rice–sorghum synteny for targeted development of EST-SSRs to enrich the sorghum genetic linkage map. *Theor. Appl. Genet.* 119, 1193–1204. doi: 10.1007/s00122-009-1120-4

Silva, L. R. D., Lopes, M. W. J., Souza, R. T. D., and Castanheira, B. D. (2015). ProGeRF: proteome and genome repeat finder utilizing a fast parallel hash function. *BioMed Res. Int.* 2015, 1–9. doi: 10.1155/2015/394157

Song, Q. J., Shi, J. R., Singh, S., Fickus, E. W., Costa, J. M., Lewis, J., et al. (2005). Development and mapping of microsatellite (SSR) markers in wheat. *Theor. Appl. Genet.* 110, 550–560. doi: 10.1007/s00122-004-1871-x

Temnykh, S. (2001). Computational and experimental analysis of microsatellites in rice (*Oryza sativa* L.) : frequency, length variation, transposon associations, and genetic marker potential. *Genome Res.* 11, 1441–1452. doi: 10.1016/j.ces.2004.03.045

Thiel, T., Michalek, W., Varshney, R., and Graner, A. (2003). Exploiting EST databases for the development and characterization of gene-derived SSR-markers in barley (*Hordeum vulgare* L.). *Theor. Appl. Genet.* 106, 411–422. doi: 10.1007/s00122-002-1031-0

Thompson, J. D., Gibson, T. J., and Higgins, D. G. (2002). Multiple sequence alignment using ClustalW and ClustalX. *Curr. Protoc. Bioinformatics* 2:3. doi: 10.1002/0471250953.bi0203s00

Untergasser, A., Cutcutache, I., Koressaar, T., Ye, J., Faircloth, B. C., Remm, M., et al. (2012). Primer3–new capabilities and interfaces. *Nucleic Acids Res.* 40:e115. doi: 10.1093/nar/gks596

Varshney, R. K., Graner, A., and Sorrells, M. E. (2005). Genic microsatellite markers in plants: features and applications. *Trends Biotechnol.* 23, 48–55. doi: 10.1016/j.tibtech.2004.11.005

Wang, L., Zhang, Y., Zhu, X., Zhu, X., and Zhang, X. (2017). Development of an SSR-based genetic map in sesame and identification of quantitative trait loci associated with charcoal rot resistance. *Sci. Rep.* 7:8349. doi: 10.1038/s41598-017-08858-2

Wang, X., Lu, P., and Luo, Z. (2013). GMATo: a novel tool for the identification and analysis of microsatellites in large genomes. *Bioinformation* 9, 541–544. doi: 10.6026/97320630009541

Wang, X., and Wang, L. (2016). GMATA: an integrated software package for genome-scale SSR mining, marker development and viewing. *Front. Plant Sci.* 7:1350. doi: 10.3389/fpls.2016.01350

Wang, X., Yang, S., Chen, Y., Zhang, S., Zhao, Q., Li, M., et al. (2018). Comparative genome-wide characterization leading to simple sequence repeat marker development for *Nicotiana*. *BMC Genomics* 19:500. doi: 10.1186/s12864-018-4878-4

Xia, E., Yao, Q., Zhang, H., Jiang, J., Zhang, L., and Gao, L. (2016). CandiSSR: an efficient pipeline used for identifying candidate polymorphic ssrs based on multiple assembled sequences. *Front. Plant Sci.* 6:1171. doi: 10.3389/fpls.2015.01171

Xu, J., Liu, L., Xu, Y., Chen, C., Rong, T., Ali, F., et al. (2013). Development and characterization of simple sequence repeat markers providing genome-wide coverage and high resolution in maize. *DNA Res.* 20, 497–509. doi: 10.1093/dnares/dst026

Yang, N., Xu, X. W., Wang, R. R., Peng, W. L., Cai, L., Song, J. M., et al. (2017). Contributions of *Zea mays* subspecies mexicana haplotypes to modern maize. *Nat. Commun.* 8:1874. doi: 10.1038/s41467-017-02063-5

Zalapa, J. E., Cuevas, H., Zhu, H., Steffan, S., Senalik, D., Zeldin, E., et al. (2012). Using next-generation sequencing approaches to isolate simple sequence repeat (SSR) loci in the plant sciences. *Am. J. Bot.* 99, 193–208. doi: 10.3732/ajb.1100394

Zhang, L., Cai, R., Yuan, M., Tao, A., Xu, J., Lin, L., et al. (2015). Genetic diversity and DNA fingerprinting in jute(*Corchorus* spp.) based on SSR markers. *Crop J.* 3, 416–422. doi: 10.1016/j.cj.2015.05.005

Zhang, Z., Deng, Y., Tan, J., Hu, S., Yu, J., and Xue, Q. (2007). A genome-wide microsatellite polymorphism database for the indica and japonica rice. *DNA Res.* 14, 37–45. doi: 10.1093/dnares/dsm005

Zhou, R., Wu, Z., Cao, X., and Jiang, F. L. (2015). Genetic diversity of cultivated and wild tomatoes revealed by morphological traits and SSR markers. *Genet. Mol. Res. GMR* 14, 13868–13879. doi: 10.4238/2015.october.29.7

Zhou, Z., Jiang, Y., Wang, Z., Gou, Z., Lyu, J., Li, W., et al. (2015). Resequencing 302 wild and cultivated accessions identifies genes related to domestication and improvement in soybean. *Nat. Biotechnol.* 4, 408–414. doi: 10.1038/nbt.3096

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.