**Supplementary Methods**

**1. Short-read mapping and annotation**

The clean reads were aligned to the *Arabidopsis* reference genome (Araport 11, https://www.arabidopsis.org/) and the rice reference genome (Oryza\_sativa.IRGSP-1.0, http://plants.ensembl.org/Oryza\_sativa/Info/Index) with HISAT2 v2.2.1 (Kim et al., 2019). Bioconductor packages (GenomicRanges v1.32.6, GenomicAlignments v1.16.0, and Biotrings v2.48.0) were used to analyze the alignment bam files. SplicingTypesAnno version 1.0.2 (Sun et al., 2015) took annotation files (GTF or GFF) as input and provided detailed intron information, including the start position, end position, strand information, intron number, and gene name, among other characteristics. Read locations were classified into three possible categories based on the reference genome: exon, intron, and intergenic region. R packages (ggplot2 v3.3.5 and lattice v0.20-38) were used for visualization. The alignment files (bam file) were sorted with SAMTools v1.9 (Li et al., 2009), and indexed for further analysis.

**2. Differential expression analysis**

Differential expression analysis between samples collected at different times after R2 inoculation was performed by using the Bioconductor package edgeR v3.22.3 (Robinson et al., 2010). Differentially expressed intronic RNAs exhibiting a fold change > 2 and a Benjamini–Hochberg-adjusted *P* ≤0.05 (false discovery rate [FDR] corrected) were selected. The expression values were visualized by using the R package ggplot2 v3.3.5 (Wickham et al., 2016).

**3. Long-read mapping and annotation**

Minimap2 v2.21 (Li, 2018) was used to align the reads to the *Arabidopsis* reference genome

Araport 11, using the parameter “-ax splice.” SAMTools version 1.9 (Li et al., 2009) was used to sort and index the alignment file for further analysis. The mapped reads of each sample were then assembled into transcripts based on the reference-guided assembly strategy using StringTie v1.3.1 (Pertea et al., 2015) in long-read mode. The assembled transcripts were merged with StringTie to obtain a non-redundant unified set of transcripts. Finally, the set of transcripts was compared with Araport 11 gene annotations by using gffcompare v0.9.9c (Pertea et al., 2020).

**4. Comparison of the sequencing strategies**

We compared intron-capture RNA-seq, total RNA-seq, and mRNA-seq by using several approaches. (1) We compared the sequencing reproducibility with Pearson correlation coefficients, using the “cor” function in R to compute the coefficients, and “ggplot2” in R to visualize the results. (2) We compared the efficiency of intron capture. We used the Bioconductor packages GenomicRanges v1.32.6 and GenomicAlignments v1.16.0 and the R package SplicingTypesAnno v1.0.2 (Sun et al., 2015) to compare the fraction of the total number of reads that were mapped to coding regions, untranslated regions, exon, introns, and intergenic regions. Then, we measured the percent of reads actually in the target regions. In addition, we calculated coverage with alignment data as input files, and we set bedtools v2.28.0 with the following parameter: “bedtools coverage -hist.” (3) We compared the number of intronic RNAs and splice junctions. For intronic RNAs, we used the R packages mentioned above to count the reads in introns considering that the strand-specific reads must be located in the introns. For splice junctions, we used the “translateGTF” function to get intronic regions, and the “countOverlaps” function to count the reads as splice junctions.

**5. Intronic RNA classification**

We used the Bioconductor packages GenomicRanges v1.32.6 and GenomicAlignments v1.16.0 and the R package SplicingTypesAnno v1.0.2 to analyze the alignment file (bam file). To compare the sequencing reads to the genome annotation, we used the “countOverlap,” “findOverlap,” and “subsetByOverlap” functions from the GenomicRanges and GenomicAlignments packages. To extract the intron sequences from the genome annotation, we used the “translateGTF” function from the SplicingTypesAnno package. To extract the nucleotide sequence, we used “DNAStringSet” and “substring” from the Biostrings v2.48.0 package. To identify the introns with intronic RNAs, we first selected reads with the following criteria: (1) reads are inside introns, (2) reads are strand-specific as introns, and (3) reads with no junctions. To avoid ambiguity, we removed all reads that overlapped with both exons and introns. Then, we merged all the reads inside the introns with the “reduce” function. Finally, we visualized the results in Integrative Genomics Viewer (IGV) v2.11.9 to confirm the finding.

To remove intron retention to avoid ambiguity, we filtered the candidates with the following criteria: (a) > 90% read coverage in the intron; (b) found in at least one mRNA-seq samples; and (c) found in at least one nanopore sequencing samples.

**6. Intronic RNA characterization**

To calculate the GC content of introns, we extracted the candidate intron sequences with bedtools v2.28.0, and then split the intronic region into 10 sections. The GC content was calculated by counting G and C bases as the percent of total bases in each section. The mean and standard deviation were also calculated for all the related introns. For comparison, we extracted all the introns without intronic RNAs in the same gene. To avoid ambiguity, we only selected those introns with the following criteria: (1) introns with at least 10 reads and (2) introns with reads in at least 3 samples. Finally, we extracted all the intron sequences of *Arabidopsis* and rice and calculated the GC content.

To calculate the intron number, we used the “translateGTF” function from SplicingTypesAnno v1.0.2 to extract all the introns. Then, we ordered all the introns in the same gene with R v 3.5.1. For introns in the negative strand, the intron numbers were reversed. The intron location was determined by the intron number in all introns.

Splicing signal in *Arabidopsis* was determined by the following pipeline with Bioconductor package Biostrings v2.48.0. (1) The genome sequence of *Arabidopsis* was first parsed by using “readDNAStringSet.” (2) All the sequences at the intron boundary were extracted with the “subseq” function. For those sequences in the negative strand, the sequence was translated by using the “reverseComplement” function.

**7. Circular RNA analysis**

To detect circular RNAs, we used the “segemehl.x” function from segemehl v0.3.4 (Hoffmann et al., 2014) to align all sequencing reads to the reference genome (Araport 11) of *Arabidopsis* using the following parameters: “-S -t 2”. Then, we used samtools to sort and index the alignment file. Finally, we extracted all the circular RNAs by using the “haarz.x” function with default parameters. We also used CIRI2 v2.0.6 (Gao et al., 2018) with default parameters as the alternative detection algorithm to increase the prediction accuracy. First, we aligned the sequencing data to the reference genome (Araport 11) of *Arabidopsis* with BWA-MEM v0.7.17-r1188 (Li H. and Durbin R., 2009). The parameter was set as “-T 19”. Then, “CIRI2.pl”function was applied to extract all circular RNA candidates from the aligned SAM files. The circular RNAs detected from two algorithms were merged together for further analysis. To combine all the candidate circular RNAs from different samples, we kept circular RNAs found in at least 2 samples with at least 2 reads. Then, we annotated circular RNAs by using SplicingTypesAnno v1.0.2 and Bioconductor package: GenomicAlignments v1.16.0. In addition, we extracted all the sequences by using bedtools v2.28.0 (Quinlan et al., 2010) for the subsequent analysis.

To compare our results to known circular RNAs, we download the circular RNA data from three circular RNA data bases: AtCircDB (<http://deepbiology.cn/circRNA/>), PlantCircBase (<http://ibi.zju.edu.cn/plantcircbase/>), and PlantCircNet (<http://bis.zju.edu.cn/plantcircnet/index.php>).

**8. Splice junction and novel transcript detection**

We used the Bioconductor package GenomicRanges and IRanges and the R package

SplicingTypesAnno v1.0.2 (http://sourceforge.net/projects/splicingtypes/) to extract all junctions detected from the bam files generated in the previous step. The pipeline took the bam files as the input and produced an html report as an output, including the details of the novel and known splice junctions. To avoid PCR artifacts, we only kept those junctions in at least 2 samples that had > 10 reads for the true junctions with the parameter “minReadCounts” as 10. All the junctions were merged into one file with junction counts, sample names, chromosome number, junction start, junction end, and strand information. We imported the final output for the splice junction sites into R for further analysis. Novel transcripts in introns were determined if they were not annotated in the GTF/GFF and were located in the introns. Furthermore, we confirmed these novel transcripts by visualizing the read alignment in Integrative Genomics Viewer (IGV) v2.11.9.

**9. Time-series analysis of intronic RNA under biotic stress**

We examined the *Arabidopsis* leaf transcriptome at 0, 2, 6, and 12 hpi after pathogen infection (three replicates per time point). We used the Bioconductor package edgeR v2.0.2 to quantify the known transcripts and to compare the expression difference between 0 and 2 hpi, between 0 and 6 hpi, and between 0 and 12 hpi. The transcripts with a fold change > 2 and FDR-adjust *P* < 0.05 were considered significant and kept for further analysis. We further determined the overlap of intronic RNA expression profiles from the different time points by the location of intronic RNAs, and visualized the results by using the R package vennDiagram v1.7.1.

To cluster the differentially expressed intronic RNAs, we used the Bioconductor package TCseq v1.18.0 to analyze the time-series sequencing data. We created a time course table for clustering analysis, with the rows representing intronic RNAs, and the columns representing time points. The values were normalized read counts (rpm). In the analysis, we chose the “standardize” parameter to perform z-score transformation on the data to be clustered, which reduces the noise introduced by the difference in the absolute values. We utilized unsupervised clustering method—k-means clustering—to cluster these intronic RNAs, setting the “k” parameter as 6 groups.

**10. Conservation analysis of intronic RNA**

We downloaded the “Orthologous Groups” generated from OrthoMCL with default parameters at <http://rice.uga.edu/annotation_pseudo_apk.shtml>. Then, we extracted all those orthologous groups that contain both *Arabidopsis* and rice. We extracted the intron sequences of these orthologous groups for further analysis by using bedtools v2.28.0.

**11. Pathway analysis**

We utilized DAVID (Huang et al., 2007) to perform Gene Ontology analysis. The input gene list included targeted genes. Gene ID were mapped to DAVID pathway database in terms of “TAIR ID” as the gene identifiers. We selected “Gene List” as the list type. The Functional Annotation Chart showed the gene functional groups related to the gene list, and we treated FDR-adjusted *P* < 0.05 as the statistical significance cut-off for further exploration of gene groups.

**12. Motif analysis**

We first extracted all the candidate intron sequences by using bedtools v2.28.0 with the parameter “-name -s.” Then, we analyzed these sequences with XSTREME (https://meme-suite.org/meme/tools/xstreme) to perform comprehensive motif analysis. In this analysis, the intron sequences were the input sequences, and we selected the default parameters for the analysis. All locations with *P* ≤ 10-3 were deemed motif matches.

**13. Experimental validation of intronic RNAs**

a. Validation for types of intronic RNAs: total RNA was reverse transcribed into cDNAs using random hexamers. Appropriate upstream (Intron-F) and downstream (Intron-R) primers were designed to clone the full length of intronic RNAs and verified their authenticity. We then designed additional primers to distinguish the distinct types of these intronic RNAs. For type A, we designed a pair of primers localized on the immediate upstream and downstream exons of the intronic RNAs (exon-F and Exon-R). For type B, we designed a primer localized on the immediate upstream or downstream exons of the intronic RNAs (exon-F or Exon-R). For type C, we designed a pair of primers localized to the 5’ and 3’ end of the same intron (Intron-F and Intron-R). For transcripts from fusion genes, we designed a pair of primers crossing the upstream and downstream gene of intronic RNAs. DNA was retrieved from electrophoresis gel and sequenced. Sequence identify is confirmed by blast against NCBI database.

b. Validation for differentially expressed intronic RNAs: we used semi-qRT-PCR analysis to validate the differentially expressed intronic RNAs along time-series. The expression of intronic RNAs at 0 hpi and 12 hpi was compared by semi-fluorescent quantitative PCR. The same set of primers were used in both verification and expression assays.

c. Validation for conserved intronic RNAs: to verify the conservation between Arabidopsis and rice, all Arabidopsis intronic RNAs were aligned to rice transcriptome database. Primers were designed according to the conservative intron region of rice, and verified by PCR.

d. Validation for new splicing junctions: to verify the novel splicing patterns predicted in the intron region, the full length intronic RNAs were cloned. Bands recovered from the agarose gel were cloned and sequenced. Sequence identify is confirmed by blast against NCBI database.

e. Validation for circRNAs: we selected the chr2:17138312-17139117\_- region to verify the super-circular region existed in introns. We performed reverse transcription with specific reverse primers to each circRNAs (chr2:17138462-17139089\_-, chr2:17138323-17139107\_- and chr2:17138312-17139117\_-), followed by PCR amplification with divergent primers (Supplementary Table 11).