LTR_retriever: a highly accurate and sensitive program for identification of LTR retrotransposons

Shujun Ou and Ning Jiang*

Department of Horticulture, Michigan State University, East Lansing, MI, 48824, USA

ORCID IDs: 0000-0001-5938-7180 (S.O.); 0000-0002-2776-6669 (N.J.)

* To whom correspondence should be addressed. Tel: +1 (517) 353-0381; Fax: +1 (517) 353-0890; Email: jiangn@msu.edu

ABSTRACT

Long terminal repeat retrotransposons (LTR-RTs) are prevalent in most plant genomes. Identification of LTR-RTs is critical for achieving high-quality gene annotation. The sequences of LTR-RTs are diverse among species, yet the structure of the element is well conserved. Based on the conserved structure, multiple programs were developed for *de novo* identification of LTR-RTs. Most of these programs are associated with low specificity, and excessive curation is required since false positives are very detrimental for downstream analyses. Here we report LTR_retriever, a multithreading empowered Perl program that identifies LTR retrotransposons and generates high-quality LTR libraries from genomes with various assembly qualities. LTR_retriever demonstrated significant improvements by achieving high levels of sensitivity, specificity, accuracy, and precision, which are 91.7%, 96.9%, 95.7%, and 90.0%, respectively, in rice (*Oryza sativa*). Besides LTR-RTs with canonical ends (TG..CA), LTR_retriever also identifies non-canonical LTRs accurately. A scan of 50 public plant genomes identified seven non-canonical types of LTRs. LTR_retriever is also compatible with long-read sequencing technologies. With 40k self-corrected PacBio reads equivalent to 4.5X of genome coverage in Arabidopsis, the quality of constructed LTR library surpasses that constructed from the genome alone. LTR_retriever has demonstrated the highest performance with great flexibility for automatically retrieving LTR-RTs.

Keywords: LTR retrotransposon, LTR_retriever, repeat library, plant genome annotation, evolution

INTRODUCTION

Transposable elements (TEs) are ubiquitous interspersed repeats in most sequenced eukaryote genomes (1-3). According to their transposition schemes, TEs are categorized into two classes. Class I TEs (retrotransposons) use RNA intermediates with a “copy and paste” transposition mechanism (2-5). Class II TEs (DNA transposons) use DNA intermediates with a “cut and paste” mechanism (2-4). Depending on the presence of long terminal repeats (LTRs), Class I TEs are further classified as LTR retrotransposons (LTR-RTs) and non-LTR retrotransposons, including short interspersed transposable elements (SINEs) and long interspersed transposable elements (LINEs) (2, 3). For simplicity, TEs other than LTR-RT, including both non-LTR retrotransposons and DNA transposons, are called non-LTR in this study. In plants, LTR-RTs contribute significantly to genome size expansion due to their high copy number and large size (6-11). For example, retrotransposons contribute to approximately 75% to the size of the maize (*Zea mays*) genome (8). In *Oryza australiensis*, a wild relative of rice (*O. sativa*), the amplification of three families of LTR retrotransposons is attributed to the genome size doubling within the last 3 MY (12). Understanding genome size evolution by studying the amplification of LTR-RT is demonstrably powerful (8, 9, 13, 14). The amplification and elimination of LTR-RTs has shaped genome landscapes (15, 16), thereby affecting the expression of adjacent genes (3, 17-19).

An intact LTR-RT carries an LTR at both termini (**Figure 1A**). The LTR regions usually span 85-5000 base pairs (bp) with intra-element sequence identity $\geq$ 85%. In plants, LTRs are typically flanked by 2bp palindromic motifs (**Figure 1A**), commonly 5'-TG..CA-3' (3). The sequence between the 5' and 3' LTR is defined as the internal region, and usually ranges from 1000-15000bp (**Supplementary Figure S1**). To confer transposition activities, the internal region of an autonomous LTR should contain a primer binding site (PBS), a polypurine tract (PPT), a *gag* gene (i.e., encoding structural proteins for reverse transcription), and a *pol* gene (i.e., functioning as protease, reverse transcriptase, and integrase) (20). Depending on the order of protein domains in the *pol* gene, intact LTR-RTs can be further categorized into two families called *gypsy* and *copia* (3). If the internal region does not contain any open reading frames (ORFs), e.g., reverse transcriptase genes, the belonging LTR-RT is unable to transpose independently, and it relies on the transposition-related proteins from other autonomous LTR-RTs (2, 20). There are two groups of non-coding LTR-RTs: terminal-repeat retrotransposon in miniature (TRIM) (2, 20, 21) and large retrotransposon derivatives (LARD) (20, 22). These non-coding LTR-RTs are distinguished by their average length: TRIMs are < 1 kb and LARDs are 5.5-9kb (2, 20).

The insertion of an LTR-RT is accompanied by the duplication of a small piece of sequence immediately flanking the element, which is called target site duplication (TSD, 4-6bp in length) (**Figure 1A**). There are many mechanisms that can introduce mutations to a newly transposed LTR-RT. Due to the sequence similarity between the long direct repeat of an LTR-RT, intra-element recombination can occur, leading to the elimination of the internal region and the formation of a solo-LTR (**Figure 1C**). New LTR-RT insertions can be silenced by methylation and chromatin modification as a genomic mechanism to suppress expression (2, 17, 23). Silenced elements have less selection constraint and accumulate more mutations including deletions, resulting in truncated LTR-RTs (**Figure 1B**). Truncated LTR-RT could also be the product of illegitimate recombination which generates deletions and translocations (3, 24). LTR-RTs often insert into other LTR-RTs, generating nested LTR-RTs (**Figure 1D**) (3, 24, 25). Given these mutation mechanisms, intact elements only contribute a small fraction of all LTR-RT related sequences in a genome. If the required structural components are altered, i.e., mutated, truncated, and nest-inserted by other TEs (**Figure 1**), the LTR element becomes non-autonomous and is difficult to identify using structural information.

Although the structure of LTR-RT is conserved among species, their sequences are not conserved except among closely related species. Particularly, substantial sequence diversity is observed within the long terminal repeat region. Therefore, LTR-RTs are usually not identified based on sequence homology. Due to the lack of nucleotide sequence similarity among species, constructing a species-specific LTR library (i.e., exemplars) is essential for identification of all LTR-RT related sequences in a newly sequenced genome.

Computational identification of LTR-RTs based on structural features has been implemented multiple times. Such methods are often used jointly to maximize power in genome annotation projects. However, prediction results from these tools often only partially overlap (26). Many factors could be responsible for the discrepancy of prediction programs. The discrepancy includes the differences in defining the LTR structure in the program and the different implementation of these methods. LTR_STRUC was one of the earliest developments of genome-wide LTR identification programs (27). However, LTR_STRUC is dependent on Windows systems, limiting its scalability and computational potency. LTR_finder (28) and LTRharvest (29) are by far the most sensitive programs in finding LTRs. Nevertheless, these programs suffer from reporting large numbers of false positives (30). Like LTR_STRUC, MGEScan-LTR (31) is one of the initial LTR searching programs (31). Its recent update on the web-based platform allows wider usage (32), but is still associated with the issue of false positive identifications.

The development of high-throughput sequencing technologies has led to more sequenced genomes. However, many of their assemblies were compromised by highly repetitive sequence regions. As the most sizeable content of plant genomes, the assembly of LTR-RTs is typically compromised due to the collapse of short reads from such regions. The fragmented and misassembled repetitive sequences could lead to further error propagation in downstream genome annotation. Unfortunately, most of the current programs are not well adapted to the nature of draft genomes.

In this study, we introduce LTR_retriever, a novel tool for identification of LTR-RTs. This package efficiently removes false positives from initial software predictions. We benchmarked the performance of LTR_retriever with existing programs using the well assembled and annotated rice genome (33). Our results indicated that LTR_retriever achieved very high specificity, accuracy, and precision without significantly sacrificing sensitivity, hence significantly outperforming existing methods. A further test using high-quality assembled and annotated monocot and dicot model genomes, e.g., maize, sacred lotus (*Nelumbo nucifera*), and Arabidopsis (*Arabidopsis thaliana*), also demonstrated excellent performance in identification of LTR-RTs. In addition, we implemented a module to accurately search for non-canonical LTR-RTs that featured non-TGCA motifs of LTR regions. A search in 50 published genomes revealed the rare nature of non-canonical LTR-RTs, although some non-TGCA motifs could be relatively abundant. Finally, we demonstrated the feasibility of making high-quality LTR libraries from self-corrected PacBio reads.

MATERIALS AND METHODS

De novo prediction of LTR-RTs can produce large amounts of false positives. To detect and filter out non-LTR sequences and obtain high-quality LTR-RT exemplars (representative LTR-RT sequences), we developed eight modules with adjustable parameters in LTR_retriever (**Figure 2**). Please note the module number is only for convenience of description, not implying the order of implementation in the package.

Module 1: Filtering of tandem repeats, elements with sequencing gaps and unusual size

Gap sequences represent the most uncertainty of a genome assembly. Particularly, gaps in a repetitive sequence are more likely associated with misassembly (34). In this module, LTR candidates that contain gaps more than the threshold (default *10*bp) are excluded. In addition, Tandem Repeats Finder (TRF) (35) is used to identify tandem repeat contaminations with parameters "*2 7 7 80 10 1000 2000 -ngs -h*". LTR-RT candidates containing substantial tandem repeats are excluded.

Extremely long or short LTR candidates are also likely false positives. To control the size of candidate LTR-RTs, the minimum and maximum length of the internal region is set to *100*bp and *15,000*bp, respectively, which covers most of cases (**Supplementary Figure S1)**. The LTR:internal length ratio is set to a minimum of *0.05* and a maximum of *50* to avoid the case of a short LTR with exceptionally long internal region or vice verser. These settings are flexible enough to identify special LTR-RT like TRIM (terminal-repeat retrotransposon in miniature) while stringent enough to exclude many false positives.

Module 2: Coarse-grained boundary mapping

To obtain the precise element boundaries, coarse-grained boundary mapping was implemented using BLAST+ (36) followed by boundary adjustment based on the 2bp palindromic motif. The alignment between 5' and 3' repeat regions is performed by *blastn* using default parameters. Then the original terminal motif predicted by the input software (e.g., LTRharvest) is used to search for the potential boundaries between the LTR region and the internal region. The 2bp motif search is limited to within 100bp to the original boundary for the identification of potential new boundary. An LTR-RT candidate is excluded if any of the following conditions apply: 1) no self-alignment is found within the candidate sequence (indicating the absence of the long terminal repeat); 2) the self-alignment is shifted more than 100bp compared to the original coordinate (implying gross prediction errors); and 3) eight or more alignment pairs are found (indicating heavily nested insertions). After the coarse-grained adjustment, most of the internal boundaries are corrected. However, a small percentage of boundaries may contain a 1-2bp shift from the actual boundary.

Module 3: Structure filtering and fine-grained boundary mapping

If the terminal regions of a *bonafide* LTR element were subjected to alignment, only the LTR region could be aligned. For most false positives, extended alignments were found beyond the "LTR" region (**Figure 1E**), in that it is unlikely the boundary of "LTR" coincides the boundary of other types of repeats. Moreover, structural features like TSDs and terminal motifs were frequently missing or not immediately adjacent to the ends of false LTRs. To detect possible alignment beyond LTR regions, the 50bp sequences flanking each of the direct repeat and 10bp from the repeat region of the candidate were retrieved. These 60bp sequences upstream of the 5' direct repeat and 3' direct repeat (regions "a" and "c") are aligned against each other using *blastn*, and the sequences from downstream of the direct repeat (regions “b” and “d”) were processed similarly (**Figure 1E**). The sequences are considered aligned if 60% or more of one sequence (i.e., 36bp) has a minimum of 60% identity to the other. In the case that any flanking sequences of the direct repeat are aligned, the LTR-RT candidate is considered a false positive.

To identify the TSD and the motif together, 11bp sequences consisting of 3bp of the element end and 8bp of the flanking sequence from both the 5' and 3' ends of the LTR-RT are extracted. The canonical structure, with a 5bp TSD immediately connected with the 5'-TG..CA-3' (abbreviated as TGCA) motif (**Figure 1A**), is preferentially recognized in the exhaustive search within the 11bp element ends. If the TGCA motif is not present, the longest k-mer between the two element ends is obtained as the TSD candidate. It is possible that the TSD pair has extended identity to their flanking sequence by chance, resulting in a longer TSD candidate. For such cases, the structure that a 5bp TSD immediately connected to the known motifs is searched within the TSD candidate, which can effectively recognize the real TSD and motif. In the joint TSD-motif search, the TGCA motif is searched preferentially. If the TGCA motif does not exist, seven high confidence non-canonical motifs based on curation results are subsequently searched. These motifs are TGCT, TACA, TACT, TGGA, TATA, TGTA, and TCCA. Alternatively, custom motif lists are allowed. Finally, the accurate coordinates of an LTR candidate are defined based on the coordinate information of TSDs and motifs. For those LTR-RT candidates without any extended terminal alignment, the ones with identified TSD and motif are labeled as “pass”, otherwise “truncated”.

Module 4: Insertion time estimation

Since the direct repeat of an LTR-RT is identical upon insertion, the divergence between the LTR of an individual element reflects the time of the insertion. Based on the neutral theory, the divergence time between the direct repeats can be estimated as $\mathbf{T=K/2\mu}$, where K is the divergence rate and μ is the neutral mutation rate (37). Sequence identity (%) between the 5' and 3' direct repeats of an LTR candidate is approximated using *blastn*, so the proportion of sequence differences is calculated as **d=100%-identity%**. Then K is estimated by the Jukes-Cantor model for non-coding sequences with **K=-3/4*ln(1-d*4/3)** (38). The rice mutation rate of 1.3×10^{-8} mutations per site per year (39) is set as default but customizable.

Module 5: classification and strand phasing

Depending on the order of protein domains in the internal region, LTR-RTs can be categorized into two families, i.e., *gypsy* and *copia* (5). In this module, the profile hidden Markov model (pHMM) was applied to identify conserved protein domains in each LTR candidate sequence. A six-frame translation using the entire sequence of an LTR candidate is performed to recover any coding potentials. The translated sequences are then subjected to pHMM search using *hmmsearch* (40) from the HMMER/3.1b2 package (http://hmmer.org/) with parameters of “*-E 0.05 --domE 0.05*”.

To construct a compact and efficient pHMM database for LTR_retriever, the entire protein family collection Pfam 28.0 (41) was used to search against the manually curated rice TE database (**Supplementary Methods**). Matched pHMMs were counted based on TE families and those with at least 3 matches were retained as TE-related. Those pHMMs specifically matched to other TE elements, e.g., DNA TE, LINE, and Helitron, were further tagged as other-TE pHMMs. For the pHMMs specifically matched to LTR-RTs, a further categorization based on *copia*, *gypsy*, and unknown was applied for the identification of LTR families. The unknown category represents the ambiguous type of pHMMs found in the element, which could belong to either *copia* or *gypsy*. Strand information (plus/minus) about the pHMM matching is used to phase the candidate sequence. LTR candidates on the minus strand are transformed to reverse complement sequences to facilitate further use.

A BLAST search scheme is also applied to identify non-LTR coding sequences. The *blastx* program is called to perform protein sequence search on LTR-RT candidates using a non-LTR database with parameters "*-word_size 3 -max_target_seqs 10*". The database contains all kinds of coding sequence except LTR-RT, including DNA TE transposases, LINE retrotransposons, and the MAKER2 database of proteins in plants (42). To exclude ambiguous alignments, hits with an *e-value* higher than *0.001*, identity less than *30%*, and/or nucleotide alignment less than *90*bp are not considered as real alignments. LTR candidates that have more than *1,000*bp sequence or *30%* of the whole sequence aligned to the non-LTR protein database are considered false positives and discarded. Partial alignments to the non-LTR protein database are removed and non-aligned sequences are retained.

Module 6: Restore truncated LTRs and eliminate nested insertions

This module was designed for restoring high-quality LTR sequences from slightly truncated LTR elements and to eliminate nested insertions for library construction. LTR candidates labeled "truncated" from **Module 3** are masked using a special library constructed from intact LTR-RTs and high-confidence non-LTRs. Intact LTR-RTs are candidates marked as "pass" from **Module 3**. A list of high-confidence non-LTR sequences is initially collected from the "false" category identified by LTR_retriever. Further, such sequences are required to be annotated as "non-LTR" using **Module 5** and do not carry the "TGCA" motif to minimize the possibility of containing any LTR-related sequences. The special library is used to exclude known LTR-RTs and non-LTR sequences in the truncated LTR-RT sequence pool. Filtered truncated LTR-RTs are retained as LTR-RT sequences. Users can decide whether retaining truncated LTR-RT sequences (default) or keeping only intact LTR-RT depending on their research goals.

To eliminate nested insertions in internal regions of candidate LTR elements, LTR sequences are used to mask internal regions of LTR-RTs. Sequence masking is performed by *RepeatMasker* (v4.0.0) (http://repeatmasker.org/) with parameters "*-q -no_is -norna -nolow -div 40 -cutoff 225*". Candidate sequences masked by *80%* or more by the special library are excluded, while the retained sequences are saved with the elimination of the masked portion.

Module 7: recovery of non-TGCA motif-containing LTRs

Non-TGCA LTR-RTs are rare but detectable. This two-step module was developed to identify additional high-confidence non-canonical LTRs. The first step is to identify LTR-RT exemplars with the canonical “TGCA” motif using **Modules 1-6**. The input file for this step can be obtained from LTRharvest (29) with “-motif TGCA” parameter or other methods such as LTR_finder (28) and MGEScan-LTR (31) with default parameters. The second step of this module uses an additional input obtained from LTRharvest without specifying the “-motif” parameter. In the additional input, elements with both the “TGCA” motif and non-“TGCA” motifs are collected. LTR-RTs with the “TGCA” motif from the extra input are masked by the canonical LTR-RT exemplars obtained in the first step using RepeatMasker. The non-masked sequences are considered non-TGCA candidates and screened using **Modules 1-6**. The remaining LTR candidates, with motifs starting with “T” and with 5bp TSDs, are recognized as high-quality non-TGCA LTR-RTs. From the first step, non-TGCA LTR-RTs can be also detected, which are congregated into the non-TGCA LTR collection. To retain sensitivity, the redundancy of the non-TGCA LTR collection is not reduced and is not processed by **Module 8**.

Further filtering steps were applied on LTR candidates obtained from 50 plant genomes to ensure the non-TGCA motifs were unequivocal. First, the 5' and 3' repeat with an extra 10bp flanking sequences were extracted and aligned to each other using *blastn* with default parameters. Then, the alignment was examined to see if it extended beyond the motif or if there were multiple alignment hits using custom Perl scripts. Extended alignments indicated ambiguous endings and multiple alignment hits indicated nested-insertions, thus, such non-canonical LTR-RT candidates were excluded.

Module 8: library construction

This module was designed to identify and remove redundant sequences to generate non-redundant LTR-RT exemplars as the LTR library. To improve compatibility, this module is engineered to utilize either *blastclust* from BLAST (43) or *cd-hit-est* from CD-HIT (44) to generate non-redundant LTR-RT exemplars. Before clustering, LTR elements are split into 5' LTRs, 3' LTRs, and internal regions. For systems with BLAST installed, the parameters are set to “*-L 0.9 -b T -S 80*” for clustering sequences with at least 90% of length overlapped with more than 80% of identity. In a cluster, the sequence from the element with highest similarities between LTR regions (most recent insertions) is preferentially chosen as the exemplar sequence. The clustering procedure is set to iterate until no more sequence clusters can be found with a maximum of 10 iterations. For systems with CD-HIT installed, the parameters are set to “*-c 0.8 -G 0.8 -s 0.9 -aL 0.9 -aS 0.9*”. While both methods provided excellent clustering performance, CD-HIT is more efficient (44).

Implementation of LTR_retriever

LTR_retriever is a command line program developed based on Perl. The package supports multi-threading, which was achieved using the Semaphore module in Perl, and multithreading requests are passed to dependent packages. LTR_retriever takes genomic sequences in the FASTA format as input. The program can handle fragmentized and gapped regions, which is a benefit when annotating draft genomes. LTR_retriever has been optimized for plant genomes; however, its parameters can be adjusted for the genomes of other organisms. The output of the program contains a set of high-quality, comprehensive LTR exemplars (library), which can be used to identify or mask LTR sequences using

RepeatMasker. Additionally, a summary table that includes LTR-RT coordinates, length, TSDs, motifs, insertion time, and LTR families is produced. The program also provides gff3 format output, which is convenient for downstream analysis.

Genomes and sequences

The initial BAC sequences of "Nipponbare" were downloaded from the Rice Genome Research Program (http://rgp.dna.affrc.go.jp) for our early efforts to construct the rice TE library. The rice reference genome "Nipponbare" release 7 was downloaded from the MSU Rice Genome Annotation Project (http://rice.plantbiology.msu.edu) (45). The sacred lotus genome was downloaded from the National Center for Biotechnology Information (NCBI) under the project ID "AQOG01". The Arabidopsis reference genome "Columbia" version 10 was downloaded from The Arabidopsis Information Resource (TAIR) (www.arabidopsis.org) (46). The maize genome "B73" version AGPv4 was downloaded from Ensembl Plants release 34 (9). An additional of 46 plant genomes were downloaded from Phytozome v11 (47) (**Supplementary Methods**).

The Arabidopsis "L*er*-0" genome was sequenced and assembled by Pacific Biosciences using the PacBio RS II platform and the P5-C3 chemistry. The assembly is about 131 MB with contig N50 6.36 MB (https://github.com/PacificBiosciences/DevNet). A total of 184,318 self-corrected reads were also downloaded, which is about 2.69 GB with an average read length of 14.6kb and sequence error rate < 2%, covering 20.58 X coverage of the genome.

Standard LTR libraries

In this study, LTR libraries from four genomes (rice, maize, Arabidopsis, and sacred lotus) were used to evaluate the performance of LTR_retriever as well as existing tools. The TE database of maize was downloaded from the Maize TE database (http://maizetedb.org). The Arabidopsis repeat library athrep.ref was downloaded from Repbase (48). The LTR libraries for rice and sacred lotus were manually curated in the Jiang Lab (**Supplementary Methods, Supplementary sequence files**).

Benchmark programs and parameters

LTR_STRUC (27) was obtained from Mr. Vinay Mittal (vinaykmittal@gatech.edu) via personal communications. No parameter settings were available for LTR_STRUC. LTRharvest (29) is part of the GenomeTools v1.5.4 (49). Parameters for running LTRharvest were optimized based on our experience, which are "*-minlenltr 100 -maxlenltr 7000 -mintsd 4 -maxtsd 6 -motif TGCA -motifmis 0 -similar 90 -vic 10 -seed 20*". For other LTRharvest analysis, default parameters with "*-motif TGCA -motifmis 1*" were also used in the non-TGCA LTR analysis and the PacBio read analysis to allow mutations or sequencing errors. Optimized parameters were also applied to MGEScan-LTR (31) and LTR_finder (28). The modified version of MGEScan-LTR was obtained from the DAWG-PAWS package (50) and was run with parameter settings "*-min-mem=20 -mim-dist=1000 -max-dist=15000 -min-ltr=50 -max-ltr=7000 -min-orf=200*". LTR_finder v1.0.6 was run with parameter settings "*-D 15000 -d 1000 -L 7000 -l 100 -p 20 -M 0.9*".

Based on the annotation using the standard LTR library, the whole genome was categorized into four parts which are true positive (TP, LTR was identified), false negative (FN, LTR was not identified), false positive (FP, non-LTR was identified as LTR), and true negative (TN, non-LTR was not identified as LTR). Four metrics were used to evaluate the performance of LTR_retriever and its counterparts, which are sensitivity, specificity, accuracy, and precision defined as follows.

Sensitivity = TP/(TP+FN)

Specificity = TN/(FP+TN)

Accuracy = (TP+TN)/(TP+TN+FP+FN)

Precision = TP/(TP+FP)

Sensitivity, specificity, accuracy, and precision of each test were calculated using genomic sequence lengths by custom Perl scripts.

RESULTS

Recovery of LTR elements based on structural features has been implemented in multiple packages. However, a high level of false positives is a key issue. It is possible to reduce false positives by defining more stringent parameters such as high LTR similarity, intermediate LTR length, and “TGCA” motif (**Figure 3, Supplementary Table S1**). Unfortunately, the level of false negatives becomes high when more stringent parameters are applied (**Figure 3, Supplementary Table S1**). The trade-off between sensitivity and specificity cannot be minimized by merely adjusting parameters of existing tools (**Figure 3, Supplementary Table S1**). To establish efficient filters, it is essential to understand the fundamental differences between true LTR elements and false positives. In this study, we employed four statistical metrics (sensitivity, specificity, accuracy, and precision) to evaluate the performance of LTR-RT recovery programs (**Materials and Methods**).

Features of LTR false positives and solutions

In genome assembling practices, one of the most difficult tasks is to assemble highly repetitive regions. Even in the best-assembled genomes, there are still gaps to be filled. In assemblies of non-overlapped scaffolds, sequence space is manually added based on their inferred order. The gap length could be scaled to genetic distance or sometimes an arbitrary length. For a sequence with gaps, it is not uncommon that genome assemblers mistakenly join two similar sequences that belong to different transposable elements from the same family. Under these situations, the ambiguous sequence replaced by gaps is much less reliable than continuous sequence.

Tandem repeats are locally duplicated sequences of two or more bases such as centromere repeats and satellite sequences (35). Although it is possible that an LTR element carries small portions of tandem repeats, it becomes an LTR false positive when the majority sequence of an LTR-RT candidate consists of tandem repeats including low complexity sequences. We deploy **Module 1** in LTR_retriever to eliminate candidates that are consisted of gap and tandem repeats. **Module 1** also controls sequence length in consideration of both extremely long and short LTR-RT. By default, the length of the internal region is set to range from 100-15,000bp, while the LTR:internal length ratio is set to [0.05, 50]. The broad range of length settings allows LTR_retriever to identify very short elements like TRIM or exceptionally long elements. The implementation of **Module 1** allows

LTR_retriever to exclude 4~12% of total candidates which are very likely false positives.

Identifying the exact boundaries of an LTR candidate is critical for further structural analysis such as motifs and TSDs. Published methods have applied some schemes to define boundaries. For example, in LTRharvest, the *-vic* parameter and *-motifmis* parameter were deployed to control the motif search range and ambiguousness of boundaries. Similarly, in LTR_finder, the boundary alignment sharpness thresholds (*-b* and *-B*) were applied. In practice, we found that the external boundaries of an LTR candidate were defined quite precisely by prediction methods. However, for the internal boundaries which define the start and end of the internal region, predictions of existing methods are often incorrect. By manual inspections, we found the percentage of inaccurate internal boundary could be as high as 30%. The misdefined internal boundary of an LTR candidate will result in an incorrect prediction of LTR structures, such as motif, PBS, and PPT, which is likely to fail in the next filtering steps. By correcting the internal boundaries of raw LTR predictions using **Module 2**, we were able to recover an extra 27% high-quality LTR candidates in the rice genome.

LTR-RT features with long terminal repeats flanking each side of the internal region. To exhaustively search for LTR candidates from genomic sequences, most published tools start with finding sequence alignments that are close to each other. This approach can effectively identify LTR elements featured with a pair of long terminal repeats as well as finding non-LTR TE pairs that are similar to each other (**Figure 1**). Such non-LTR TE fragments could be contributed by tandem repeats, DNA TEs, SINEs, LINEs, solo-LTRs from the same LTR-RT family, or other repetitive sequences including tandemly located gene families. Excluding such LTR-like false positives is challenging. Moreover, consider that some TEs prefer to insert into other TE sequences, TE clusters are frequently found (51, 52). The dense distribution of TEs creates a significant amount of false LTRs in *de novo* predictions. With close inspection, we found that in most cases, the intra-element sequence similarity of such false positives extended beyond the predicted boundary of the direct repeat (**Figure 1E**). In contrast, for a true LTR-RT, the sequence alignment terminates at the boundary of the LTR region. This represents an important structural feature that could distinguish LTR-RTs and its false positives. Another distinctive feature between true LTR and such false positives is the existence of TSDs. In an LTR-RT, TSDs flanking the element are identical (**Figure 1A**). However, in an LTR false positive, sequences at each end have different origins (**Figure 1E**). For 4-6bp random sequences, the possibility of one being identical to the other is 0.02-0.39%, which is very unlikely. To utilize the structural difference between LTR-RT and false positives for the exclusion of the latter, **Module 3** was developed. Benefiting from the accurate boundaries of candidate elements corrected by **Module 2**, this module could effectively identify most of the false positives which could account for nearly half (42.6%) of total LTR candidates.

Module 3 also allows fine-grained adjustment of the internal and external element boundaries by jointly searching TSDs and motifs. As LTR-RTs are predominantly represented by 5bp TSD and the 5'-TG..CA-3' motif, searching for such sequence structure at the termini of direct repeats is prioritized. If the canonical motif is absent, the seven non-canonical motifs (TGCA, TGCT, TACA, TACT, TGGA, TATA, and TGTA) is searched instead. This function allows LTR_retriever flexibly while accurately characterizing the terminal structure of an LTR candidate. In rice, up to 99% of recognized LTR-RTs carry the canonical 5'-TG..CA-3' motif immediately flanked by 5bp TSDs, while less than 0.1% of LTR-RTs have non-canonical motifs with 5bp TSDs. In other cases, LTR candidates were found carrying the canonical motif with TSDs less than 5bp, which could be due to inter-element recombination or mutation. For example, in the maize genome, LTR-RT with TSD length of 3bp and 4bp have 108 and 483 occurrences out of 43,226 intact LTR-RTs, respectively.

Similar to retroviruses, direct repeats of a newly inserted LTR-RT are identical to each other. Based on the neutral theory (17), **Module 4** was developed for the estimation of insertion time of each intact LTR-RT. In most cases, the structure of LTR-RT, e.g., motif, TSD, and direct repeat, is recognizable for a recent insertion. In the rice genome, more than 99% of intact LTR-RTs are inserted less than 4 million years ago (mya) given the mutation rate of 1.3×10^{-8} mutations per site per year (53) (**Supplementary Figure S2**).

In the internal region of an LTR element, coding sequences like *gag*, *pol*, and *env* are usually found (**Figure 1A**) (29). The probability of finding a true LTR-RT is significantly increased giving the presence of a retrotransposition-specific coding region, which could also help to discriminate LTR-RTs and non-LTRs efficiently. In **Module 5**, we applied the profile hidden Markov model (pHMM) to identify conserved protein domains that occur in LTR-RT candidate sequences. A total of 102 TE-related pHMMs were identified using the rice TE library, with 55 non-LTR profiles and 47 LTR-RT profiles which include 30 *gypsy* profiles, 9 *copia* profiles and 8 profiles with ambiguous LTR-RT family classifications (unknown). In rice, 82.6% of intact LTR-RTs could be classified as either *copia* or *gypsy* using **Module 5**. Furthermore, the direction of LTR-RT could be phased using the profile match information. Eventually, 60.5% of LTR-RTs in rice could be phased to either on the positive strand or negative strand. A BLAST-based search for non-LTR transposase and plant coding proteins in LTR-RT candidates are also implemented in **Module 5** for the further exclusion of non-LTR contaminations. About 1-4% of the candidate sequences were recognized as non-LTR originated and could be further eliminated.

After screening and adjustment of LTR candidates using **Module 1** to **Module 5**, the retained candidates are structurally intact LTR-RTs. Such sequences could be used to specifically identify LTR-only remnants (solo LTRs, truncated elements, and fragments) in the genome. However, since the screening criteria are very stringent, some true LTR-RTs could be excluded. Through manual inspection, we found that some LTR-RT candidates passed all the screening criteria but only have deletions of a few base pairs at the 5' or 3' termini, resulting in the failure in the identification of terminal structures. Such candidates are categorized as truncated LTR-RTs. Further inspection found that truncated LTR-RTs are usually defective in only one LTR region, while the other LTR region and the internal region are typically intact. In such cases, the intact LTR region and the internal region will be retained if there is no highly similar copy in the intact LTR element pool. **Module 6** was designed to retain sequence information from truncated LTR-RTs which contributes about 10% of sensitivity increment of LTR_retriever.

New LTR-RT tends to insert into other LTR-RTs, creating nested insertions. To exclude nested insertions from the LTR exemplars, we developed a function in **Module 6**, which utilizes all newly identified LTR regions to search for homologous sequences in identified internal regions. This search could recognize and removes LTR-RTs that are nested in intact LTR-RTs. Using this method, about 8% of LTR-RT internal regions in rice and 67.7% in maize are identified as nested within other LTR elements. By removing such nested insertions, the library size can be reduced significantly without sacrifice of sensitivity. More importantly, it avoids the misannotation of LTR sequences as internal regions.

Identification of LTR-RTs with non-canonical motifs

LTR-RT features dinucleotide motifs flanking the direct repeat regions (**Figure 1**). The most common motif is the palindromic 5'-TG..CA-3' motif. However, during manual curation of LTR-RTs, we discovered many LTRs with non-TGCA motifs (Ferguson and Jiang, unpublished). These non-canonical motifs can be non-palindromic, for example, *Tos17*, a rice LTR-RT that can be activated by tissue culture, has non-canonical motifs of 5'-TG…GA-3' (54); *AtRE1* in Arabidopsis has 5'-TA…TA-3' motifs (55); and *TARE1*, intensively amplified in the tomato genome, has 5'-TA…CA-3' motifs (56). In addition, three copies of *gypsy*-like elements with 5'-TG..CT-3' motifs were annotated in the soybean genome (57).

In order to identify non-TGCA LTR-RT with high confidence, we developed **Module 7** as an optional add-on to LTR_retriever. LTRharvest enables the “-motif” parameter allowing users to specify the motif to be discovered, which requires prior motif knowledge. When users apply the default setting (no motif specified), the number of LTR-RT candidates can be 2-4 times more than the result with “-motif TGCA”. The significant increase of predicted candidates does not necessarily indicate a large number of non-TGCA LTR recovered. With annotations and further curations, we found 99% of the additional candidates are false positives in the rice genome.

The sacred lotus genome carries many non-canonical LTR elements. We tested the performance of LTR_retriever in identifying such elements using the manually curated non-canonical LTR-RTs from this genome (**Supplementary methods**). Our results showed that LTR_retriever found non-canonical LTR-RTs, with a sensitivity of 74.7% and a precision of 81.6% (FDR=18.4%). The specificity and accuracy were 98.5% and 96.5%, respectively, indicating that the identified non-canonical LTR-RTs are highly accurate. Despite the lower level of sensitivity, LTR_retriever showed similar (or even higher) performance (in terms of precision, specificity and accuracy) in recovering non-canonical LTR-RTs comparing to canonical LTR-RTs.

To characterize non-TGCA LTR-RTs in plant genomes, we searched through 50 published plant genomes. A total of 870 high-confidence non-TGCA LTR-RTs were found from these genomes (**Materials and methods**). Further categorization of non-TGCA LTR-RTs identified seven types of high-confident non-canonical motifs including three (TACT, TGTA, and TCCA) that were not previously reported (**Table 1**). Further classification of ORFs within these elements based on pHMM search indicated that among the classified non-TGCA LTR elements, 89% were the *copia* type, while only 11% were the *gypsy* type (**Table 1**). We also identified 83,368 canonical LTR-RTs in these genomes, with a *gypsy* - *copia* ratio of 2.9:1 (**Table 2**).

For canonical LTR-RTs, the length of the LTR region in *gypsy* elements is about 40% longer than *copia* elements (**Table 2**). However, in the case of non-canonical LTR-RTs, this size difference is intensified to 400%. This is due to the significant reduction of LTR length of non-canonical *copia* elements, from an average size of 911bp to 272bp (**Table 2**). The internal region length and whole element length of non-canonical *copia* are also much shorter than those of *copia* elements carrying the TGCA motif (**Table 2**). These results suggest that shorter LTRs may have facilitated the amplification and survival of non-TGCA LTR-RTs.

Construction of non-redundant LTR library

Construction of the repeat library that collects high-quality TE exemplars is critical for RepeatMasker-based TE annotations, with the size of the repeat library being one of the limiting factors for speed. The required time for whole genome TE annotations using RepeatMasker is highly correlated to the size of TE libraries. Since the identified LTR-RTs are redundant, it would significantly speed up whole genome LTR-RT annotation if the redundancy is eliminated. To reduce redundancy of identified LTR-RTs, **Module 8** was developed using the clustering function of BLAST or CD-HIT. Due to the reduced redundancy and exclusion of nested insertions (**Module 6**), the LTR-RT sequence size was reduced to 10-30% of its original size. Accordingly, whole genome LTR-RT annotation could be accelerated ~4-fold with similar sensitivity comparing to a non-redundant LTR library.

Comparison of performances to other LTR identification tools

To compare the performance between LTR_retriever and other existing methods, we employed the rice genome as a reference. The rice genome is one of the best sequenced and assembled genomes (33). To set a standard for our comparison study, we manually curated all LTR candidates obtained from the rice genome (cv. Nipponbare) and generated a compact repeat library which contains 897 sequences with the size of 2.34 Mb. The 897 sequences represent 508 non-redundant LTR elements (**Supplementary Methods**). Using this library, LTR-RT contributes 23.5% of the assembled genome (374 Mb). This number is slightly higher than the two highest estimates from previous studies (20.6%, 22%) (58, 59), suggesting the current identification of LTR retrotransposon in Nipponbare is close to saturation and the library is reasonably comprehensive. As a result, this library is used as a reference library for subsequent analysis. The accurate annotation of LTRs in the rice genome allows us to summarize the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) of a *de novo* LTR prediction and annotation, hence allowing the evaluation of different methods.

The sensitivity of all existing LTR discovery tools is very high (28, 29, 60), however, systematic evaluation of specificity using the whole genome sequence length is not available. Specificity describes the proportion of true negative, i.e., non-LTR sequences, being correctly ruled out, which is as important as sensitivity for evaluation of a diagnostic test (61). To better describe the performance of these methods, precision and accuracy are also calculated (62). Precision, or positive predictive value, is the proportion of true positives, i.e., LTR sequences, among all positive results revealed by the test. The precision is an indication of false discovery rate (FDR), with the equation FDR=1-precision. Accuracy is the proportion of true predictions, which controls systemic errors and random errors (**Materials and Methods**).

For comparison, we chose four of the most widely used LTR searching methods, LTR_STRUC (27), MGEScan-LTR (31), LTR_finder (28), and LTRharvest (29), for performance benchmarks. As LTRharvest is the most flexible program with more than 20 modifiable parameters, we optimized the parameters based on our experience for more accurate predictions (**Figure 3**). The optimized parameters were also applied to the parameter settings of LTR_finder and MGEScan-LTR. LTR_retriever can utilize multiple input sources including the results from LTR_finder, LTRharvest, and MGEScan-LTR. We used separate and combined inputs in LTR_retriever for comparisons.

As expected, sensitivities of the most published methods are very high, ranging from 91.2% to 95.3% (**Figure 3, Supplementary Table S1**). However, specificities of these methods are not desirable, ranging from 72.3% to 87.7% (**Figure 3, Supplementary Table S1**) with the exception of LTR-finder (91.0%). Specificity of 72.3% indicates that 27.7% of non-LTR genomic sequences were falsely recognized as LTR-RT sequences. The optimized parameters in LTRharvest led to an improvement of the specificity from 79.2% to 87.7% (**Supplementary Table S1**). The optimized LTR_finder had the best balance, with sensitivity and specificity both reached to the level of 90%, however, its precision is only 75.8% (**Figure 3, Supplementary Table S1**). As a reminder, FDR=1-precision. Although LTR_finder has the highest precision among the published methods, the precision of 75.8% indicates that 24.2% of "LTR-RT related sequences" identified in the genome were falsely reported as LTR-RT. The accuracy of existing methods ranges from 77.5-91.3%, showing variations in true prediction rate.

We tested LTR_retriever using the optimized LTRharvest results as input. As a stringent filter, LTR_retriever achieved specificity and accuracy of 96.8% and 95.5%, respectively, greatly outperforming existing methods (**Figure 3, Supplementary Table S1**). The precision also increased from the original 69.9% to 89.9%, indicating the FDR dropped to 1/3 and is among the lowest of all methods (**Figure 3, Supplementary Table S1**). Strikingly, the sensitivity of LTR_retriever remained as high as 91.1% compared to the original 93.0%, meaning that we only sacrificed less than 2% of sensitivity to achieve the observed performance improvements (**Figure 3, Supplementary Table S1**). Other input sources such as those from LTR_finder and MGEScan-LTR were also tested and showed excellent performance (**Supplementary Table S1**). Upon combination of two or more input sources, the sensitivity is increased to 94.5%, which is equivalent to the highest level that was achieved by the existing methods, providing a workaround to achieve comprehensive and high-quality predictions (**Supplementary Table S1**). By excluding the majority of false positives, the final library size was substantially reduced, from the largest 44.4 MB by MGEScan-LTR to the final 4.4 MB by the

LTR_retriever (**Supplementary Table S1**). The reduced library size significantly reduced the annotation time using RepeatMasker.

Benchmarking on other genomes

LTR_retriever was developed based on the rice genome, which has demonstrated the highest specificity, accuracy, and precision among its counterparts with the same level of sensitivity. To test whether the excellent performance of LTR_retriever can be reproduced with other genomes, we chose four other genomes with variable amounts of LTR elements including two maize genomes (cv. B73 and cv. Mo17) (8, 63), Arabidopsis (64), and sacred lotus (14). All these genomic sequences are associated with reasonable repeat libraries so that performance of LTR_retriever could be evaluated by comparisons between the respective standard annotations and LTR_retriever generated libraries.

For all the genomes we tested, LTR_retriever demonstrated very sensitive and accurate performance in retrieving LTRs. Most metrics reached the levels of 90% (**Table 3**). For Arabidopsis, we obtained a very high specificity and accuracy, which were 98.9% and 98.4%, respectively, indicating the nearly perfect prediction by LTR_retriever. For the ancient eudicot sacred lotus, the four metrics ranged from 81.2% to 91.3%. The maize genome is known to be highly repetitive, and we used both the reference B73 (v4) and the Mo17 genomes to evaluate the performance of LTR_retriever. With LTR-RTs comprising ~75% of the 2.1 GB genome, LTR_retriever could identify 91.1% and 95.7% LTR-RTs with specificities of 90.6% and 95.7%, respectively. Due to the high LTR-RT content and the nearly perfect performance of LTR_retriever, the precisions reached 96.6% (FDR=3.4%) and 98.7% (FDR=1.3%), respectively. It is known that structure of the maize genome is very complex due to intensive nested TE insertions (65), LTR_retriever is able to overcome complex structures and recover most LTR-RTs from the genome.

Direct LTR library construction from PacBio reads

The recent development of long-read sequencing technologies has provided a solution for resolving highly repetitive regions in *de novo* genome sequencing projects (66). The PacBio single molecule, real-time (SMRT) sequencing technology produces long reads with an average length of 10-15kb. Empirically, more than 95% of LTR-RTs range from 1-15kb (**Supplementary Figure S1**). Thus, theoretically, the long-read sequencing technology may allow us to identify intact LTR elements directly from the reads.

It is known that the current PacBio RS II platform has an average sequencing error rate of 15%. In our experience, most LTR-RT insertions are structurally detectable if inserted 4 million years ago or younger (**Supplementary Figure S2**) which is equivalent to 89.6% of identity between two LTR regions. When mutations/sequencing errors accumulated, the fine structure such as TSD and terminal motifs could be mutated and element would be beyond the detection limit. Thus the sequencing error rate of 15% could have artificially aged the actual LTR element to become undetectable. We tested the LTR_retriever using raw PacBio reads and no confident intact LTR element was reported. However, LTR_retriever performed excellently using self-corrected PacBio reads with an error rate of 2%.

To test the efficiency of LTR_retriever, we used 20 thousands (k) self-corrected PacBio reads from Arabidopsis L*er*-0 as an initial input (**Materials and Methods**), and with 20 k reads as an increment until 180 k. The Arabidopsis repeat library from Repbase was used to calculate sensitivity, specificity, accuracy, and precision. The LTR library constructed from the Arabidopsis L*er*-0 genome was used as the control to compare to the quality of LTR libraries constructed from PacBio reads. As more reads were used, the prediction of intact LTR-RTs increased linearly (**Figure 4A**). However, the size of LTR libraries constructed from these candidates are not increased at the same rate (**Figure 4A**), and the sensitivity exceeds the library developed from the genome sequence after 40 k reads input and being saturated at 93% after 120 k reads being used (**Figure 4B**). Since the average length of these reads is 14.6kb, and the Arabidopsis "L*er*-0" genome was assembled as ~131 MB, the sample of 40 k and 200 k reads is equivalent to 4.5- and 13.4-fold genome coverage, respectively. Moreover, despite the amount of reads being used, the average specificity, accuracy, and precision were 99.5%, 98.8%, and 94.0%, respectively, indicating very high-quality LTR libraries could be constructed from PacBio reads. Furthermore, masking potentials (percentage of the genome that could be masked) of PacBio LTR libraries surpass the standard library level after using 40 k or more reads (**Supplementary Figure S3**), indicating that it is sufficient to construct a comprehensive library using as little as 4.5X PacBio self-corrected reads. To summarize, LTR_retriever shows high sensitivity, specificity, accuracy, and precision to construct LTR libraries directly from self-corrected PacBio reads prior to genome assembly.

DISCUSSION

Technological advances have minimized the cost of sequencing a genome. The real bottleneck to establishing genomic resources of an organism is the annotation of its genomic sequence. As mentioned above, TEs, particularly LTR retrotransposons, are the largest component of most plant genomes. If TEs are left unmasked prior to gene annotation, they would seed numerous of spurious sequence alignments, producing false evidence for gene identification. Even worse, the open reading frames of TEs look like *bonafide* genes to most gene-prediction software, corrupting the final annotations. As a result, the first step of genome annotation is to identify TEs and other repeats. Subsequently, these repeats are masked to facilitate gene annotation. As a result, the quality of repeat library is not only important for the study of repeats, but also critical for high-quality gene prediction.

In this study, we reported the development of LTR_retriever, a multithreading empowered Perl program that can process LTR-RT candidates from LTR_finder, LTRharvest, and MGEScan-LTR and generate high-quality and compact LTR libraries for genome annotations or study of transposable elements. We curated LTR elements identified from the rice genome and used the curated LTR library as the standard to test the performance of LTR_retriever in terms of sensitivity, specificity, accuracy, and precision. Benchmark tests on existing programs indicated very high sensitivities achieved, however, specificities and accuracies were not satisfactory, and the FDR could be as high as 49%, suggesting the necessity for improvement (**Supplementary Table S1**).

Since annotation of TE sequences usually precedes the annotation of functional genes for a newly sequenced genome, propagation of false positives in the construction of LTR library will significantly increase misidentification of LTR sequences in the genome and further dampen the power of downstream annotations. For example, it is known that most DNA transposons target genic regions and avoid repetitive sequences (67, 68). As a result, it is not uncommon that the sequence between two adjacent DNA transposons represents gene coding regions or regulatory sequences. If the two DNA transposons are mistakenly annotated as the LTR of an individual LTR-RT, the intervening genes would be considered as the internal region of an LTR-RT and would be masked before gene annotation. In this scenario, the false positives could be extremely detrimental for downstream analyses. LTR_retriever effectively eliminates such false positives. By processing LTR-RT candidates using LTR_retriever, the specificity and accuracy reached to 96.9% and 95.7%, respectively, and the FDR is reduced to 10% which is among the lowest of all existing methods (**Figure 3, Supplementary Table S1**). Strikingly, the sensitivity of LTR_retriever remained as high as 91.7%, meaning that we only sacrificed less than 2% of sensitivity to achieve all these performance improvements (**Figure 3, Supplementary Table S1**). Further benchmark tests on two maize genomes, the sacred lotus genome, and the Arabidopsis genome also showed excellent performance (**Table 3**), suggesting that LTR_retriever is compatible with both monocot and dicot genomes.

The majority of LTR-RTs we identified carried a palindromic dinucleotide motif flanking each direct repeat. The motif is well conserved and is usually 5'-TG..CA-3'. However, the importance of such conservation is poorly understood. Retrovirus, e.g., HIV-1, is thought to be the close relative of LTR elements with the addition of an envelope protein (69, 70). Studies of retrovirus integration indicated that the terminal sequences of retroviral LTR regions, especially the 3' CA ends, are essential and important for integration of the virus (69, 70). That may explain why most LTR elements have the conserved TG..CA motif.

Despite the conservation, non-TGCA motifs were also found but in a much lower frequency. LTR_retriever also demonstrated high performance in identifying such non-canonical LTR-RTs. A broad scan on 50 published plant genomes retrieved 7 non-TGCA type LTR-RTs with the majority belonging to the *copia* family (**Table 1**). For some, the abundance is not ignorable. It appears that, among the four terminal nucleotides (TGCA), only the first nucleotide is invariable. We noticed that the sensitivity of LTR_retriever to search non-canonical LTR-RTs was lower (74.7%) than that of the modules for canonical LTR-RT searching (89.4%) (**Table 3**). One of the main reasons is that we applied very stringent screening criteria to ensure the genuineness of non-canonical terminal motif. Hence, it is possible that some non-canonical LTR-RTs with slightly ambiguous terminal structures were excluded, which leads to the decrease of sensitivity. Future studies may focus on improving the sensitivity of identifying non-canonical LTR-RTs.

The recent development of single molecule sequencing technology enables the assembly of low complexity and repetitive regions. Many genome sequencing projects have benefited from the PacBio SMRT sequencing technique which features with 10-15kb average read length (11, 66). Given the length of most LTR elements is less than 15kb (**Supplementary Figure S1**), it is possible to identify full-length LTRs from PacBio long reads. We applied LTR_retriever on self-corrected PacBio reads which proved a successful strategy to identify LTR-RTs. For the Arabidopsis "L*er*-0" genome, 40 thousand self-corrected reads covering approximately 4.5X of the genome were more than sufficient to generate an LTR library with higher quality compared to that generated from the assembled genome (**Figure 4**). Although self-corrected reads still have ~2% sequencing error rate, the generated LTR library was proven highly sensitive and accurate (**Figure 4**). The pre-identified full-length LTRs may help to estimate LTR percentages of the new genome, study the evolution of LTR-RTs without performing the computationally intensive whole genome assembly, and facilitate downstream *de novo* gene annotation. Since LTR-RTs contribute greatly to the size of plant genomes, identification and removal of repetitive sequences in advance could speed up the genome assembly by as much as 50-fold (Gregory Concepcion, Pacific Bioscience, personal communication).

In summary, we developed a package which takes genome sequences or corrected PacBio reads as input and generates high-quality, non-redundant libraries for LTR elements. It also provides information about the insertion time and location of intact LTR elements in the genome. This tool demonstrates significant improvements in specificity, accuracy, and precision while maintaining the high sensitivity compared to existing methods. As a result, it will facilitate future genome assembly and annotation as well as enable rapid comparative studies of LTR-RT dynamics in multiple genomes.

AVAILABILITY

LTR_retriever is an open source software available in the GitHub repository (https://github.com/oushujun/LTR_retriever).

FUNDING

This work was supported by the National Science Foundation [MCB-1121650 to N.J., IOS-1126998 to N.J.]; and the United States Department of Agriculture National Institute of Food and Agriculture and AgBioResearch at Michigan State University (Hatch grant MICL02120 to N.J.).

ACKNOWLEDGMENTS

We thank Dr. Yi Liao (Institute of Genetics and Developmental Biology, Chinese Academy of Sciences) for valuable discussions. We thank Stefan Cerbin, Drs. Cornelius Barry, Rebecca Grumet, Steve van Nocker, and Wayne Loescher for critical reading of the manuscript.

REFERENCE

1. Wessler SR. Transposable elements and the evolution of eukaryotic genomes. Proceedings of the National Academy of Sciences of the United States of America. 2006;103(47):17600-1.
2. Jiang N. Plant Transposable Elements. eLS: John Wiley & Sons, Ltd; 2016.
3. Zhao D, Ferguson AA, Jiang N. What makes up plant genomes: The vanishing line between transposable elements and genes. Biochimica et biophysica acta. 2015;1859(2):366-80.
4. Wicker T, Sabot F, Hua-Van A, Bennetzen JL, Capy P, Chalhoub B, et al. A unified

classification system for eukaryotic transposable elements. Nature reviews Genetics. 2007;8(12):973-82.

5. Kumar A, Bennetzen JL. Plant retrotransposons. Annual review of genetics. 1999;33:479-532.

6. Rensing SA, Lang D, Zimmer AD, Terry A, Salamov A, Shapiro H, et al. The *Physcomitrella* genome reveals evolutionary insights into the conquest of land by plants. Science. 2008;319(5859):64-9.

7. Nystedt B, Street NR, Wetterbom A, Zuccolo A, Lin Y-C, Scofield DG, et al. The Norway spruce genome sequence and conifer genome evolution. Nature. 2013;497(7451):579-84.

8. Schnable PS, Ware D, Fulton RS, Stein JC, Wei F, Pasternak S, et al. The B73 maize genome: complexity, diversity, and dynamics. Science. 2009;326(5956):1112-5.

9. Jiao Y, Peluso P, Shi J, Liang T, Stitzer MC, Wang B, et al. The complex sequence landscape of maize revealed by single molecule technologies. bioRxiv. 2016.

10. The Tomato Genome Consortium. The tomato genome sequence provides insights into fleshy fruit evolution. Nature. 2012;485(7400):635-41.

11. Ming R, VanBuren R, Wai CM, Tang H, Schatz MC, Bowers JE, et al. The pineapple genome and the evolution of CAM photosynthesis. Nature genetics. 2015;47(12):1435-42.

12. Piegu B, Guyot R, Picault N, Roulin A, Saniyal A, Kim H, et al. Doubling genome size without polyploidization: Dynamics of retrotransposition-driven genomic expansions in *Oryza australiensis*, a wild relative of rice. Genome research. 2006;16(10):1262-9.

13. Amborella Genome Project. The *Amborella* genome and the evolution of flowering plants. Science. 2013;342(6165):1241089.

14. Ming R, VanBuren R, Liu Y, Yang M, Han Y, Li L-T, et al. Genome of the long-living sacred lotus (*Nelumbo nucifera* Gaertn.). Genome biology. 2013;14(5):R41.

15. Ammiraju JS, Zuccolo A, Yu Y, Song X, Piegu B, Chevalier F, et al. Evolutionary dynamics of an ancient retrotransposon family provides insights into evolution of genome size in the genus *Oryza*. The Plant journal : for cell and molecular biology. 2007;52(2):342-51.

16. Ammiraju JSS, Fan C, Yu Y, Song X, Cranston KA, Pontaroli AC, et al. Spatio-temporal patterns of genome evolution in allotetraploid species of the genus *Oryza*. The Plant Journal. 2010;63(3):430-42.

17. vonHoldt BM, Takuno S, Gaut BS. Recent retrotransposon insertions are methylated and phylogenetically clustered in *japonica* rice (*Oryza sativa* ssp. *japonica*). Molecular biology and evolution. 2012;29(10):3193-203.

18. Elbarbary RA, Lucas BA, Maquat LE. Retrotransposons as regulators of gene expression. Science. 2016;351(6274).

19. Makarevitch I, Waters AJ, West PT, Stitzer MC, Hirsch CN, Ross-Ibarra J, et al. Transposable elements contribute to activation of maize genes in response to abiotic stress. PLoS genetics. 2015;11(1):e1004915.

20. Havecker ER, Gao X, Voytas DF. The diversity of LTR retrotransposons. Genome biology. 2004;5(6):225.

21. Gao D, Chen J, Chen M, Meyers BC, Jackson S. A highly conserved, small LTR retrotransposon that preferentially targets genes in grass genomes. PloS one. 2012;7(2):e32010.

22. Kalendar R, Vicient CM, Peleg O, Anamthawat-Jonsson K, Bolshoy A, Schulman AH. Large retrotransposon derivatives: abundant, conserved but nonautonomous retroelements of barley and related genomes. Genetics. 2004;166(3):1437-50.

23. Fedoroff NV. Transposable elements, epigenetics, and genome evolution. Science. 2012;338(6108):758-67.

24. Tian Z, Rizzon C, Du J, Zhu L, Bennetzen JL, Jackson SA, et al. Do genetic recombination and gene density shape the pattern of DNA elimination in rice long terminal repeat retrotransposons? Genome research. 2009;19(12):2221-30.

25. Levy A, Schwartz S, Ast G. Large-scale discovery of insertion hotspots and preferential integration sites of human transposed elements. Nucleic acids research. 2010;38(5):1515-30.

26. Hoen DR, Hickey G, Bourque G, Casacuberta J, Cordaux R, Feschotte C, et al. A call for benchmarking transposable element annotation methods. Mobile DNA. 2015;6:13.

27. McCarthy EM, McDonald JF. LTR_STRUC: a novel search and identification program for LTR retrotransposons. Bioinformatics. 2003;19(3):362-7.

28. Xu Z, Wang H. LTR_FINDER: an efficient tool for the prediction of full-length LTR retrotransposons. Nucleic acids research. 2007;35(Web Server issue):W265-8.

29. Ellinghaus D, Kurtz S, Willhoeft U. LTRharvest, an efficient and flexible software for *de novo* detection of LTR retrotransposons. BMC bioinformatics. 2008;9(1):18.

30. Lerat E. Identifying repeats and transposable elements in sequenced genomes: how to find your way through the dense forest of programs. Heredity. 2010;104(6):520-33.

31. Rho M, Choi J-H, Kim S, Lynch M, Tang H. *De novo* identification of LTR retrotransposons in eukaryotic genomes. BMC genomics. 2007;8(1):90.

32. Lee H, Lee M, Mohammed Ismail W, Rho M, Fox GC, Oh S, et al. MGEScan: a Galaxy-based system for identifying retrotransposons in genomes. Bioinformatics. 2016.

33. Goff SA, Ricke D, Lan TH, Presting G, Wang R, Dunn M, et al. A draft sequence of the rice genome (*Oryza sativa* L. ssp. *japonica*). Science. 2002;296(5565):92-100.

34. Veeckman E, Ruttink T, Vandepoele K. Are we there yet? Reliably estimating the completeness of plant genome sequences. The Plant cell. 2016.

35. Benson G. Tandem repeats finder: a program to analyze DNA sequences. Nucleic acids research. 1999;27(2):573-80.

36. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, et al. BLAST+: architecture and applications. BMC bioinformatics. 2009;10:421-.

37. Bowen NJ, McDonald JF. *Drosophila* euchromatic LTR retrotransposons are much younger than the host species in which they reside. Genome research. 2001;11(9):1527-40.

38. Jukes TH, Cantor CR. Evolution of Protein Molecules. In: MUNRO HN, editor. Mammalian Protein Metabolism: Academic Press; 1969. p. 21-132.

39. Ma J, Bennetzen JL. Rapid recent growth and divergence of rice nuclear genomes. Proceedings of the National Academy of Sciences of the United States of America. 2004;101(34):12404-10.

40. Eddy SR. Accelerated profile HMM searches. PLoS computational biology. 2011;7(10):e1002195.

41. Finn RD, Coggill P, Eberhardt RY, Eddy SR, Mistry J, Mitchell AL, et al. The Pfam protein families database: towards a more sustainable future. Nucleic acids research. 2016;44(D1):D279-D85.

42. Holt C, Yandell M. MAKER2: an annotation pipeline and genome-database management tool for second-generation genome projects. BMC bioinformatics. 2011;12(1):1-14.

43. The National Center for Biotechnology Information (NCBI). BLASTCLUST - BLAST score-based single-linkage clustering 2000 [Available from:

ftp://ftp.ncbi.nih.gov/blast/documents/blastclust.html.

44. Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics. 2006;22(13):1658-9.

45. Kawahara Y, de la Bastide M, Hamilton J, Kanamori H, McCombie W, Ouyang S, et al. Improvement of the *Oryza sativa* Nipponbare reference genome using next generation sequence and optical map data. Rice. 2013;6(1):1-10.

46. Berardini TZ, Reiser L, Li D, Mezheritsky Y, Muller R, Strait E, et al. The Arabidopsis information resource: Making and mining the "gold standard" annotated reference plant genome. genesis. 2015;53(8):474-85.

47. Goodstein DM, Shu S, Howson R, Neupane R, Hayes RD, Fazo J, et al. Phytozome: a comparative platform for green plant genomics. Nucleic acids research. 2012;40(Database issue):D1178-D86.

48. Jurka J. Repbase update: a database and an electronic journal of repetitive elements. Trends in Genetics. 2000;16(9):418-20.

49. Gremme G, Steinbiss S, Kurtz S. GenomeTools: A comprehensive software library for efficient processing of structured genome annotations. IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM. 2013;10(3):645-56.

50. Estill JC, Bennetzen JL. The DAWGPAWS pipeline for the annotation of genes and transposable elements in plant genomes. Plant Methods. 2009;5(1):1-11.

51. Bergman C, Quesneville H, Anxolabehere D, Ashburner M. Recurrent insertion and duplication generate networks of transposable element sequences in the *Drosophila melanogaster* genome. Genome biology. 2006;7(11):R112.

52. SanMiguel P, Gaut BS, Tikhonov A, Nakajima Y, Bennetzen JL. The paleontology of intergene retrotransposons of maize. Nature genetics. 1998;20(1):43-5.

53. Ma J, Bennetzen JL. Rapid recent growth and divergence of rice nuclear genomes. Proceedings of the National Academy of Sciences of the United States of America. 2004;101(34):12404-10.

54. Hirochika H, Sugimoto K, Otsuki Y, Tsugawa H, Kanda M. Retrotransposons of rice involved in mutations induced by tissue culture. Proceedings of the National Academy of Sciences of the United States of America. 1996;93(15):7783-8.

55. Kuwahara A, Kato A, Komeda Y. Isolation and characterization of *copia*-type retrotransposons in *Arabidopsis thaliana*. Gene. 2000;244(1-2):127-36.

56. Yin H, Liu J, Xu Y, Liu X, Zhang S, Ma J, et al. *TARE1*, a mutated *Copia*-like LTR retrotransposon followed by recent massive amplification in tomato. PloS one. 2013;8(7):e68587.

57. Du J, Tian Z, Bowen NJ, Schmutz J, Shoemaker RC, Ma J. Bifurcation and enhancement of autonomous-nonautonomous retrotransposon partnership through LTR swapping in soybean. The Plant cell. 2010;22(1):48-61.

58. Ma J, Devos KM, Bennetzen JL. Analyses of LTR-retrotransposon structures reveal recent and rapid genomic DNA loss in rice. Genome research. 2004;14(5):860-9.

59. Chaparro C, Guyot R, Zuccolo A, Piégu B, Panaud O. RetrOryza: a database of the rice LTR-retrotransposons. Nucleic acids research. 2007;35(Database issue):D66-D70.

60. You FM, Cloutier S, Shan Y, Ragupathy R. LTR Annotator: Automated identification and annotation of LTR retrotransposons in plant genomes. International Journal of Bioscience, Biochemistry and Bioinformatics. 2015;5(3):165-74.

61. Zhu W, Nancy Z, Ning W, editors. Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS® implementations. NorthEast SAS Users Group; 2010; Baltimore, Maryland.

62. Fawcett T. An introduction to ROC analysis. Pattern Recognition Letters. 2006;27(8):861-74.

63. Xin M, Yang R, Li G, Chen H, Laurie J, Ma C, et al. Dynamic expression of imprinted genes associates with maternally controlled nutrient allocation during maize endosperm development. The Plant cell. 2013;25(9):3212-27.

64. Arabidopsis Genome Initiative. Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. Nature. 2000;408(6814):796-815.

65. SanMiguel P, Tikhonov A, Jin Y-K, Motchoulskaia N, et al. Nested retrotransposons in the intergenic regions of the maize genome. Science. 1996;274(5288):765-8.

66. VanBuren R, Bryant D, Edger PP, Tang H, Burgess D, Challabathula D, et al. Single-molecule sequencing of the desiccation-tolerant grass *Oropetium thomaeum*. Nature. 2015;advance online publication.

67. Han Y, Qin S, Wessler SR. Comparison of class 2 transposable elements at superfamily resolution reveals conserved and distinct features in cereal grass genomes. BMC genomics. 2013;14(1):1-10.

68. Feschotte C, Pritham EJ. DNA transposons and the evolution of eukaryotic genomes. Annual review of genetics. 2007;41:331-68.

69. Hobaika Z, Zargarian L, Boulard Y, Maroun RG, Mauffret O, Fermandjian S. Specificity of LTR DNA recognition by a peptide mimicking the HIV-1 integrase α4 helix. Nucleic acids research. 2009;37(22):7691-700.

70. Zhou H, Rainey GJ, Wong S-K, Coffin JM. Substrate sequence selection by retroviral integrase. Journal of virology. 2001;75(3):1359-70.

TABLE AND FIGURE LEGENDS

Table 1. LTR-RTs with non-canonical motifs from 50 sequenced plant genomes.

Table 2. Element size of different types of LTR-RTs in 50 sequenced plant genomes.

Table 3. Performances of LTR_retriever on model plant genomes. Methods and program settings refer to that described in Figure 3. *Redundancy of the Arabidopsis library is not reduced since it is already very compact.

Figure 1. The structure of LTR retrotransposons (LTR-RT), their derivatives, and false positives. (**A**) The structure of an intact LTR-RT with long terminal repeat (LTR) (navy pentagons), a pair of di-nucleotide palindromic motifs flanking each LTR (magenta triangles), the internal region including protein coding sequences for *gag*, *pol*, and *env* (green boxes), and 5bp target site duplication (TSD) flanking the element (gray boxes). (**B**) A truncated LTR-RT with missing structural components. (**C**) A solo-LTR. (**D**) A nested LTR-RT with another LTR-RT inserted into its coding region. (**E**) A false LTR-RT detected due to two adjacent non-LTR repeats (gray boxes). The counterfeit also features with a direct repeat (blue pentagons) but usually has extended

sequence similarity on one or both sides of the LTR (orange and brown boxes). Regions a-d are extracted and analyzed by LTR_retriever.

Figure 2. Workflow of LTR_retriever. Modules 1-8 are indicated in parentheses.

Figure 3. Comparison of the performance of LTR-RT recovery programs on the rice genome. LTR libraries of the rice genome were constructed using LTR_STRUC, MGEScan-LTR, LTR_finder, LTRharvest, and LTR_retriever, respectively, and then were used to identify LTR sequences in the genome using RepeatMasker. Identified candidate sequences were compared to whole-genome LTR sequences recognized by the manually curated standard library (**Supplementary Methods**). The genomic size (bp) of true positive, false positive, true negative, and false negative were used to calculate sensitivity, specificity, accuracy, and precision. *Indicates the analysis were using optimized parameters (**Materials and Methods**) while the remainder was in default parameters.

Figure 4. Direct library construction using self-corrected PacBio reads. (**A**) Identification of intact LTR elements and construction of libraries using the Arabidopsis "L*er*-0" genome and 20k - 180k self-corrected PacBio reads. (**B**) The performance of custom LTR libraries compared with that from the Arabidopsis reference (Col-0) genome.

Supplementary Table S1. Performances of LTR-RT recovery programs on the rice genome. ^{Z}Input source for LTR_retriever is indicated in parentheses obtained using optimized parameters unless notified. Har_dft, LTRharvest with default parameters; Har, LTRharvest; MGE, MGEScan-LTR; Fin, LTR_finder; NMTF, LTRharvest with the default "-motif" parameter. ^{Y}Jiang_rice6.9.lib_LTR, the manually curated standard library. ^{X}Programs were run using default parameters. ^{W}Programs were applying optimized parameters except noted.

Supplementary Figure S1. Size distributions of (**A**) full-length LTR elements, (**B**) internal regions, and (**C**) LTR regions in the rice genome.

Supplementary Figure S2. Insertion time distributions of intact LTRs in the rice genome. Mutation rate $\mu=1.3*10^{-8}$ per base pair per year.

Supplementary Figure S3. Masking efficiency of LTR libraries derived from PacBio reads of the Arabidopsis "L*er*-0" genome. The standard LTR library from the reference genome (Col-0) was used to mask the assembled "L*er*-0" genome. Custom libraries generated from the Arabidopsis "L*er*-0" genome and 20k - 180k self-corrected PacBio reads were used to mask the Arabidopsis "L*er*-0" genome.

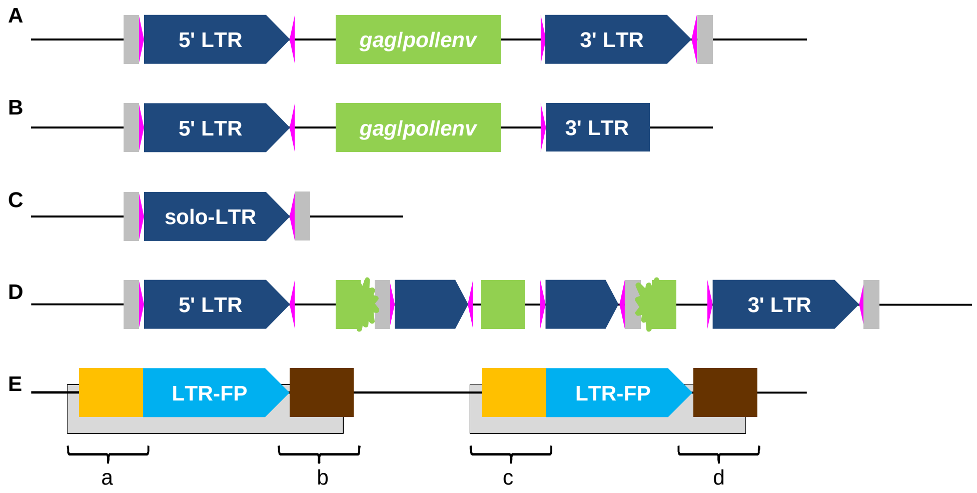


Figure 1. The structure of LTR retrotransposons (LTR-RT), their derivatives, and false positives. (**A**) The structure of an intact LTR-RT with long terminal repeat (LTR) (navy pentagons), a pair of di-nucleotide palindromic motifs flanking each LTR (magenta triangles), the internal region including protein coding sequences for *gag*, *pol*, and *env* (green boxes), and 5bp target site duplication (TSD) flanking the element (gray boxes). (**B**) A truncated LTR-RT with missing structural components. (**C**) A solo-LTR. (**D**) A nested LTR-RT with another LTR-RT inserted into its coding region. (**E**) A false LTR-RT detected due to two adjacent non-LTR repeats (gray boxes). The counterfeit also features with a direct repeat (blue pentagons) but usually has extended sequence similarity on one or both sides of the LTR (orange and brown boxes). Regions a-d are extracted and analyzed by LTR_retriever.

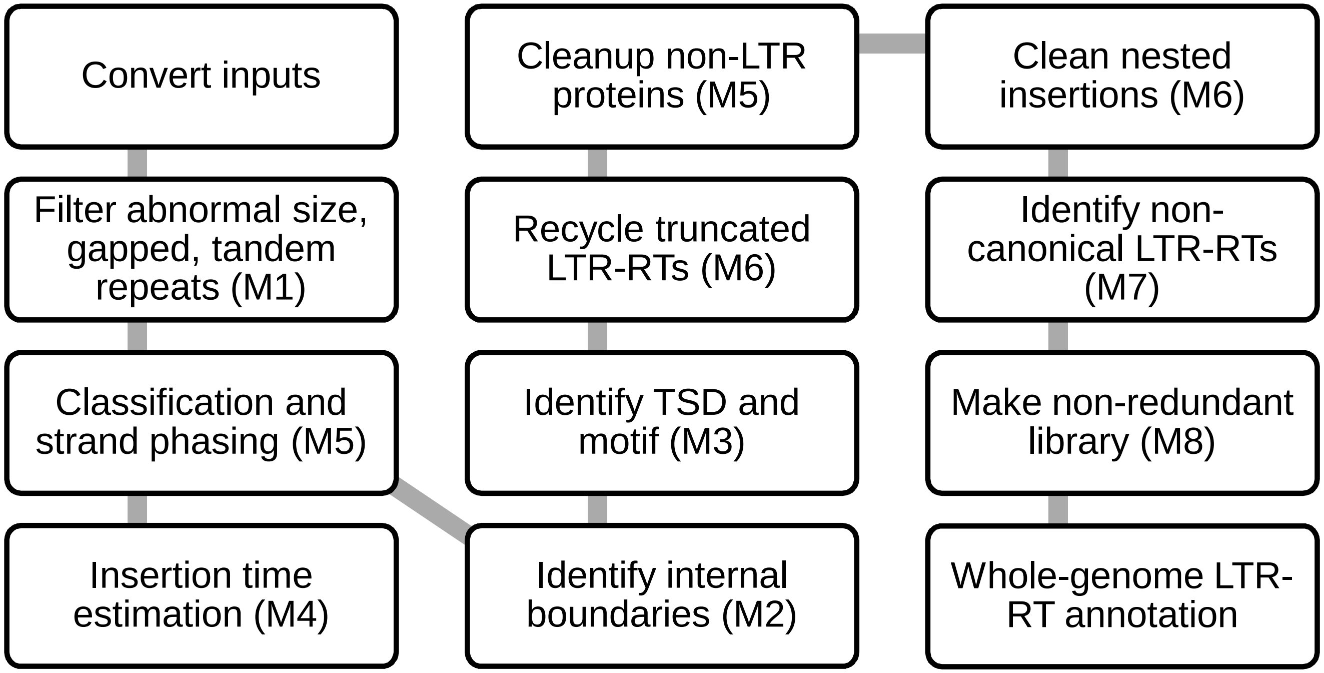

Convert inputs
Filter abnormal size, gapped, tandem repeats (M1)
Classification and strand phasing (M5)
Insertion time estimation (M4)
Cleanup non-LTR proteins (M5)
Recycle truncated LTR-RTs (M6)
Identify TSD and motif (M3)
Identify internal boundaries (M2)
Clean nested insertions (M6)
Identify non-canonical LTR-RTs (M7)
Make non-redundant library (M8)
Whole-genome LTR-RT annotation

Figure 2. Workflow of LTR_retriever. Modules 1-8 are indicated in parentheses.

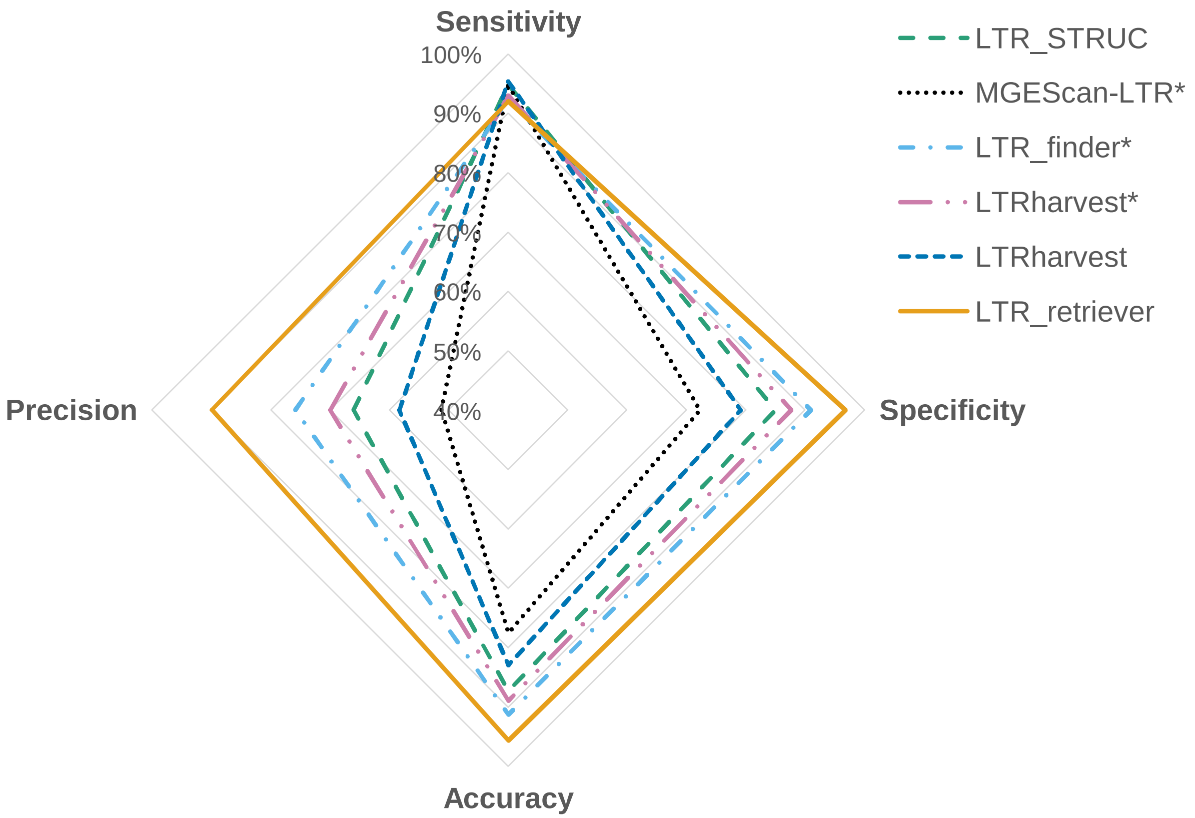


Figure 3. Comparison of the performance of LTR-RT recovery programs on the rice genome. LTR libraries of the rice genome were constructed using LTR_STRUC, MGEScan-LTR, LTR_finder, LTRharvest, and LTR_retriever, respectively, and then were used to identify LTR sequences in the genome using RepeatMasker. Identified candidate sequences were compared to whole-genome LTR sequences recognized by the manually curated standard library (**Supplementary Methods**). The genomic size (bp) of true positive, false positive, true negative, and false negative were used to calculate sensitivity, specificity, accuracy, and precision (**Materials and Methods**). *Indicates the analysis were using optimized parameters (**Materials and Methods**) while the remainder was in default parameters.

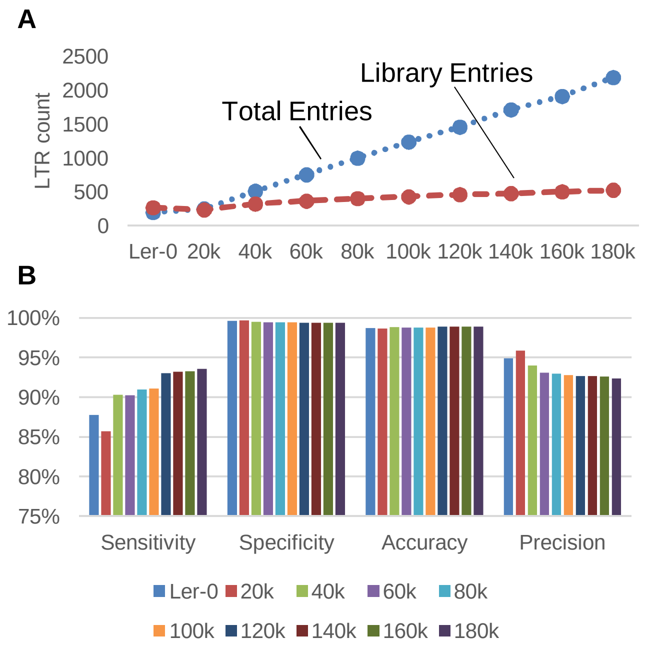


Figure 4. Direct library construction using self-corrected PacBio reads. (**A**) Identification of intact LTR elements and construction of libraries using the Arabidopsis “L*er*-0” genome and 20k - 180k self-corrected PacBio reads. (**B**) The performance of custom LTR libraries compared with that from the Arabidopsis reference (Col-0) genome.

Table 1.
LTR-RTs with non-canonical motifs from 50 sequenced plant genomes.

Motif	Copia	Gypsy	Unknown	Total
TACA	82	12	240	334
TGTA	111	17	157	285
TATA	36	3	123	162
TGCT	11	0	32	43
TGGA	10	1	19	30
TACT	4	0	6	10
TCCA	6	0	0	6

Table 2.
Element size of different types of LTR-RTs in 50 sequenced plant genomes.

	Non-TGCA LTR-RT					TGCA LTR-RT				
	Count	Percentage	LTR (bp)	IN (bp)	Total (bp)	Count	Percentage	LTR (bp)	IN (bp)	Total (bp)
copia	255	29.2%	272	4435	4979	14854	17.8%	911	5765	7588
gypsy	34	3.9%	1115	5044	7273	42667	51.2%	1288	7352	9928
unknown	583	66.9%	233	4684	5151	25847	31.0%	1184	4656	7025
All LTR	872	100%	279	4625	5184	83368	100%	1189	6234	8611

Table 3.

Performance of LTR_retriever on model plant genomes.

Genomes	Rice Nipponbare	Sacred Lotus	Maize B73 v4	Maize Mo17	Arabidopsis*
Lib size (MB)	5.92	2.75	35.97	2.57	1.21
Std-lib masking	23.53%	28.70%	75.40%	77.44%	6.98%
Fraction masked	25.30%	29.61%	70.08%	75.05%	7.43%
Run time (-t 20)	42 min	2.08 h	94.88 h	24.8 h	10 min
Sensitivity	91.70%	89.35%	91.10%	95.65%	91.17%
Specificity	96.86%	91.26%	90.58%	95.66%	98.92%
Accuracy	95.65%	90.70%	90.97%	95.65%	98.38%
Precision	89.99%	81.18%	96.61%	98.69%	86.33%

*Redundancy of the Arabidopsis library is not reduced since it is already very compact.

Supplementary Table S1.

Performances of LTR-RT recovery programs on the rice genome.

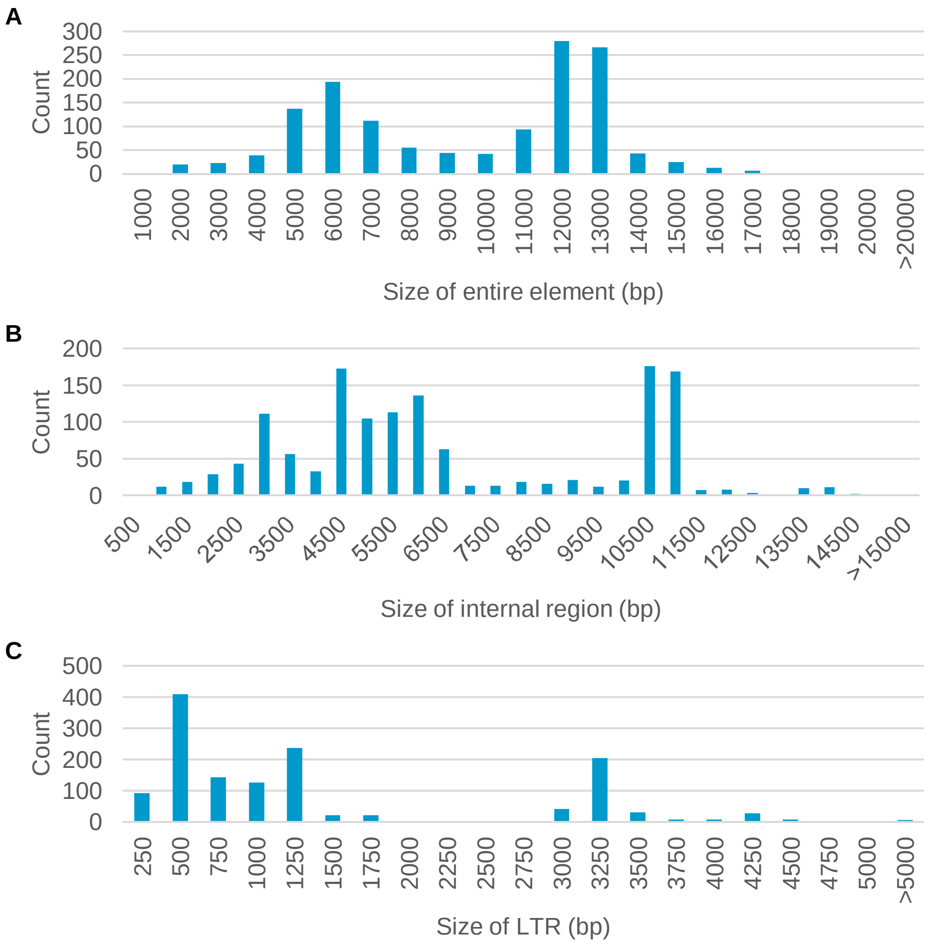
Input source[Z]	Lib size (MB)	Fraction masked	Sensitivity	Specificity	Accuracy	Precision
Jiang_rice6.9.lib_LTR[Y]	2.34	23.53%	100%	100%	100%	100%
LTR_STRUC[X]	9.74	33.79%	94.79%	84.98%	87.29%	66.02%
LTRharvest[X]	11.30	38.27%	95.32%	79.17%	82.96%	58.32%
LTRharvest[W]	7.34	31.28%	92.95%	87.70%	88.94%	69.94%
MGEScan-LTR[W]	44.44	43.47%	94.64%	72.28%	77.54%	51.24%
LTR_finder[W]	14.10	28.62%	92.23%	90.95%	91.25%	75.83%
LTR_retriever (Har_dft)	4.17	24.39%	91.57%	96.29%	95.18%	88.38%
LTR_retriever (Har)	4.43	23.87%	91.16%	96.84%	95.50%	89.88%
LTR_retriever (MGE)	2.24	19.31%	77.27%	98.54%	93.53%	94.21%
LTR_retriever (Fin)	5.74	25.29%	94.38%	95.98%	95.61%	87.85%
LTR_retriever (Har+MGE)	4.81	23.99%	91.70%	96.86%	95.65%	89.99%
LTR_retriever (Har+Fin)	5.92	25.30%	94.48%	95.99%	95.64%	87.90%
LTR_retriever (Fin+MGE)	5.90	25.66%	94.44%	95.52%	95.26%	86.65%
LTR_retriever (Har+Fin+MGE)	6.06	25.66%	94.52%	95.53%	95.30%	86.70%
LTR_retriever (Har+Fin+MGE+NMTF)	6.07	26.15%	94.52%	94.90%	94.81%	85.08%

[Z] Input source for LTR_retriever is indicated in parentheses obtained using optimized parameters unless notified. Har_dft, LTRharvest with default parameters; Har, LTRharvest; MGE, MGEScan-LTR; Fin, LTR_finder; NMTF, LTRharvest with the default “-motif” parameter.

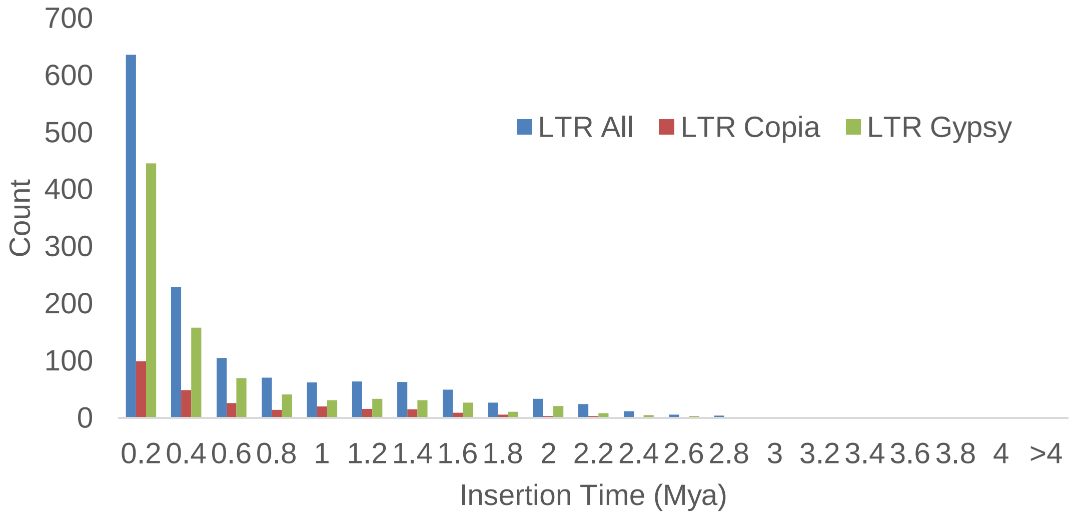
[Y] Jiang_rice6.9.lib_LTR, the manually curated standard library.

[X] Programs were run using default parameters.

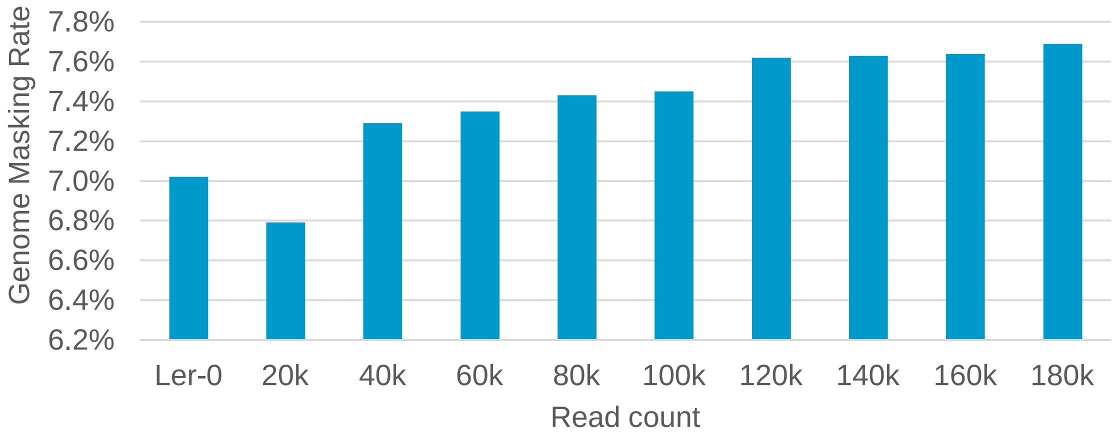
[W] Programs were applying optimized parameters except noted.



Supplementary Figure S1. Size distributions of (**A**) full-length LTRs, (**B**) internal regions, and (**C**) LTR regions in the rice genome.



Supplementary Figure S2. Insertion time distributions of intact LTRs in the rice genome. Mutation rate μ=1.3*10^{-8} per base pair per year.



Supplementary Figure S3. Masking efficiency of LTR libraries derived from PacBio reads of the Arabidopsis "L*er*-0" genome. The standard LTR library from the reference genome (Col-0) was used to mask the assembled "L*er*-0" genome. Custom libraries generated from the Arabidopsis "L*er*-0" genome and 20k - 180k self-corrected PacBio reads were used to mask the Arabidopsis "L*er*-0" genome.