Published online 12 November 2021

NAR Genomics and Bioinformatics, 2021, Vol. 3, No. 4 **1**
https://doi.org/10.1093/nargab/lqab108

TSSr: an R package for comprehensive analyses of TSS sequencing data

Zhaolian Lu[1,†], Keenan Berry[2,†], Zhenbin Hu[1], Yu Zhan[1], Tae-Hyuk Ahn[2,3,*] and Zhenguo Lin[1,2,*]

[1]Department of Biology, Saint Louis University, St. Louis, MO 63103, USA, [2]Program of Bioinformatics and Computational Biology, Saint Louis University, St. Louis, MO 63103, USA and [3]Department of Computer Sciences, Saint Louis University, St. Louis, MO 63103, USA

Received June 04, 2021; Revised October 05, 2021; Editorial Decision October 26, 2021; Accepted October 27, 2021

ABSTRACT

Transcription initiation is regulated in a highly organized fashion to ensure proper cellular functions. Accurate identification of transcription start sites (TSSs) and quantitative characterization of transcription initiation activities are fundamental steps for studies of regulated transcriptions and core promoter structures. Several high-throughput techniques have been developed to sequence the very 5′end of RNA transcripts (TSS sequencing) on the genome scale. Bioinformatics tools are essential for processing, analysis, and visualization of TSS sequencing data. Here, we present TSSr, an R package that provides rich functions for mapping TSS and characterizations of structures and activities of core promoters based on all types of TSS sequencing data. Specifically, TSSr implements several newly developed algorithms for accurately identifying TSSs from mapped sequencing reads and inference of core promoters, which are a prerequisite for subsequent functional analyses of TSS data. Furthermore, TSSr also enables users to export various types of TSS data that can be visualized by genome browser for inspection of promoter activities in association with other genomic features, and to generate publication-ready TSS graphs. These user-friendly features could greatly facilitate studies of transcription initiation based on TSS sequencing data. The source code and detailed documentations of TSSr can be freely accessed at https://github.com/Linlab-slu/TSSr.

INTRODUCTION

Gene transcription is finely regulated to ensure proper cellular functions. Most transcriptional regulatory signals are integrated as inputs to control the process of transcription initiation (1,2). In eukaryotes, transcription of protein-coding RNA (mRNA) and several classes of non-coding RNAs are carried out by the preinitiation complex (PIC) that consists of RNA polymerase II (pol II) and many general transcription factors. PIC assembly occurs at core promoter regions and positions pol II to initiate transcriptions from transcription start sites (TSSs) (3). Recent studies revealed that transcription of most eukaryotic genes can be initiated from multiple core promoters, and each core promoter may contain an array of neighboring TSSs (1,4,5). Furthermore, alternative usage of different core promoters by a gene was found prevalent, which was believed to play an important role in gene function and regulation (4,5). Transcription initiation from undesired TSSs was found associated with many human diseases, such as breast cancer, Alzheimer's disease, etc. (6,7). Therefore, identification of TSSs and characterization of core promoter activities are fundamental steps that toward better understandings of regulatory mechanisms of gene expression and how cells perform their functions.

Several high-throughput sequencing techniques have been developed to sequence the very 5′ends of RNA transcripts for TSS identification on the genome scale. Examples of these techniques include cap analysis of gene expression (CAGE) (8), nano–cap analysis of gene expression (NanoCAGE) (9), transcript leader sequencing (TL-seq) (10), transcript isoform sequencing (TIF-seq) (11), TSS-seq (12), RAMPAGE (13,14), single-cell tagged reverse transcription (STRT) (15), global nuclear run-on cap (GRO-cap) (16), MAPCap (17) and STRIPE-seq (18). The 5′end reads of transcripts generated by these high-throughput techniques can be used to determine the origins of transcription at single-nucleotide resolution, providing accurate 5′boundary information of genes for genome annotations. In addition, the high-resolution TSS map allows us to identify core promoters by grouping neighboring TSSs that form distinct TSS clusters (TCs) (19). Examination of

*To whom correspondence should be addressed. Tel: +1 314 977 9816, Email: zhenguo.lin@slu.edu
Correspondence may also be addressed to Tae-Hyuk Ahn. Email: taehyuk.ahn@slu.edu
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

sequence context near core promoters facilitates the discovery of core promoter elements and other regulatory motifs, contributing to a better understanding of the genetic architecture of gene promoters and regulatory networks. TSS sequencing data also quantify the abundance of transcripts initiated from each TSS and core promoter. The quantitative information is valuable for characterizations of regulated transcription initiation activities in different cell types or in response to environmental stimuli at the level of individual TSS or core promoter.

Another important application of TSS data is to study core promoter shape, which is the distribution of TSS signals within a core promoter. Transcription initiation from some core promoters mainly occurs from one predominant TSS, recognized as a 'sharp' shape of core promoters. In contrast, transcription initiation activities in some core promoters are highly dispersed, forming a 'broad' shape of core promoters (4,20). Recent evidence demonstrated that core promoter shape is a genetic trait that reflects distinct regulatory mechanisms of transcription initiation (4,20,21). For example, 'broad' core promoters in *Drosophila* are over-represented in ubiquitously expressed genes, while 'sharp' core promoters are mostly found in tissue-specific expressed genes (20). Thus, characterization of core promoter shape could provide valuable insights into studies of regulated transcription. Several quantification algorithms of core promoter shape have been reported, such as inter-quantile width of core promoters (22), shape index (20) and promoter shape score (5).

In summary, interrogations of TSS sequencing data are essential for accurate genome annotation and studies of regulated gene transcription. Multiple bioinformatics tools have been developed for analyzing TSS sequencing data, such as CAGEr (22), TSRchitect (23), CAGEfightR (24), icetea (17) and TSRexploreR (25). Most of these tools provide functions for identifying TSSs from aligned sequencing reads, identifying core promoters, quantification of core promoter shape, and differential expression analyses. These bioinformatics tools have greatly facilitated the studies of transcription initiation. Our previous studies based on a large collection of TSS sequencing data from various organisms identified several areas of improvement for bioinformatic processing of TSS data (5,26–28). Here, we present a novel R package 'TSSr' by implementing several new algorithms we developed to provide rich functions for processing and comprehensive analyses of different types of TSS sequencing data. These new algorithms address existing issues related to TSS calling, TSS clustering, filtering of reads that are generated by technical artifacts, and identification of bona fide core promoters. TSSr also provides a variety of utilities for downstream analyses of TSS data, such as annotation of core promoters and generation of publication-quality TSS graphs. These features make TSSr an all-in-one tool for comprehensive interrogations of TSS sequencing data.

MATERIALS AND METHODS

Development of TSSr as an R package

We developed TSSr, an R package for comprehensive analyses of TSS sequencing data generated by different library preparation protocols. TSSr provides a variety of functions for processing and analysis of TSS data, such as identification of TSSs from mapped sequencing reads, core promoter identification by TSS clustering, core promoter annotation, quantification of core promoter shape and alternative usage of core promoter, inference of enhancers, differential expression analysis, as well as generation of various vector graphics for publication (Figure 1). The package also includes detailed descriptions of each function of TSSr, example commands and default parameters. The source code of TSSr package is available at GitHub https://github.com/Linlab-slu/TSSr.

Workflow of TSSr

The workflow and functions of TSSr are illustrated in Figure 1. TSSr accepts two types of input data: read alignment files or TSS tables. The read alignment files in compressed binary alignment map (BAM) format are required if users intend to call TSSs from raw sequencing data. BAM files can be derived from mapping of either paired-end or single-end TSS sequencing reads. Users should set 'inputFilesType' as 'bam' for single-end reads and as 'bamPairedEnd' for paired-end BAM files. To provide more accurate quantification of transcription initiation events at each TSS, we recommend excluding reads mapped to rRNA from BAM files before TSS calling and subsequent analyses. Removal of rRNA reads from BAM files can be carried out by rRNAdust (https://fantom.gsc.riken.jp/5/sstar/Protocols:rRNAdust) based on rRNA sequences provided by users. rRNAdust adds a value of 512 to the FLAG score for each rRNA reads, which will be excluded by TSSr during TSS calling. The reference genome stored as a BSgenome data package must be provided for TSS calling. TSSr also accepts TSS tables generated by TSSr or other bioinformatics tools as input data. A TSS table can be a tab-delimited text file, a BigWig (bw) binary type file, or browser extensible data (bed) type file. A tab-delimited TSS table contains chromosome ID, genomic coordinates, strand information, and raw or normalized read counts of each sample. An example tab-delimited TSS table is included in the package. Users should set 'inputFilesType = TSStable' in this scenario.

Before TSS calling, TSSr removes reads that are below certain sequencing quality and mapping quality. The default threshold for Phred quality score is 10, and mapping quality (MAPQ score) is 20. Users may change these parameters by setting different values for 'sequencingQualityThreshold' and 'mappingQualityThreshold' when running the 'getTSS' function.

The TSS data are stored as an S4 class R object. Based on the provided sample list, TSS data from different biological replicates can be merged and normalized as TPM (Tags per million mapped tags). Users may use the 'filterTSS' option to remove TSSs with low support from mapped reads based on TPM value or *P*-value inferred by 'Poisson distribution'. The matrix of raw counts or normalized TPM of each TSS can be exported to either as tab-delimited files or bedGraph/BigWig files which can be visualized by the UCSC Genome Browser (29) or Integrative Genomics Viewer (IGV) (30). The consistency between bi-

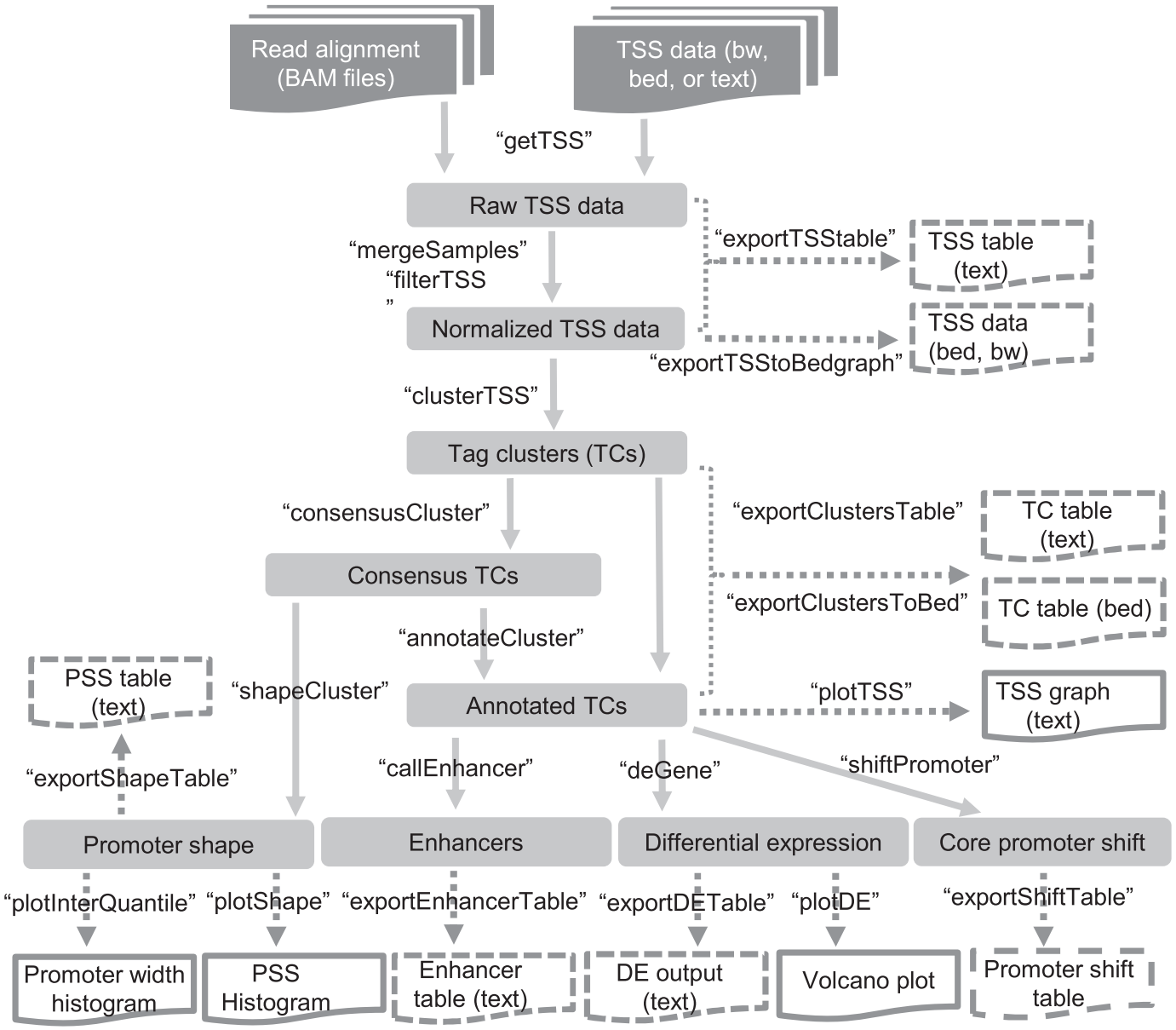

Figure 1. Workflow of TSSr. The flow chart illustrates main steps and functions of TSSr. The function names for each step are shown inside the quotation marks.

ological replicates and relationships among RNA samples can be inspected by plotting correlations of TSS tags (Supplementary Figure S1A) using 'plotCorrelation' or principal component analysis (PCA) using 'plotTssPCA' in TSSr (Supplementary Figure S1B).

After normalizing and filtering TSS data, the 'clusterTSS' function is then used to group neighboring TSSs into distinct TSS clusters (TCs), representing putative core promoters. For each TC in every sample, its dominant TSS (TSS with the highest TPM value) and boundaries will be inferred. Due to its dynamic nature, transcription initiation activities within the same core promoters may vary substantially in different cell types or growth environments; thus, the position of dominant TSS and boundaries of the same TC may be different among samples. TSSr infers a set of consensus core promoters using the 'consensusCluster' function to assign the same ID for TCs belonging to the same core promoter, which allows subsequent comparative studies across samples. TCs from different samples are considered to belong to the same consensus core promoter if the distance of their dominant TSSs is smaller than a user-defined distance (default 'dis = 50' bp).

TSSr assigns a TC to its downstream genes using the 'annotateCluster' function based on the distance between its dominant TSS and the 5′boundary of its immediately downstream gene. Differential gene expression analysis can be performed in TSSr using the 'deGene' function, which is carried out by the DESeq2 package. TSSr also implements various algorithms to quantify core promoter shape ('shapeCluster'). The degree of core promoter shift can be

inferred using the 'shiftPromoter' function in TSSr. Major functions and implemented algorithms in TSSr are described in the Results section.

Comparisons of the functionalities and features between TSSr and other R packages for TSS analysis, including CAGEr, TSRexploeR, TSRchitect, CAGEfightR and icetea, are summarized in Supplementary Table S1.

Example data

We used a subset of CAGE sequencing data obtained from our previous study to demonstrate the functionality of the TSSr package (5). The example data include four BAM files (SL01, SL02, SL03 and SL04). SL01 and SL02 are two biological replicates of CAGE reads obtained from *Saccharomyces cerevisiae* grown in rich medium (YPD), while SL03 and SL04 were obtained by treating *S. cerevisiae* with α factor (Arrest). The CAGE reads were mapped to the reference genome of *S. cerevisiae* (R64-2-1) using HISAT2 (31). To reduce file size, each BAM file only includes sequencing reads mapped to two chromosomes (Chr I and Chr II). The example BAM files can be downloaded from http://www.zlinlab.org/TSSr.html. The BSgenome object of *S. cerevisiae* 'BSgenome.Scerevisiae.UCSC.sacCer3' was obtained from BSgenome Bioconductor package (https://bioconductor.org/packages/BSgenome). Genome annotation of *S. cerevisiae* (R64-2-1) was downloaded from the Saccharomyces Genome Database (32).

RESULTS

TSS calling based on an improved algorithm

Mapping the exact position of TSSs from read alignment files is a crucial step of TSS sequencing data processing. Due to technical artifacts and stochastic transcriptional activities, a portion of TSSs inferred from TSS sequencing reads may not represent bona fide TSSs, which should be removed from the TSS list. The most significant structural feature of the 5′end of mRNA transcripts is the presence of a cap structure (e.g. N7-methylated guanosine, or m7G) which was added during transcription. The m7G cap protects the transcript from exonuclease cleavage and is required for cap-dependent initiation of protein synthesis (33,34). For techniques based on cap capturing, such as CAGE, m7G was reverse transcribed and sequenced, referred to as systematic G nucleotide addition (4). CAGE reads with one or more uncoded 5′ end Gs provide direct evidence of their origination from complete mRNA molecules.

TSSr implements a novel algorithm for determining TSSs from qualified mapped TSS sequencing reads (27). In brief, if a mapped TSS sequencing read starts with one or more G that mismatch to the reference genome, the uncoded 5′ end Gs are likely the m7G cap, and thus they will be removed from TSS calling. However, when the first nucleotide at the 5′ end of sequencing reads is G, and it matches with the reference genome, it is unclear whether the G is the m7G cap or not. In this scenario, CAGEr proportionally splits the reads into capped and uncapped based on the percentage of reads with unmatched G in the sample (22). This strategy yields two neighboring TSSs for reads mapping to the same position. TSSr uses a different strategy of TSS calling

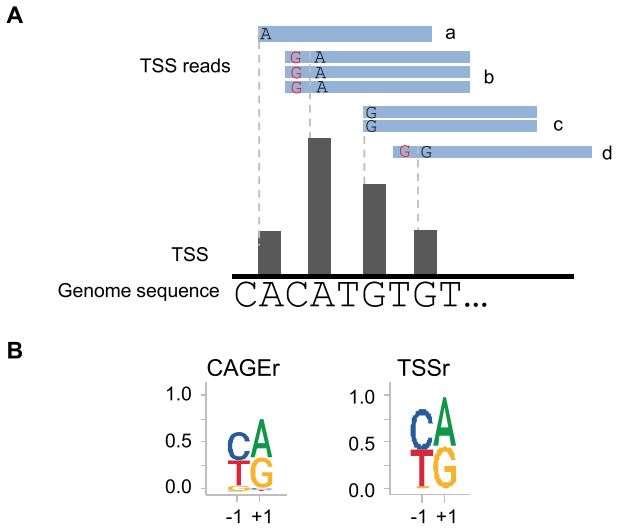

Figure 2. Algorithm of TSS identification from mapped read data. (**A**) Schematic representation of TSS calling by TSSr based on mapped reads. An uncoded G at the 5′end of transcripts is considered as a G cap, and TSS is called as the next position of the read alignment. A matched G at the first mapping site is not removed from TSS calling as it is unlikely an added G cap. The height of bars represents the number of reads mapped to a TSS. (**B**) Sequence logos of pyrimidine-purine dinucleotide at positions −1 and +1 of TSSs defined by CAGEr and TSSr using the same dataset.

based on the known sequence context of TSS. Specifically, transcription initiation in all living organisms is mostly initiated at a purine (A or G) preceding by a pyrimidine (C or T), demonstrating a strong preference of pyrimidine-purine (PyPu) dinucleotide at the [−1, +1] positions of TSS (4,27,35). Therefore, if a matched G at the 5′end of a tag is considered as an added cap (4), removing the matched G usually results in a TSS with non-PyPu dinucleotide at [−1, +1], which is extremely unlikely. Therefore, the proportional split of reads with a matched G could lead to many TSSs that lack the PyPu dinucleotide at the [−1, +1] sites (Figure 2A). To address this issue, TSSr treats the 5′end of reads with matched G as genome-coded G, and the first G is not removed when calling TSS positions (Figure 2A). This strategy makes TSSr also suitable for calling TSSs from 5′end sequencing reads that are not based on cap capture techniques. The differences in TSS calling between TSSr and CAGEr for reads with genome-coded G at 5′end are illustrated by an empirical example in Supplementary Figure S2. Comparison of the sequence preference of TSSs called by TSSr and CAGEr shows that TSSr-called TSSs demonstrate a much stronger preference of PyPu dinucleotide at the [−1, +1] sites (Figure 2B, Supplementary Table S2), supporting a more accurate TSS calling.

During read mapping, users may use 'soft-clipping' option to ignore bases from either side of reads that do not match well to the reference genome. By excluding unmatched bases of read ends from read mapping, soft-clipping could increase the overall mapping rate. However, soft-clipping ignores not only non-genome-matching Gs, but also other non-matching 5′end bases, which introduces false-positive TSSs. Among those reads with soft-clipping of multiple bases at the 5′end, only a small portion are un-

coded Gs. Therefore, we recommend not to use soft-clipping for TSS sequencing data. If the bam files are generated with 'soft-clipping', the 'softclippingAllowed' argument of 'getTSS' function in TSSr should be set as 'TRUE'. TSSr defines the 5′-most non-soft-clipped base as TSS for soft-clipped bam files.

TSSr reduced false-positive TSSs by filtering non-bona fide TSSs

Stochastic and cryptic transcription was found prevalent in eukaryotic cells (5,36). In addition, technical artifacts of library preparation also include some non-5′end RNA fragments in TSS sequencing libraries (27). Thus, it is necessary to remove as many non-bona fide TSSs as possible to provide more accurate mappings of the 5′boundaries of transcripts and inference of core promoters. As these TSSs tend to be supported by a lower number of sequencing reads, previous studies remove TSSs based on the number of supported tags or normalized value, e.g., TPM. Considering that the effectiveness of these filtering methods depends on sequencing depths and genome size, TSSr implements a new filtering method that estimates the statistical significance of read support for each called TSS given a sequencing depth and genome size (27). In brief, TSSr calculates the probability of observing k numbers of reads supporting each TSS based on the sequencing depth of the sample per the Poisson distribution (27). Only TSSs with a significantly larger number of supporting reads than expected (default threshold $P < 0.01$) are considered as qualified TSSs. Non-significant TSSs are thus filtered by TSSr. Alternatively, users may choose the 'tpm' method to filter TSSs that below a user-defined threshold (default TPM threshold = 0.1).

Identification of core promoters using the peak-based clustering (peakclu) method

Reconstruction of core promoters by clustering nearby TSSs is a challenging task due to the presence of transcriptional noises and technical artifacts. Several algorithms have been developed to cluster TSSs to infer core promoters. Frith *et al*. (37) developed a parametric clustering (paraclu) algorithm that attempts to find genomic intervals that contain stronger TSS signals than their surrounding regions, which can be contained within each other and generate a hierarchy of peaks. Collapsing the overlapped regions into a single level of the peak hierarchy by excluding all peaks contained within others might give rise to broad clusters, which is not explanatory from a biological perspective. RECLU improved paraclu to identify reproducible clusters across replicas based on the irreproducible discovery rate (IDR) (38). A distance-based clustering ('distclu'), which was implemented in the CAGEr package (22) reconstruct promoters based on the distances between neighboring TSSs, which might result in super-broad clusters if there are continuous weak signals adjacent to strong ones or many trivial clusters when neighboring TSSs are in a greater distance than the maximal allowed distance. ADAPT-CAGE distinguishes between CAGE signal derived from TSSs and transcriptional noise using a Machine Learning framework, which relies on histone modification data and genomic location (39).

In the TSSr package, we implemented a newly developed TSS clustering algorithm based on peaking identification, namely 'peakclu' (peak clustering) (27). Briefly, peakclu applies a sliding window approach (default window size = 100 bp with step size = 1) to scan TSS signals from the 5′ end of both strands of each chromosome (Figure 3A). In each window, the TSS with the highest TPM value was identified as the peak. The surrounding TSSs are grouped with the peak into a TC. The clustering process of a TC terminates if a TSS is $\geq n$ bp (default $n = 30$) away from the nearest upstream TSS. In addition to setting a minimal allowed distance between peaks, TSSr offers another option to set maximal allowed extension distance between neighboring TSSs around peaks, which enables users to define the boundaries between neighboring core promoters. Based on empirical data, it shows that the peakclu algorithm provides a better way of identifying core promoters, which reduces the risk of joining small TSS clusters together (Figure 3B-C). TSSr calculates inter-quantile width of a TC based on the cumulative distribution of TSS signals within the TC. The positions of the 10th to 90th quantiles of TSS signals, which include at least 80% transcription initiation signals within a cluster, were defined as the 5′ and 3′ boundaries of the core promoter (Figure 3A).

We also noticed that many low TPM TCs are commonly found downstream of a highly expressed TC, and most of these weak TCs are located within coding regions. These weak downstream TCs could be produced by stochastic transcription initiation, transcript recapping events, or inclusions of partial RNA fragments generated during library preparation. Therefore, it is unlikely that these TCs represent genuine core promoters, and it is better to remove them from subsequent analyses. TSSr implements a local filtering strategy to remove these TCs if their signals are lower than a user-defined percentage of the strongest upstream TC of a gene (known as a representative core promoter) (Figure 3B, C). The default threshold of local filtering is 0.02. We evaluated the impacts of different thresholds of local filtering on the total number of inferred TCs and assigned TCs (those located in canonical promoter regions) by using four 'localThreshold' values (localThreshold = 0, 0.02, 0.06 and 0.1,). As shown in Supplementary Table S3, an increase in local filtering threshold increases the total number of inferred TCs, while it reduces the maximum inter-quantile width of TCs, suggesting that stronger local filtering could split broad TCs into multiple TCs. However, the numbers of TCs assigned to genes are largely unaffected, supporting the local filtering mainly influences TSS clustering in coding regions (Supplementary Table S3). With two layers of local filtering on TSSs and TCs, TSSr reduces potentially false-positive core promoters of genes (Figure 3B, C). From a genome-scale perspective, TSSr significantly reduces broad core promoters of which inter-quantile widths are >100 bp (Figure 3D).

Quantification of core promoter shape

TSSr provides three different options, inter-quantile width, shape index (SI), and promoter shape score (PSS), to quantify core promoter shape. Inter-quantile width refers to the distance between the locations of the 10th percentile to the

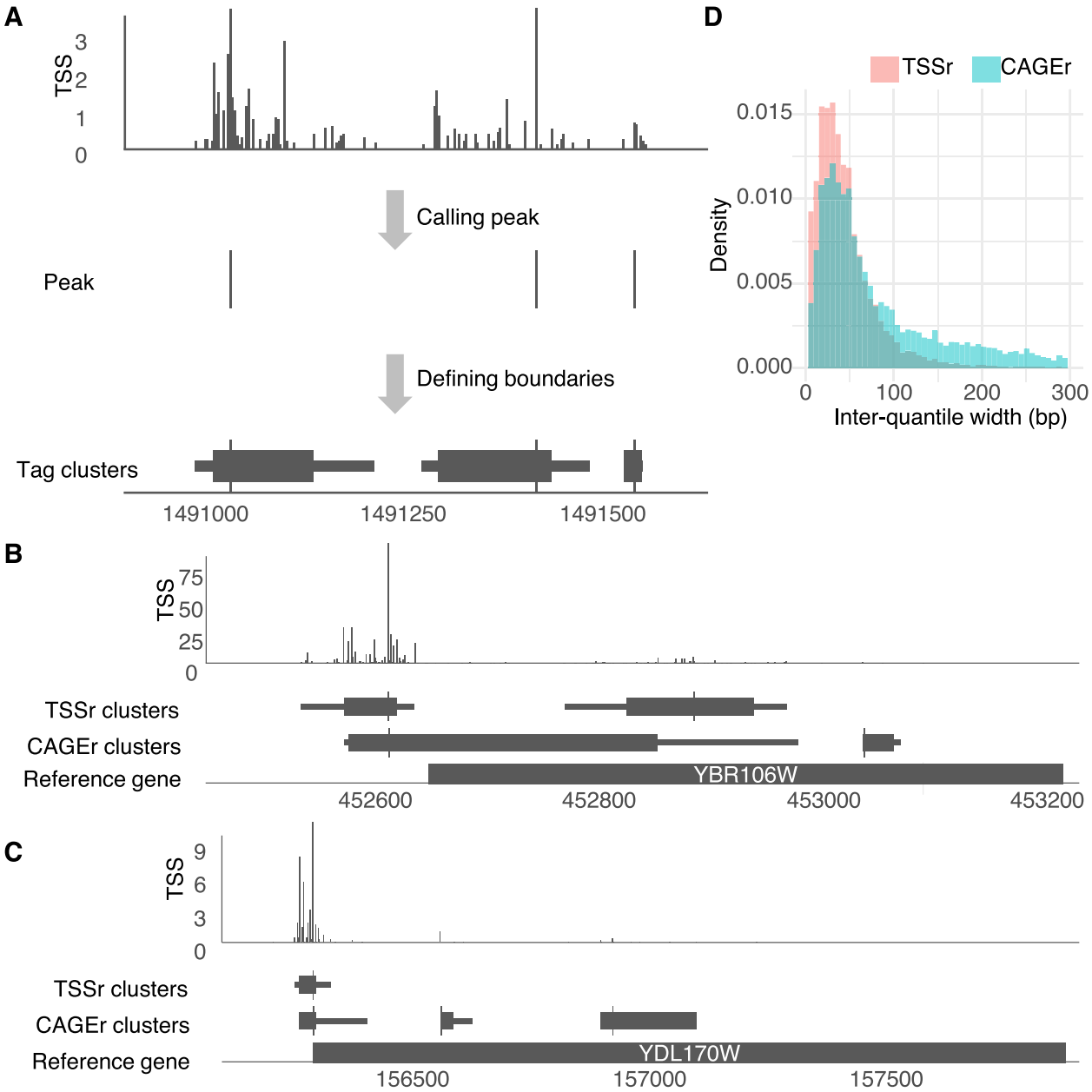

Figure 3. Peak-based clustering algorithm (peakclu). (**A**) A schematic representation of 'peakclu' method implemented in TSSr. The thinner box represents the starting and ending points of a TSS cluster. The thicker box indicates its inter-quantile boundaries (from 10th percentile to 90th percentile). The vertical line represents the peak or dominant TSS of a TSS cluster. (**B**) An example of TSS clustering around gene *YBR106W*. (**C**) An example of TSS clustering around gene *YDL170W*. (**D**) Histogram showing the distribution of inter-quantile widths of TSS clusters defined by the 'peakclu' algorithm in TSSr and 'disclu' algorithm in CAGEr.

90th percentile TSS signals within a TSS cluster. Thus, it measures the width of a core promoter, but lacks the information of distribution patterns of TSS signals within a core promoter. Inter-quantile width could be significantly affected by different clustering methods. SI takes into consideration the distribution patterns of TSS signals within core promoters, while it ignores the spacing between different TSSs and core promoter width. PSS takes both core promoter inter-quantile width and distributions of TSS signals into consideration (5), providing a more accurate characterization of core promoter shape (Figure 4A). TSSr also offers a function to generate histograms of inter-quantile width, SI, and PSS values to illustrate the distributions of core promoter shape (Figure 4B).

Associate TCs to annotated genes as putative core promoters and inference of putative enhancers

Associate TCs to annotated genes as their core promoters is required for annotation of the 5′ boundaries of genomic features. This process is also a prerequisite for further interrogations of regulated transcription initiation at the gene

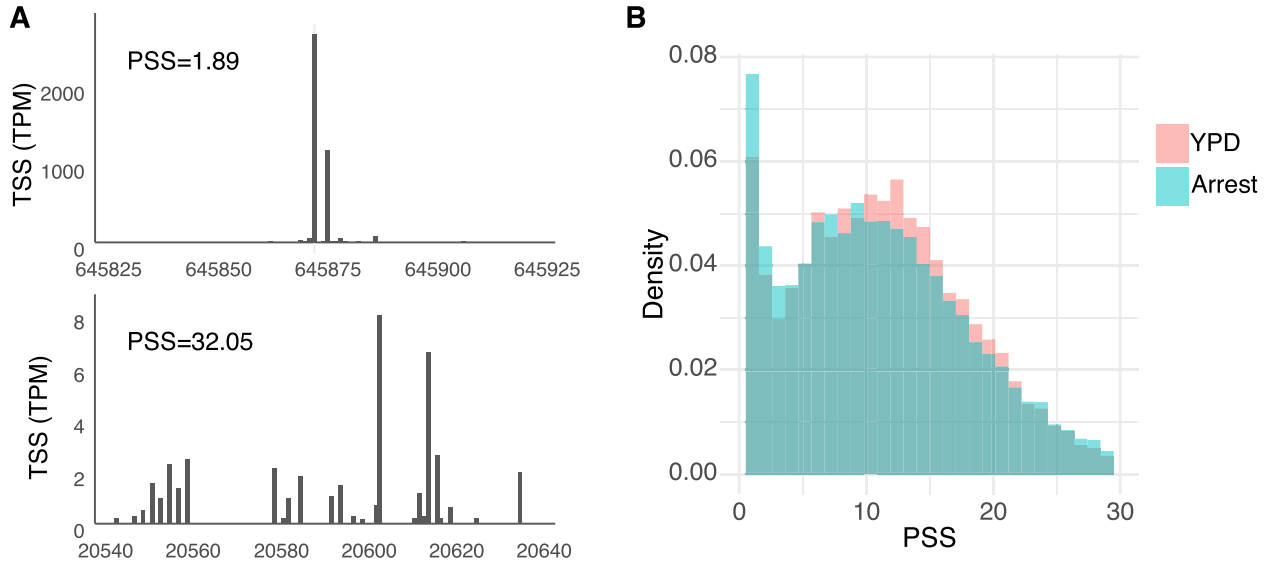

Figure 4. Core promoter shape. (**A**) Examples of sharp and broad core promoter shape and their corresponding PSS values. PSS starts from 0, which is the sharpest shape of core promoter that contains a single TSS. The PSS value increases with an increase of TSS number in a core promoter and more even distribution of TSS signals. (**B**) A density histogram demonstrates the different distribution of PSS values of the two example data (YPD versus Arrest) used in this study.

level. TSSr offers the 'annotateCluster' function to assign TCs to their downstream genes. By default, only TCs with ≥0.02 TPM are used for the annotation process. The assignment of a TC to a gene is based on the distance between the position of the dominant TSS of a TC and the annotated 5′ends of coding sequences (start codon of CDS) or transcripts (with annotated TSS). If the genome annotation does not include annotated TSSs, the user will need to use CDS for TC association (annotationType = 'genes'), and the default maximum distance between the dominant TSS and CDS is 1000 bp ('upstream = 1000'). For compact genomes, such as *S. cerevisiae*, a TC might overlap with the CDS of an upstream gene, so the user will need to set the maximum overlapping region between the dominant TSS of the TC and the 3′end of the overlapping CDS. By default, the distance must be less than 500 bp ('upstreamOverlap = 500'). These default parameters work best for compact genomes with limited introns, such as budding yeasts. A longer 'upstream' distance should be used for genomes with a much larger size or higher intron density. If the 'transcript' feature of genome annotation includes 5′UTRs or annotated TSSs, the user is recommended to use the 'transcript' feature for TC annotation because introns could be prevalent in 5′UTRs in many organisms. In this scenario, 'annotationType' should be set as 'transcript' (the default distance parameter is 500 bp). Because the genomes size and the number of introns vary substantially among organisms, it is necessary to apply customized criteria for TC assignment for different organisms. Users are advised to adjust the assignment criteria for core promoter assignment in TSSr based on the information included in genome annotation and gene structure of an organism (Figure 5A).

In higher eukaryotes, enhancers are usually found in remote locations of the corresponding transcription units and are independent of their orientation (40). Analysis of human CAGE data demonstrated that enhancer activity could be inferred based on the presence of balanced bidirectional capped transcripts (41). TSSr provides a function 'callEnhaner' to infer putative enhancers from unassigned TCs following the criteria defined in (41). In summary, two TCs are inferred as putative enhancer TCs if they are: (i) bidirectional; (ii) located within 400 bp; (iii) have a directionality score $|D| < 0.8$; and (iv) at least 2 kb away from the dominant TSS of core promoter in any annotated genes (user may change the default distance through the 'dis2gene' argument).

Differential expression analysis and core promoter shift

CAGE and other TSS sequencing data can be used to quantify transcription abundance and differential expression analysis, as each read represents a transcript. TSSr implements DESeq2 (42) for differential expression analysis between any pair of RNA samples. TSSr quantifies the read count of each gene as the total number of TSS reads in all TCs assigned to the gene, and the obtained raw read counts are then used as the input matrix for DESeq2. The results of differential expression analysis by DESeq2 can be exported as a tab-delimited text file by using the function 'exportDETable'. TSSr also provides 'plotDE' function to generate a volcano plot for visualization of the DESeq2 results (Figure 5B).

Differential expression analysis quantifies changes of transcription abundance for each gene between different samples, but it does not reflect alternative usage of core promoters within a gene. TSSr implements an algorithm ('shiftPromoter') to calculate the core promoter shift score D_s, which quantifies the degree of alternative usage of core promoters by a gene between different RNA samples (5). D_s is calculated based on the proportional changes of TSS reads in the two most highly expressed core promoters of a gene between two samples (Figure 6A). We only considered the

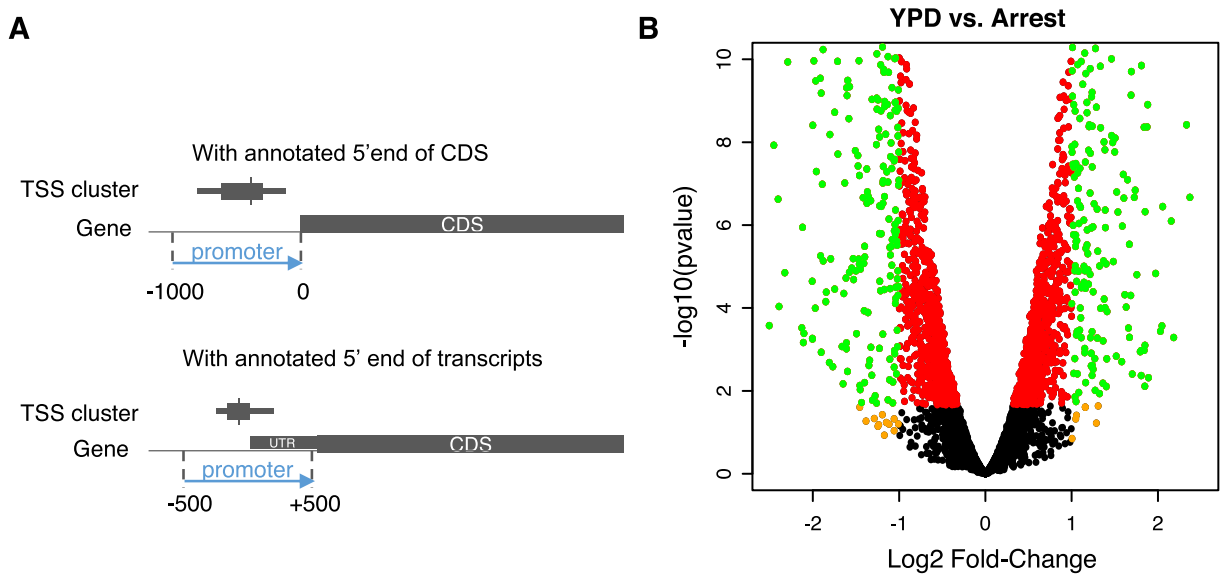

Figure 5. Assigning TCs to genes and differential gene expression. (**A**) Schematic illustration of different strategies of assigning TCs to a downstream gene depending on types of genome annotations (CDS or transcript). (**B**) Volcano plot shows differentially expressed genes (in red, $P < 0.01$ and $\log_2$ fold-change >1 or <-1 between two RNA samples (YPD versus Arrest) based on CAGE reads. Each dot represents a gene.

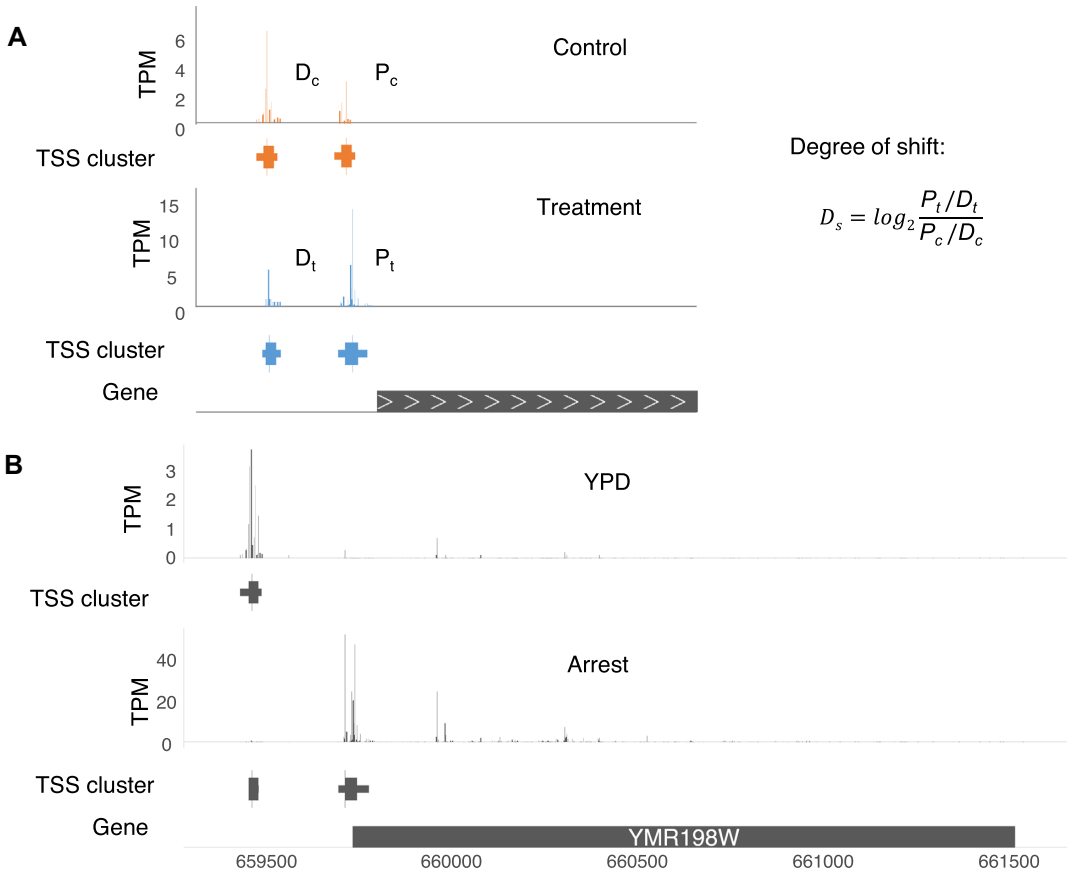

Figure 6. Core promoter shift. (**A**) Schematic illustration of genes with two core promoters and the equation of degree of core promoter shift (D_s). (**B**) An example of core promoter shift between different growth conditions (YPD versus Arrest).

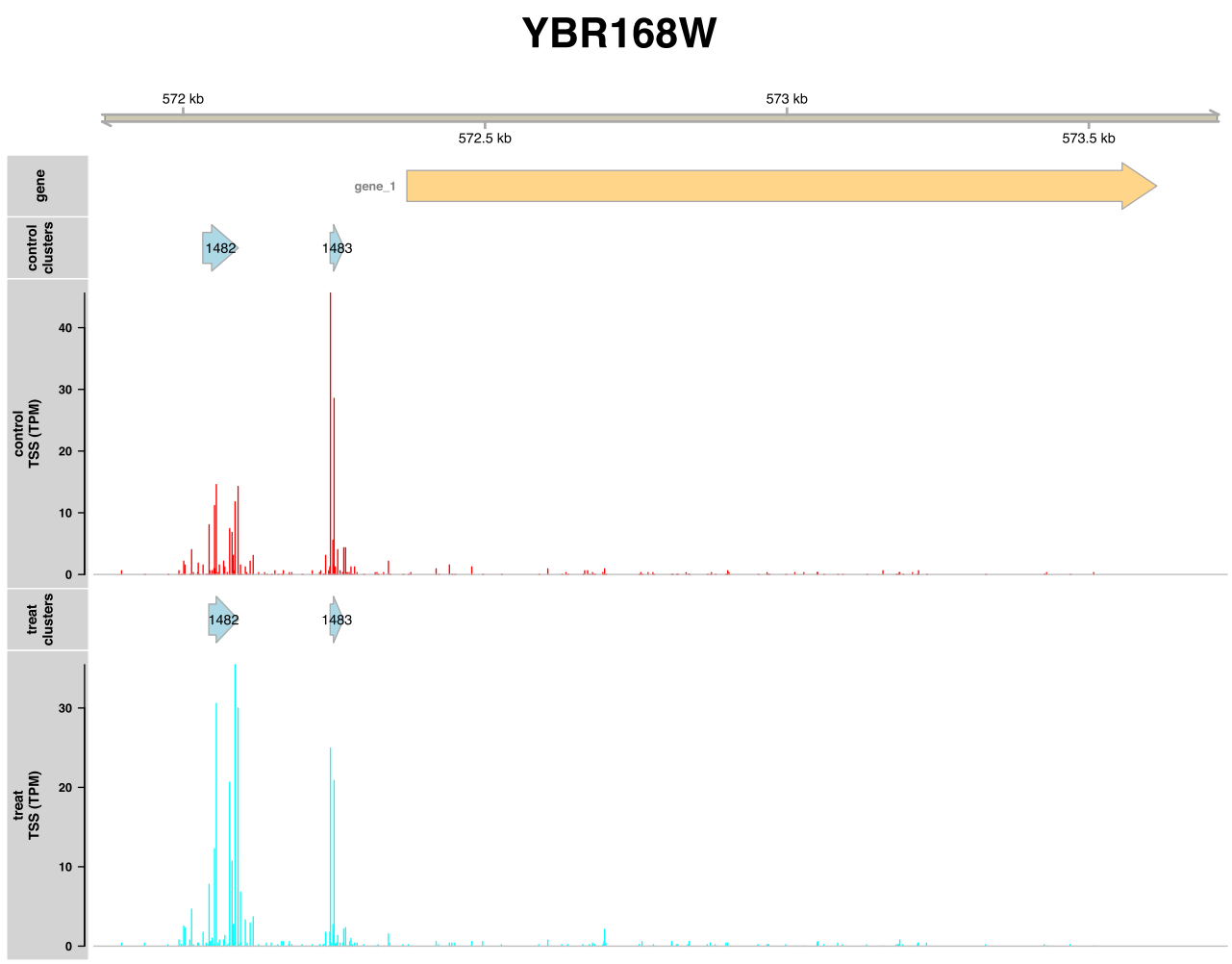

Figure 7. An unaltered output of TSS graph generated by TSSr. In this example, the TSS graph illustrates the distribution of TSS signals and TSS clusters near the YBR168W gene. Only TSS signals from the same strand are shown in this example. Users may set 'Bidirection = TRUE' to include TSS signals from both directions.

top 2 clusters for D_s calculation because the top 2 TCs usually account for >99% TSS signals in genes with ≥2 TCs. The statistical significance of the D_s score is inferred by the chi-square test, and an FDR value is calculated by adjusting *P*-value for multiple tests (by the Benjamini-Hochberg procedure). For example, two core promoters are present upstream of the *CIK1* gene (YMR198W) in *S. cerevisiae*, designated as distal core promoter (*D*) and proximal core promoter (*P*) here. Almost all transcription initiation signals are detected within distal core promoter (D_c) in yeast cells grown in rich medium (YPD). After treatment of α factor that arrests the yeast cells in G1 phase (Figure 6B), most transcription initiation signals have switched to the proximal core promoter (P_t), demonstrating a significant core promoter shift ($D_s = 15.1$, FDR $= 1.51 \times 10^{-8}$). If a gene contains more than two core promoters, only the two with the highest TPM values are selected to calculate the D_s score.

Transcription initiation from different core promoters of a gene yields different transcript isoforms. Thus, alternative usage of core promoters leads to differential transcript usage (DTU). Several RNA-seq bioinformatics tools, such as DRIMSeq (43), DEXSeq (44) or edgeR (45) provide a function to detect DTU using more sophisticated statistical tests. To compare the D_s statistics to these established methods for DTU, we obtained the raw read counts from the example data and used them as the input for DRIMSeq, which infer DTU based on reads of each transcript isoform. Among 75 genes with significant alternative usage of core promoters detected by TSSr based on the example data, 66 of them were also identified with significant DTU by DRIMSeq (Supplementary Table S4, adj. *P*-value < 0.01). In addition, all of the top 30 genes detected by TSSr are in the list of genes with significant DTU by DRIMSeq, suggesting good agreement between the two methods.

Generation of publication-quality TSS graphs

Visualization of TSS signals of genes provides visual insights into the complex transcription initiation landscape, such as the presence of multiple core promoters, different transcriptional activities among these promoters, promoter shape, and dynamic change of promoter activities across samples. TSSr includes the 'plotTSS' function to generate

publication-quality TSS graphs in pdf format. A TSS graph is a multitrack vector image that illustrates the locations, transcriptional direction, and signal strengths of each TSS near a gene in one or multiple samples (Figure 7). The top track indicates the genomic coordinates of the shown region, and the second track shows the location of the coding region of a gene of interest as a yellow arrow. Two data tracks are provided for each sample: a core promoter track and a TSS signal track. In the core promoter track, a core promoter is presented as a horizontal arrow, which indicates its transcriptional direction. The start and end points of the arrow represent the positions of the 10th to 90th percentile of TSS signals within a core promoter, and the arrow length indicates its inter-quantile width of the core promoter. The TSS track depicts the locations, direction, and TPM values of all TSSs in this region in a sample. TSS signals in this region are shown as bar graphs with a scale provided on the left side. Positive TPM values represent transcription initiation from the forward strand, while negative values mean the reverse strand. The TSS graphs are vector images that can be resized without any loss of quality, which can be used as publication-quality figures to illustrate the dynamic changes of transcription initiation across different samples. TSSr allows batch production of TSS graphs for a large number of genes simultaneously if users provide a list of gene names when running 'plotTSS'.

TSSr can handle a large number of human samples

We conducted benchmark tests using different numbers of TSS sequencing files obtained in human samples to test how TSSr performs for large genomes. Bertin *et al*. generated 62 CAGEscan libraries from 56 human RNA sources, with 6 of them were prepared in duplicate (46). The bam files of the 62 libraries were downloaded from the FANTOM5 database (47). We ran five benchmark tests for TSSr using different numbers of bam input files to obtain its running times and memory usage. In the first four tests, we randomly selected 4, 8, 16 and 32 libraries from the 62 bams as input for TSSr. For Test 5, all the 62 bams files were used. Because 50 of the 62 bam files lack biological replicates, we randomly group 4 bam files as biological replicates of a sample for benchmark tests. The running times were recorded using the Sys.time() function in R for four key functions of TSSr, which are time-consuming, including calling TSS from bam files (getTSS), TSS clustering (clusterTSS), calculating PSS (shapeCluster), and associating TCs to genomic features (annotateCluster). The R object size and memory usage were measured by pryr::object_size() and pryr::mem_used(). All the tests were carried out using a Dell PowerEdge T630 server (2 × Intel Xeon CPU E5-2640 @ 2.60 GHz, 128GB RAM, CentOS 7.0). In general, the processing times and object R size increases linearly with the increase of input file numbers. TSSr was able to complete the analyses of 62 human bam files for less than 8 hours, supporting that TSSr is capable of handling a large number of samples from large genomes with a single-node server.

DISCUSSION

TSSr is a user-friendly R package that provides a wide range of functions for comprehensive analyses of various types of TSS sequencing data, such as CAGE, TF-seq, and TIL-seq. One of the key features implemented in TSSr is a new strategy for identifying TSS based on the strong preference of pyrimidine-purine dinucleotides at position [−1, +1] of TSSs (5,20,48). In addition, a new TSS clustering algorithm, peakclu, was employed in TSSr for accurate identifications of core promoters. Furthermore, TSSr accepts multiple formats of input data and exports a variety of result tables, publication-ready graphs and data tracks, which can be easily visualized in UCSC Genome Browser or IGV, presenting a powerful tool for comprehensive TSS data analyses. This application would facilitate studies related to transcription initiation and its underlying genetic basis.

SUPPLEMENTARY DATA

Supplementary Data are available at NARGAB Online.

FUNDING

U.S. National Science Foundation [NSF 1951332 to Z.L., NSF 1564894 to T.A.].
Conflict of interest statement. None declared.

REFERENCES

1. Haberle,V. and Stark,A. (2018) Eukaryotic core promoters and the functional basis of transcription initiation. *Nat. Rev. Mol. Cell Biol.*, **19**, 621–637.
2. Juven-Gershon,T. and Kadonaga,J.T. (2010) Regulation of gene expression via the core promoter and the basal transcriptional machinery. *Dev. Biol.*, **339**, 225–229.
3. Smale,S.T. and Kadonaga,J.T. (2003) The RNA polymerase II core promoter. *Annu. Rev. Biochem.*, **72**, 449–479.
4. Carninci,P., Sandelin,A., Lenhard,B., Katayama,S., Shimokawa,K., Ponjavic,J., Semple,C.A., Taylor,M.S., Engstrom,P.G., Frith,M.C. *et al.* (2006) Genome-wide analysis of mammalian promoter architecture and evolution. *Nat. Genet.*, **38**, 626–635.
5. Lu,Z. and Lin,Z. (2019) Pervasive and dynamic transcription initiation in saccharomyces cerevisiae. *Genome Res.*, **29**, 1198–1210.
6. Arrick,B.A., Lee,A.L., Grendell,R.L. and Derynck,R. (1991) Inhibition of translation of transforming growth factor-beta 3 mRNA by its 5′ untranslated region. *Mol. Cell. Biol.*, **11**, 4306–4313.
7. Mihailovich,M., Thermann,R., Grohovaz,F., Hentze,M.W. and Zacchetti,D. (2007) Complex translational regulation of BACE1 involves upstream AUGs and stimulatory elements within the 5′ untranslated region. *Nucleic Acids Res.*, **35**, 2975–2985.
8. Murata,M., Nishiyori-Sueki,H., Kojima-Ishiyama,M., Carninci,P., Hayashizaki,Y. and Itoh,M. (2014) Detecting expressed genes using CAGE. *Methods Mol. Biol.*, **1164**, 67–85.
9. Salimullah,M., Sakai,M., Plessy,C. and Carninci,P. (2011) NanoCAGE: a high-resolution technique to discover and interrogate cell transcriptomes. *Cold Spring Harb. Protoc.*, **2011**, pdb prot5559.
10. Arribere,J.A. and Gilbert,W.V. (2013) Roles for transcript leaders in translation and mRNA decay revealed by transcript leader sequencing. *Genome Res.*, **23**, 977–987.
11. Pelechano,V., Wei,W. and Steinmetz,L.M. (2013) Extensive transcriptional heterogeneity revealed by isoform profiling. *Nature*, **497**, 127–131.
12. Malabat,C., Feuerbach,F., Ma,L., Saveanu,C. and Jacquier,A. (2015) Quality control of transcription start site selection by nonsense-mediated-mRNA decay. *eLife*, **4**, e06722.
13. Batut,P., Dobin,A., Plessy,C., Carninci,P. and Gingeras,T.R. (2013) High-fidelity promoter profiling reveals widespread alternative promoter usage and transposon-driven developmental gene expression. *Genome Res.*, **23**, 169–180.
14. Batut,P. and Gingeras,T.R. (2013) RAMPAGE: promoter activity profiling by paired-end sequencing of 5′-complete cDNAs. *Curr Protoc Mol Biol*, **104**, Unit 25B 11.

15. Islam,S., Kjallquist,U., Moliner,A., Zajac,P., Fan,J.B., Lonnerberg,P. and Linnarsson,S. (2012) Highly multiplexed and strand-specific single-cell RNA 5′ end sequencing. *Nat. Protoc.*, **7**, 813–828.
16. Core,L.J., Martins,A.L., Danko,C.G., Waters,C.T., Siepel,A. and Lis,J.T. (2014) Analysis of nascent RNA identifies a unified architecture of initiation regions at mammalian promoters and enhancers. *Nat. Genet.*, **46**, 1311–1320.
17. Bhardwaj,V., Semplicio,G., Erdogdu,N.U., Manke,T. and Akhtar,A. (2019) MAPCap allows high-resolution detection and differential expression analysis of transcription start sites. *Nat. Commun.*, **10**, 3219.
18. Policastro,R.A., Raborn,R.T., Brendel,V.P. and Zentner,G.E. (2020) Simple and efficient profiling of transcription initiation and transcript levels with STRIPE-seq. *Genome Res.*, **30**, 910–923.
19. Carninci,P., Kasukawa,T., Katayama,S., Gough,J., Frith,M.C., Maeda,N., Oyama,R., Ravasi,T., Lenhard,B., Wells,C. *et al.* (2005) The transcriptional landscape of the mammalian genome. *Science*, **309**, 1559–1563.
20. Hoskins,R.A., Landolin,J.M., Brown,J.B., Sandler,J.E., Takahashi,H., Lassmann,T., Yu,C., Booth,B.W., Zhang,D., Wan,K.H. *et al.* (2011) Genome-wide analysis of promoter architecture in drosophila melanogaster. *Genome Res.*, **21**, 182–192.
21. Schor,I.E., Degner,J.F., Harnett,D., Cannavo,E., Casale,F.P., Shim,H., Garfield,D.A., Birney,E., Stephens,M., Stegle,O. *et al.* (2017) Promoter shape varies across populations and affects promoter evolution and expression noise. *Nat. Genet.*, **49**, 550–558.
22. Haberle,V., Forrest,A.R., Hayashizaki,Y., Carninci,P. and Lenhard,B. (2015) CAGEr: precise TSS data retrieval and high-resolution promoterome mining for integrative analyses. *Nucleic Acids Res.*, **43**, e51.
23. Raborn,R.T., Spitze,K., Brendel,V.P. and Lynch,M. (2016) Promoter architecture and sex-specific gene expression in daphnia pulex. *Genetics*, **204**, 593–612.
24. Thodberg,M., Thieffry,A., Vitting-Seerup,K., Andersson,R. and Sandelin,A. (2019) CAGEfightR: analysis of 5′-end data using R/Bioconductor. *BMC Bioinformatics*, **20**, 487.
25. Policastro,R.A., McDonald,D.J., Brendel,V.P. and Zentner,G.E. (2021) Flexible analysis of TSS mapping data and detection of TSS shifts with TSRexploreR. *NAR Genomics and Bioinformatics*, **3**, lqab051.
26. McMillan,J., Lu,Z., Rodriguez,J.S., Ahn,T.H. and Lin,Z. (2019) YeasTSS: an integrative web database of yeast transcription start sites. *Database (Oxford)*, **2019**, baz048.
27. Lu,Z. and Lin,Z. (2021) The origin and evolution of a distinct mechanism of transcription initiation in yeasts. *Genome Res.*, **31**, 1–13.
28. Zhang,H., Lu,Z., Zhan,Y., Rodriguez,J., Lu,C., Xue,Y. and Lin,Z. (2021) Distinct roles of nucleosome sliding and histone modifications in controlling the fidelity of transcription initiation. *RNA Biol.*, **18**, 1642–1652.
29. Kent,W.J., Sugnet,C.W., Furey,T.S., Roskin,K.M., Pringle,T.H., Zahler,A.M. and Haussler,D. (2002) The human genome browser at UCSC. *Genome Res.*, **12**, 996–1006.
30. Robinson,J.T., Thorvaldsdottir,H., Winckler,W., Guttman,M., Lander,E.S., Getz,G. and Mesirov,J.P. (2011) Integrative genomics viewer. *Nat. Biotechnol.*, **29**, 24–26.
31. Kim,D., Langmead,B. and Salzberg,S.L. (2015) HISAT: a fast spliced aligner with low memory requirements. *Nat. Methods*, **12**, 357–360.
32. Cherry,J.M., Adler,C., Ball,C., Chervitz,S.A., Dwight,S.S., Hester,E.T., Jia,Y., Juvik,G., Roe,T., Schroeder,M. *et al.* (1998) SGD: Saccharomyces genome database. *Nucleic Acids Res.*, **26**, 73–79.
33. Both,G.W., Furuichi,Y., Muthukrishnan,S. and Shatkin,A.J. (1975) Ribosome binding to reovirus mRNA in protein synthesis requires 5′ terminal 7-methylguanosine. *Cell*, **6**, 185–195.
34. Muthukrishnan,S., Both,G.W., Furuichi,Y. and Shatkin,A.J. (1975) 5′-Terminal 7-methylguanosine in eukaryotic mRNA is required for translation. *Nature*, **255**, 33–37.
35. Zhang,Y., Degen,D., Ho,M.X., Sineva,E., Ebright,K.Y., Ebright,Y.W., Mekler,V., Vahedian-Movahed,H., Feng,Y., Yin,R. *et al.* (2014) GE23077 binds to the RNA polymerase 'i' and 'i+1' sites and prevents the binding of initiating nucleotides. *eLife*, **3**, e02450.
36. Raj,A. and van Oudenaarden,A. (2008) Nature, nurture, or chance: stochastic gene expression and its consequences. *Cell*, **135**, 216–226.
37. Frith,M.C., Valen,E., Krogh,A., Hayashizaki,Y., Carninci,P. and Sandelin,A. (2008) A code for transcription initiation in mammalian genomes. *Genome Res.*, **18**, 1–12.
38. Ohmiya,H., Vitezic,M., Frith,M.C., Itoh,M., Carninci,P., Forrest,A.R., Hayashizaki,Y., Lassmann,T. and FANTOM Consortium (2014) RECLU: a pipeline to discover reproducible transcriptional start sites and their alternative regulation using capped analysis of gene expression (CAGE). *BMC Genomics*, **15**, 269.
39. Georgakilas,G.K., Perdikopanis,N. and Hatzigeorgiou,A. (2020) Solving the transcription start site identification problem with ADAPT-CAGE: a machine learning algorithm for the analysis of CAGE data. *Sci. Rep.*, **10**, 877.
40. Pennacchio,L.A., Bickmore,W., Dean,A., Nobrega,M.A. and Bejerano,G. (2013) Enhancers: five essential questions. *Nature reviews*, **14**, 288–295.
41. Andersson,R., Gebhard,C., Miguel-Escalada,I., Hoof,I., Bornholdt,J., Boyd,M., Chen,Y., Zhao,X., Schmidl,C., Suzuki,T. *et al.* (2014) An atlas of active enhancers across human cell types and tissues. *Nature*, **507**, 455–461.
42. Love,M.I., Huber,W. and Anders,S. (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.*, **15**, 550.
43. Nowicka,M. and Robinson,M. (2016) DRIMSeq: a Dirichlet-multinomial framework for multivariate count outcomes in genomics. *F1000Res*, **5**, 1356.
44. Anders,S., Reyes,A. and Huber,W. (2012) Detecting differential usage of exons from RNA-seq data. *Genome Res.*, **22**, 2008–2017.
45. Robinson,M.D., McCarthy,D.J. and Smyth,G.K. (2010) edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**, 139–140.
46. Bertin,N., Mendez,M., Hasegawa,A., Lizio,M., Abugessaisa,I., Severin,J., Sakai-Ohno,M., Lassmann,T., Kasukawa,T., Kawaji,H. *et al.* (2017) Linking FANTOM5 CAGE peaks to annotations with CAGEscan. *Sci Data*, **4**, 170147.
47. Lizio,M., Abugessaisa,I., Noguchi,S., Kondo,A., Hasegawa,A., Hon,C.C., de Hoon,M., Severin,J., Oki,S., Hayashizaki,Y. *et al.* (2019) Update of the FANTOM web resource: expansion to provide additional transcriptome atlases. *Nucleic Acids Res.*, **47**, D752–D758.
48. Carninci,P., Kasukawa,T., Katayama,S., Gough,J., Frith,M.C., Maeda,N., Oyama,R., Ravasi,T., Lenhard,B., Wells,C. *et al.* (2005) The transcriptional landscape of the mammalian genome. *Science*, **309**, 1559–1563.