IsoFlex: An R-Based Toolkit for High-Resolution Profiling of IsomiR Dynamics

1 Introduction

MicroRNAs (miRNAs) are crucial regulators of gene expression in eukaryotes, orchestrating critical biological processes, including cell differentiation, organ development, and environmental responses, through sequence-specific targeting of mRNAs(Bartel 2009). Advances in high-throughput sequencing have revealed that miRNAs exist not as uniform molecules, but as complex repertoires of sequence variants called isomiRs. These isoforms arise through multiple mechanisms, such as terminal heterogeneity (nucleotide additions/deletions at the 3'/5' termini), RNA editing, and single nucleotide polymorphisms (SNPs)(Chiang et al. 2010; Nishikura 2010; Xue et al. 2012). IsomiRs exhibit regulatory activities comparable to those of canonical miRNAs, while their sequence variations can refine targeting specificity or alter subcellular localization, allowing for nuanced regulatory networks(Morin et al. 2008; Neilsen, Goodall, and Bracken 2012).

R has emerged as a cornerstone for bioinformatic analysis due to its statistical prowess and the ecosystem of specialized packages (e.g., Bioconductor). Despite this, no dedicated R package exists for isomiR expression analysis. To bridge this gap, we developed isoFlex, an R-based toolkit for high-efficiency isomiR identification, quantification, and functional annotation. Its core innovation lies in an enhanced seed-and-extend algorithm featuring region-specific thresholds (5' terminus, central region, and 3' terminus) for independent control of insertion/deletion (InDel) and SNP tolerances, significantly improving sensitivity for complex variants. Functionally, isoFlex integrates miRBase data loading, isomiR detection, transcripts per million (TPM) based quantification, DESeq2-powered differential analysis, and interactive visualization. The S4 class system (Isomir and IsomirDataSet) streamlines cross-sample data management, democratizing access for non-bioinformaticians. The isoFlex package addresses critical gaps in parametric flexibility, functional comprehensiveness, and user experience, offering robust support for dissecting isomiR regulatory networks.

2 Design and implementation

The isoFlex R package delivers an integrated workflow for comprehensive isomiR analysis, including data import, detection, and expression profiling. The isoFlex package was developed in R (version 4.5), with the core algorithms optimized using Rcpp for computational efficiency. Integrates Bioconductor packages that include Biostrings(sequence manipulation), DESeq2 (differential expression analysis) and msa (multiple sequence alignment). The data structures, workflow, and core functions are described below.

2.1 Data structures

To systematically manage the complex information generated during isomiR detection and analysis, we constructed two core data structures based on the R S4 class system: the Isomir class and the IsomirDataSet class. These structures are designed to store the features of individual isomiRs and integrate data sets across multiple samples, respectively.

Isomir class: An instance of this class represents an isomiR derived from a reference miRNA. Its data structure comprises the following key attributes:

- Reference information: Reference miRNA ID (mature_id), sequence (mature_seq), precursor ID (pre_id), and precursor sequence (pre_seq).
- Alignment features: Template sequence including flanking regions; sequencing read and its abundance; counts of insertions/deletions at the 5' and 3' ends; positions of mismatches (including the central region); and a CIGAR (Compact Idiosyncratic Gapped Alignment Report) string detailing the alignment.
- Identifier and alignment result: Unique isomiR ID and result of multiple sequence alignment (MSA) with reference miRNA.

IsomirDataSet class: This class integrates isomiR data from multiple samples. Its core components include:

- A sample metadata data frame that stores sample names, treatment groups, file paths, etc.
- Sample-level matrix: The rows represent isomiRs, the columns represent samples, and the values contain normalized TPM values.
- Group-level matrix: Aggregated by sample groups (e.g., tissue type or treatment condition), containing mean TPM values per group (rows: isomiRs, columns: groups).
- IsomiR clustering information: A list of isomiRs grouped by reference miRNA, which supports cluster analysis based on sequence similarity.

The design of these data structures balances data storage efficiency with analytical flexibility. The Isomir class integrates sequence variation features with expression-level information, providing the foundation for functional analysis of individual isomiRs. The IsomirDataSet class

facilitates cross-sample expression pattern comparison, differential analysis, and visualization through hierarchical data organization. This architecture establishes an efficient and scalable data management framework for subsequent functional exploration of the isomiR.

2.2 Workflow

This package established an integrated isomiR analysis pipeline that comprises the following core steps, designed to encompass the entire workflow, from data loading to functional annotation.

2.2.1 Step 1: Loading of miRNA information

The mature and precursor miRNA sequences for the specified species were retrieved from the miRNA database and consolidated into a structured data frame. Key fields included: mature miRNA identifier (mature_id), mature miRNA sequence (mature_seq), precursor ID (pre_id), precursor sequence (pre_seq), and start position of the mature miRNA within the precursor (mature_start). Using the user-defined seed sequence length (default: 16 nucleotides [nt]), the seed sequences (seed_seq) were extracted. The template sequences for alignment were constructed by extending flanking regions (default: 3 nt at both termini) around the reference miRNAs.

2.2.2 Step 2: Sample-specific isomiR detection

Following sequencing data pre-processing, FASTQ files were efficiently parsed using the Rcpp package. Small RNA sequencing reads were stored as hash keys to merge identical sequences and quantify abundances, reducing memory usage and accelerating computation. The read expression was normalized as TPM. The reads were aligned to reference miRNAs using an enhanced seed-and-extend algorithm (for details, see Methods), with user-customizable thresholds for InDels and SNPs (default: ≤ 3 InDels at each terminus; ≤ 1 SNP per region). Sequence variations were quantified using the Hamming distance, and alignment details were documented in CIGAR strings.

2.2.3 Step 3: Group-wise isomiR filtering

To improve analytical reliability, an isomiR was defined as 'expressed' if it met both criteria within a treatment group: TPM ≥ 5 (user-adjustable threshold) in at least two biological replicates and presence across all samples within the group. This step filtered false positives arising from low expression or technical noise.

2.2.4 Step 4: Expression matrix construction

Two expression matrices were generated based on the isomiR TPM values:

- Sample-level matrix: The rows represent isomiRs, the columns represent samples, and the elements denote TPM values.
- Group-level matrix: Aggregated by experimental conditions (e.g., tissue type or treatment group), with elements representing mean TPM values per group (rows: isomiRs; columns: groups).

2.2.5 Step 5: IsomiR clustering and data integration

The isomiRs were clustered by reference miRNA and aligned using multiple sequence alignment, generating CIGAR strings to annotate sequence variations. Each isomiR was assigned a unique identifier and grouped into reference-specific Isomir object lists. The final outputs (expression matrices, alignment details, and metadata) were integrated into an IsomirDataSet object to support downstream visualization, differential analysis, and functional annotation.

2.3 Algorithm

2.3.1 Seed sequence matching

Reference miRNAs were partitioned into three functional regions: 5' flanking, 3' flanking, and a central seed region (default: 16 nt). Sequencing reads were scanned 5' to 3' for potential matches to reference miRNA seed sequences. User-defined mismatch thresholds were enforced. To optimize efficiency:

- Seed sequences (16 nt) were encoded as 64-bit integers using 4-bit binary representation (A=0001, T=0010, C=0100, G=1000).
- Mismatches were detected via bitwise XOR operations between the read and reference seed regions. Matching bases yielded 0000; mismatched bases (e.g., A-T) yielded 0011. The total count of '1's in XOR results (equivalent to twice the number of mismatched bases) determined threshold compliance.

2.3.2 Extension alignment and variant detection

Upon seed match success, flanking regions underwent extension alignment:

- InDel detection: Insertions/deletions at 5' and 3' termini were quantified by comparing read and reference lengths.

- SNP detection: Hamming Distance measured base mismatches at aligned positions, filtered by user-defined regional SNP thresholds (5'/seed/3').

Reads satisfying all thresholds were classified as isomiRs.

This region-specific control of InDel/SNP tolerance substantially improved sensitivity for complex variants. Binary encoding and bitwise operations enhanced computational efficiency for large-scale datasets while maintaining accuracy.

2.4 Core functionality

2.4.1 IsomiR detection

The `detect_isomirs()` function performs isomiR identification. Essential parameters include:

- sample_info_file: Path to sample metadata file
- fq_dir: Directory containing FASTQ files
- mirnas: Reference miRNA object
- max_indel_5p/max_indel_3p: Maximum insertions/deletions at the 5'/3' terminus (default: 3)
- max_snp_5p/max_snp_3p: Maximum mismatches in 5'/3' flanking regions (default: 1)
- max_snp_seed: Maximum mismatches in seed/central region (default: 1)
- max_snp: Global maximum mismatches (default: 3)
- min_tpm: Expression threshold for filtering (default: 1)

This function returns an IsomirDataSet object.

2.4.2 IsomirDataSet methods

All operations for querying, analyzing (e.g., expression profiling, differential expression), visualizing, and exporting isomiR results are implemented through methods associated with the IsomirDataSet class. The primary methods are cataloged in Table 1.

Table 1: Key methods for IsomirDataSet object

Function	Description
get_isoform_num(x)	Count isoforms per reference miRNA
get_isomir_by_ref(x, ref)	Extract isomiR data for a specific reference miRNA
get_alignment_by_ref(x, ref)	Extract aligned isoforms to a reference sequence
calc_group_expr_num(x)	Calculate the number of expressed isomiRs

Function	Description
get_ref_expr(x)	Extract expression data for reference miRNAs
get_expr_by_ref(x, ref)	Extract expression data for isomiRs of a specific reference
plot_expr(x, ref, tissue)	Plot the expression profile of isomiRs for a reference miRNA
calc_tsi(x)	Calculate tissue specificity index (TSI) for isoforms and reference miRNAs
get_deg_data(x, treatment, control)	Extract differential expression data from IsomirDataSet

x: An IsomirDataSet object, ref: Reference miRNA name, tissue: Tissue groups to include in the plot, treatment: Treatment group names, control: Control group name.

3 Installation

To install the `isoFlex` R package from GitHub, follow the instructions below. This package requires R version 4.5 or higher and system dependencies for compiling C++ code (via Rcpp).

3.1 Prerequisites

1. Install R (4.5):
   Download from https://cran.r-project.org.
2. Install system tools for compilation:

- Windows: Install Rtools.
- macOS: Install Xcode from the App Store.
- Linux: Install development tools (e.g., build-essential for Ubuntu/Debian).

3. Install devtools in R (if not installed): `install.packages("devtools")`

3.2 Installation command

Run the following code in R/RStudio to install `isoFlex` and its dependencies:

```r
# Bioconductor packages
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install(version = "3.21")
BiocManager::install(c("apeglm", "Biostrings", "DESeq2", "msa"))
```

```r
devtools::install_github(
  repo = "caibinperl/isoFlex",
  dependencies = TRUE,
  upgrade = "always"
)
```

4 A case study

```r
library(isoFlex)
library(tidyverse)
```

To further demonstrate the functionality and applicability of isoFlex, we employed this package to construct and analyze an isomiR expression atlas in grapevine.

4.1 Construction of a genome-wide isomiR expression atlas in grapevine

To delineate dynamic isomiR expression patterns during grapevine inflorescence development, we constructed a genome-wide isomiR expression atlas using publicly available small RNA sequencing data from the GEO database (accession: GSE59802).

We focuse on 8 samples representing 4 inflorescence developmental stages with two biological replicates per stage (Table 2).

Table 2: Sample list of grapevine used for small RNA libraries (GSE59802) downloaded from GEO database

sample name	group	tissue	developmental_stage
SRR1528372	Inflorescence_Y	Inflorescences	Young
SRR1528373	Inflorescence_Y	Inflorescences	Young
SRR1528374	Inflorescence_WD	Inflorescences	
SRR1528375	Inflorescence_WD	Inflorescences	
SRR1528376	Flower_FB	flowers	Beginning of flowering

sample name	group	tissue	developmental_stage
SRR1528377	Flower_FB	flowers	Beginning of flowering
SRR1528378	Flower_F	flowers	Flowering
SRR1528379	Flower_F	flowers	Flowering

4.1.1 Data acquisition and preprocessing

The data and metadata files are organized and stored in a specified/designated folder.

```
GSE59802/
  fastqs
      SRR1528372.fastq.gz
      SRR1528373.fastq.gz
      SRR1528374.fastq.gz
      SRR1528375.fastq.gz
      SRR1528376.fastq.gz
      SRR1528377.fastq.gz
      SRR1528378.fastq.gz
      SRR1528379.fastq.gz
  sample_info.csv
  raw_fastqs
      SRR1528372.fastq
      SRR1528373.fastq
      SRR1528374.fastq
      SRR1528375.fastq
      SRR1528376.fastq
      SRR1528377.fastq
      SRR1528378.fastq
      SRR1528379.fastq
```

Download and convert SRA files to FASTQ format using fasterq-dump (NCBI SRA Toolkit v3.2.0). For example, download a sample 'SRR1528372' by running:

```
fasterq-dump --split-3 SRR1528372 -O raw_fastqs
```

Quality control via fastp (v0.24.0), involving adapter trimming, Phred score filtering (≥ 20), and size selection (18–26 nt).

```
fastp -i raw_fastqs/SRR1528372.fastq -o fastqs/SRR1528372.fastq.gz
  --adapter_sequence auto
  --qualified_quality_phred 20
  --length_required 18
  --max_length 26
```

4.1.2 Input file specifications

4.1.2.1 Sample metadata file (*sample_info.csv*)

The metadata file must include the following columns:

- Column 1 (name): Sample names
- Column 2 (fq): Filenames of the sequencing data
- Column 3 (group): Experimental groups

sample_info.csv:

```
name,fq,group
SRR1528372,SRR1528372.fastq.gz,Inflorescence_Y
SRR1528373,SRR1528373.fastq.gz,Inflorescence_Y
SRR1528374,SRR1528374.fastq.gz,Inflorescence_WD
SRR1528375,SRR1528375.fastq.gz,Inflorescence_WD
SRR1528376,SRR1528376.fastq.gz,Flower_FB
SRR1528377,SRR1528377.fastq.gz,Flower_FB
SRR1528378,SRR1528378.fastq.gz,Flower_F
SRR1528379,SRR1528379.fastq.gz,Flower_F
```

4.1.2.2 Data directory (*fastqs*)

Sequencing data files (FASTQ or compressed formats) must be stored in a user-specified directory.

4.1.2.3 Reference miRNA database

miRNA information is prepared and stored in a miRNA file that contains the following columns.

- spe: The species name
- pre_id: The ID of the precursor miRNA
- pre_seq: The sequence of the precursor miRNA
- mirna_id: The ID of the mature miRNA

- mirna_seq: The sequence of the mature miRNA
- mirna_start: The start position of the mature miRNA in the precursor sequence
- mirna_end: The end position of the mature miRNA in the precursor sequence

The package has provided a miRNA file retrived from miRBase (v22):

```
mibase_file <- system.file("extdata", "miRBase.csv", package = "isoFlex")

read_csv(mibase_file)
```

```
Rows: 52949 Columns: 7
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (5): spe, pre_id, pre_seq, mature_id, mature_seq
dbl (2): mature_start, mature_end

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.

# A tibble: 52,949 x 7
   spe   pre_id     pre_seq         mature_id mature_seq mature_start mature_end
   <chr> <chr>      <chr>           <chr>     <chr>             <dbl>      <dbl>
 1 cel   cel-let-7  TACACTGTGGATCC~ cel-let-~ TGAGGTAGT~           17         38
 2 cel   cel-let-7  TACACTGTGGATCC~ cel-let-~ CTATGCAAT~           60         81
 3 cel   cel-lin-4  ATGCTTCCGGCCTG~ cel-lin-~ TCCCTGAGA~           16         36
 4 cel   cel-lin-4  ATGCTTCCGGCCTG~ cel-lin-~ ACACCTGGG~           55         76
 5 cel   cel-mir-1  AAAGTGACCGTACC~ cel-miR-~ CATACTTCC~           21         42
 6 cel   cel-mir-1  AAAGTGACCGTACC~ cel-miR-~ TGGAATGTA~           61         81
 7 cel   cel-mir-2  TAAACAGTATACAG~ cel-miR-~ CATCAAAGC~           20         41
 8 cel   cel-mir-2  TAAACAGTATACAG~ cel-miR-~ TATCACAGC~           61         83
 9 cel   cel-mir-34 CGGACAATGCTCGA~ cel-miR-~ AGGCAGTGT~           16         37
10 cel   cel-mir-34 CGGACAATGCTCGA~ cel-miR-~ ACGGCTACC~           53         74
# i 52,939 more rows
```

Users can also provide their custom miRNA files that contain the necessary columns as above.

4.1.3 Load microRNA data

Reference miRNA sequences for grapevine (miRBase v22.1; identifier 'vvi') were recovered by the `load_mirnas()` function, producing 186 mature miRNAs with associated precursor sequences, seed/center regions, and flanking templates.

```
vvi_mirnas <- load_mirnas(mirna_file = mibase_file, spe = "vvi")

head(vvi_mirnas)
```

```
    mature_id            mature_seq mature_start mature_end         seed_seq
1 vvi-miR156a  TGACAGAAGAGAGGGAGCAC           11         30 ACAGAAGAGAGGGAGC
2 vvi-miR156b  TGACAGAAGAGAGTGAGCAC           11         30 ACAGAAGAGAGTGAGC
3 vvi-miR156c  TGACAGAAGAGAGTGAGCAC           11         30 ACAGAAGAGAGTGAGC
4 vvi-miR156d  TGACAGAAGAGAGTGAGCAC           11         30 ACAGAAGAGAGTGAGC
5 vvi-miR156e  TGACAGAGGAGAGTGAGCAC           11         30 ACAGAGGAGAGTGAGC
6 vvi-miR156f TTGACAGAAGATAGAGAGCAC           11         31 GACAGAAGATAGAGAG
       pre_id                template_seq flank_5p_seq flank_3p_seq
1 vvi-MIR156a  TGTTGACAGAAGAGAGGGAGCACAAC          TGT          AAC
2 vvi-MIR156b  AACTGACAGAAGAGAGTGAGCACACA          AAC          ACA
3 vvi-MIR156c  AGGTGACAGAAGAGAGTGAGCACACA          AGG          ACA
4 vvi-MIR156d  GACTGACAGAAGAGAGTGAGCACATG          GAC          ATG
5 vvi-MIR156e  AGGTGACAGAGGAGAGTGAGCACTCA          AGG          TCA
6 vvi-MIR156f CTGTTGACAGAAGATAGAGAGCACAAC          CTG          AAC
```

4.1.4 Detect isomiRs

```
isomir_dataset <- detect_isomirs("GSE59802/sample_info.csv", "GSE59802/fastqs",
  vvi_mirnas, min_tpm = 5)
save(isomir_dataset, file = "GSE59802/isomir_dataset.RData")
```

IsomiR detection across all 8 samples was performed using `detect_isomirs()` with thresholds set to: expression ≥ 5 TPM, ≤ 3 InDels per terminus, and ≤ 1 SNP in 5'/3' flanking or central regions. The resultant IsomirDataSet object incorporated sample- and group-level expression matrices and MSA data.

This analysis identified 319 isomiRs that exhibited significant spatiotemporal expression patterns, establishing a genome-wide expression matrix.

```
head(isomir_dataset@group_expr)
```

```
                     Flower_F Flower_FB Inflorescence_WD Inflorescence_Y
AAGCTCAGGAGGGATAGCGCC     84.0     190.0             43.5              59
AAGCTCAGGAGGGATAGCGCCA     0.0       6.5              0.0               0
AAGCTCAGGAGGGATAGCGCCG     8.5      29.5              0.0               0
```

```
AAGCTCAGGAGGGATAGCGCCT    124.0      92.5             0.0              0
AATGGATGGTTAGGAGAG          0.0       0.0             0.0             11
AATGGATGGTTAGGAGAGA         0.0       0.0             0.0             14
```

4.1.5 Count isoforms per reference miRNA

```
get_isoform_num(isomir_dataset)
```

```
    vvi-miR156b     vvi-miR156c     vvi-miR156d     vvi-miR156f     vvi-miR156g
              1               1               1               3               3
    vvi-miR156i     vvi-miR159a     vvi-miR159b     vvi-miR159c      vvi-miR162
              3               4               4              10              11
    vvi-miR166a     vvi-miR166b     vvi-miR166c     vvi-miR166d     vvi-miR166e
             34              10              62              63              62
    vvi-miR166f     vvi-miR166g     vvi-miR166h     vvi-miR167a     vvi-miR167b
             65              65              67               1               2
    vvi-miR167c     vvi-miR167d     vvi-miR167e      vvi-miR168     vvi-miR171g
              1               2               2               3               1
    vvi-miR172c     vvi-miR172d  vvi-miR2950-5p     vvi-miR319b     vvi-miR319c
              2               1               5              23              23
    vvi-miR319e     vvi-miR319f     vvi-miR319g  vvi-miR3623-3p  vvi-miR3623-5p
              4              23               5               8               1
 vvi-miR3626-5p vvi-miR3633a-5p vvi-miR3633b-3p vvi-miR3633b-5p  vvi-miR3634-3p
              2               8               2               1              44
 vvi-miR3636-3p  vvi-miR3639-5p      vvi-miR390     vvi-miR393a     vvi-miR393b
              3               2               5               7               7
    vvi-miR394b     vvi-miR396a     vvi-miR396b     vvi-miR396c     vvi-miR396d
              1               9              12               9               9
    vvi-miR403a     vvi-miR403b     vvi-miR403c     vvi-miR403d     vvi-miR403e
              6               6               6               6               6
    vvi-miR403f      vvi-miR482
              6               1
```

This method calculates the number of isoforms associated with each reference miRNA in an IsomirDataSet object. It returns a vector of counts for reference miRNAs.

4.2 Spatiotemporal expression dynamics of grapevine isomiRs

4.2.1 Calculate the number of expressed isoforms per group

The `calc_group_expr_num()` function quantified the stage-specific isomiR abundances.

```
expr_num <- calc_group_expr_num(isomir_dataset)

head(expr_num)
```

```
             group total organ_specific common
1         Flower_F   108              5     61
2        Flower_FB   180             24     61
3 Inflorescence_WD   208             15     61
4  Inflorescence_Y   181             21     61
```

This method calculates the number of expressed isomiRs for each group in an IsomirDataSet object. It categorizes isomiRs into three types:

- Total: The total number of isomiRs expressed in each group
- Organ-Specific: The number of isomiRs expressed in one group
- Common: The number of isomiRs expressed in all groups

Plot the total number of expressed isoforms per group:

```
expr_num <- calc_group_expr_num(isomir_dataset)
ggplot(expr_num, aes(x = group, y = total)) +
  geom_bar(stat = "identity") +
  xlab("") + ylab("Number of isomiRs") +
  labs(title = "Number of isomiRs at each developmental stage") +
  theme(text = element_text(size = 12), plot.title = element_text(hjust = 0.5))
```

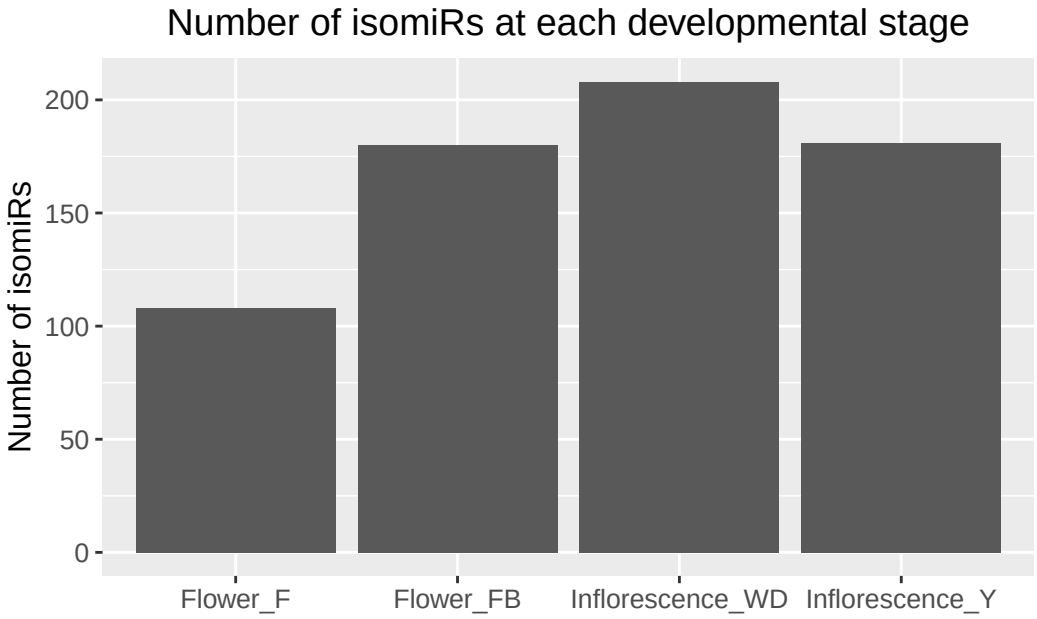

Figure 1: Number of isomiRs at each developmental stage

The expression of isomiR demonstrated pronounced heterogeneity in developmental stages. Well-developed inflorescences (Inflorescence_WD) exhibited the highest isomiR diversity (208 isomiRs), potentially linked to complex transcriptional networks during inflorescence differentiation (Figure 1). In contrast, flowering stage (Flower_F) showed minimal isomiR activity (108 isomiRs), suggesting specialized regulatory roles within discrete developmental windows.

Plot the number of isoforms that are specifically expressed in each experimental group:

```
ggplot(expr_num, aes(x = group, y = organ_specific)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  xlab("") + ylab("Number of isomiRs") +
  labs(title = "Distribution of organ-specific isomiRs") +
  theme(text = element_text(size = 12), plot.title = element_text(hjust = 0.5))
```

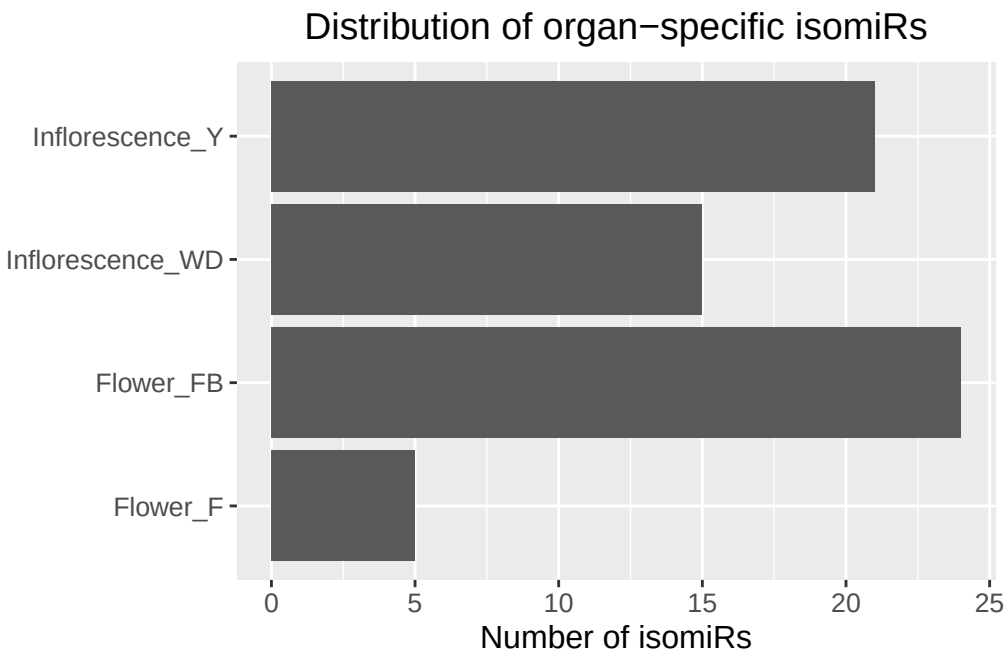

Figure 2: Distribution of organ-specific isomiRs

We identified 65 organ-specific isomiRs expressed exclusively in single stages (Figure 2).

4.2.2 Tissue specificity index

Gene expression specificity across tissues or conditions was quantified using the Tissue Specificity Index (TSI, τ-value), where TSI $= 0$ indicates ubiquitous expression and TSI $= 1$ denotes strict tissue-specificity. The TSI (τ-index) was calculated as:

$$\tau = \frac{\sum_{i=1}^{n} (1 - x_i)}{n - 1}$$

where N is the number of tissues/conditions and x_i represents the normalized expression in tissue/condition i. High TSI values imply tissue-restricted biological roles, while low values indicate broad functional relevance.

```
tsi_df <- calc_tsi(isomir_dataset)

head(tsi_df)
```

```
                read_seq   type  tsi
1  AAGCTCAGGAGGGATAGCGCC  miRNA 0.67
2 AAGCTCAGGAGGGATAGCGCCA isomiR 1.00
3 AAGCTCAGGAGGGATAGCGCCG isomiR 0.90
4 AAGCTCAGGAGGGATAGCGCCT isomiR 0.75
5     AATGGATGGTTAGGAGAG isomiR 1.00
6    AATGGATGGTTAGGAGAGA isomiR 1.00
```

Generate a violin plot with overlaid boxplots to visualize the distribution of TSI values for reference miRNAs and isoforms.

```
ggplot(tsi_df, aes(x = type, y = tsi)) + geom_violin() +
  geom_boxplot(width = 0.05, fill = "black", outlier.color = NA) +
  stat_summary(fun = median, geom = "point", fill = "white", shape = 21,
    size = 2.5) +
  xlab("isomiR type") + ylab("TSI") +
  theme(text = element_text(size = 12))
```

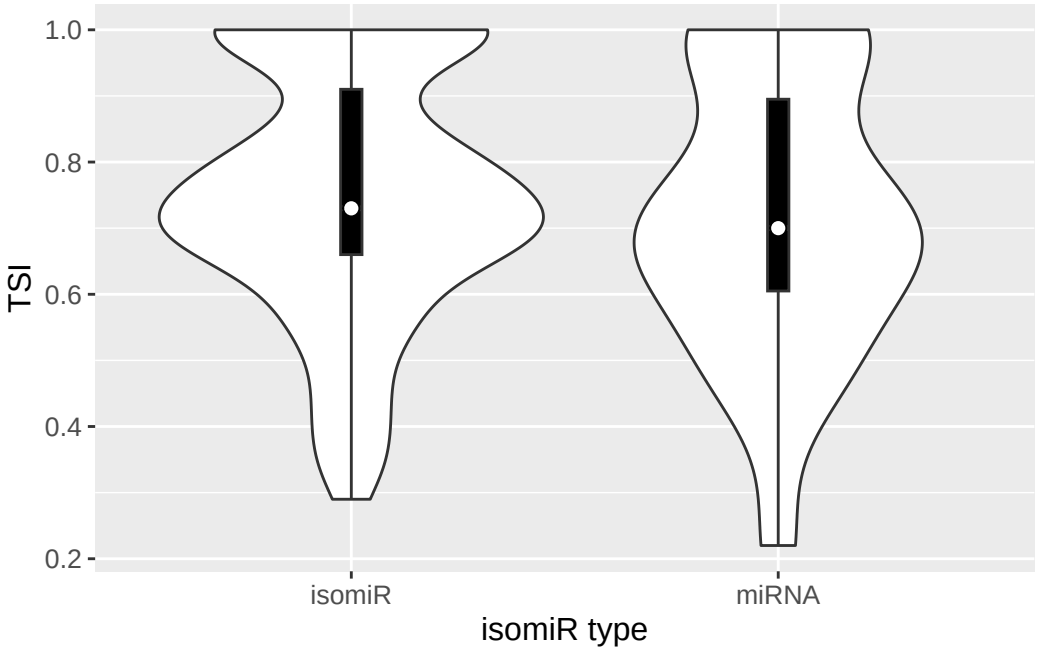

Figure 3: Comparison of tissue specificity index distributions between canonical miRNAs and isomiRs.

```
stats::ks.test(tsi_df$tsi[tsi_df$type == "isomiR"],
  tsi_df$tsi[tsi_df$type == "miRNA"])
```

```
Warning in ks.test.default(tsi_df$tsi[tsi_df$type == "isomiR"],
tsi_df$tsi[tsi_df$type == :    P-


    Asymptotic two-sample Kolmogorov-Smirnov test

data:  tsi_df$tsi[tsi_df$type == "isomiR"] and tsi_df$tsi[tsi_df$type == "miRNA"]
D = 0.15925, p-value = 0.302
alternative hypothesis: two-sided
```

The isomiR τ-values were not significantly skewed toward 1 (Figure 3; Kolmogorov-Smirnov test, P = 0.302) comparable to those of canonical miRNAs.

4.3 Expression dynamics of isomiRs in inflorescences

To elucidate the regulatory roles of isomiRs during grapevine reproductive development, we focused on the canonical miR156/miR172 regulatory circuit and their isoforms in inflorescence

4.3.1 miR156 and miR172

Extract isomiR data for vvi-miR156.

```
get_alignment_by_ref(isomir_dataset, "vvi-miR156f")
```

```
     Biostrings

DNAStringSet object of length 5:
    width seq                                           names
[1]    27 ---TTGACAGAAGATAGAGAGC-----                   19M2D
[2]    27 ---TTGACAGAAGATAGAGAGCACT--                   21M1I
[3]    27 ---TTGACAGAAGATAGAGAGCACC--                   21M1I.1
[4]    27 ---TTGACAGAAGATAGAGAGCAC---                   vvi-miR156f
[5]    27 CTGTTGACAGAAGATAGAGAGCACAAC                   template
```

This method extract the alignment of the sequences of isomiRs and their reference template sequence using multiple sequence alignment. It is useful for visualizing sequence variations and identifying potential editing events in isomiRs.

```
tissue <- c("Inflorescence_Y", "Inflorescence_WD", "Flower_FB", "Flower_F")
plot_expr(isomir_dataset, "vvi-miR156f", tissue)
```

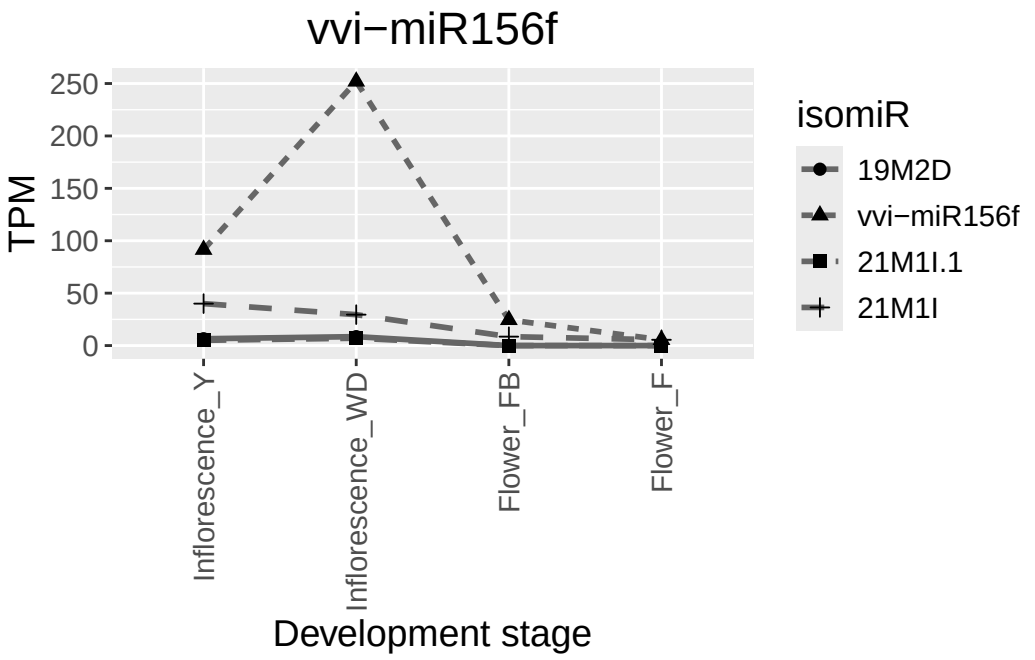

Figure 4: Number of miR156 isomiRs expressed in grapevine samples across inflorescences

```
expr <- get_expr_by_ref(isomir_dataset, "vvi-miR156f")
ref_expr <- expr["vvi-miR156f", ]
iso_expr <- expr[-c(2), ]
ref_expr <- expr["vvi-miR156f", "Inflorescence_WD"]
iso_expr <- expr[-c(2), "Inflorescence_WD"]
(mean(iso_expr) - ref_expr) / ref_expr
```

```
[1] -0.9404762
```

For miR156f, the 3'-terminal isoforms (with base deletions/additions) mirrored the expression trend of canonical miR156f (Figure 4) but showed a markedly reduced abundance in mature inflorescences (94.05% lower TPM than the canonical sequence), implying auxiliary roles in developmental fine-tuning through dosage effects.

Extract isomiR data for vvi-miR172c.

```
get_alignment_by_ref(isomir_dataset, "vvi-miR172c")
```

```
DNAStringSet object of length 4:
    width seq                                                 names
[1]    27 ---GGAATCTTGATGATGCTG------                        18M3D
[2]    27 ---GGAATCTTGATGATGCTGC-----                        19M2D
[3]    27 ---GGAATCTTGATGATGCTGCAG---                        vvi-miR172c
[4]    27 GTGGGAATCTTGATGATGCTGCAGCGG                        template
```

```
plot_expr(isomir_dataset, "vvi-miR172c", tissue)
```

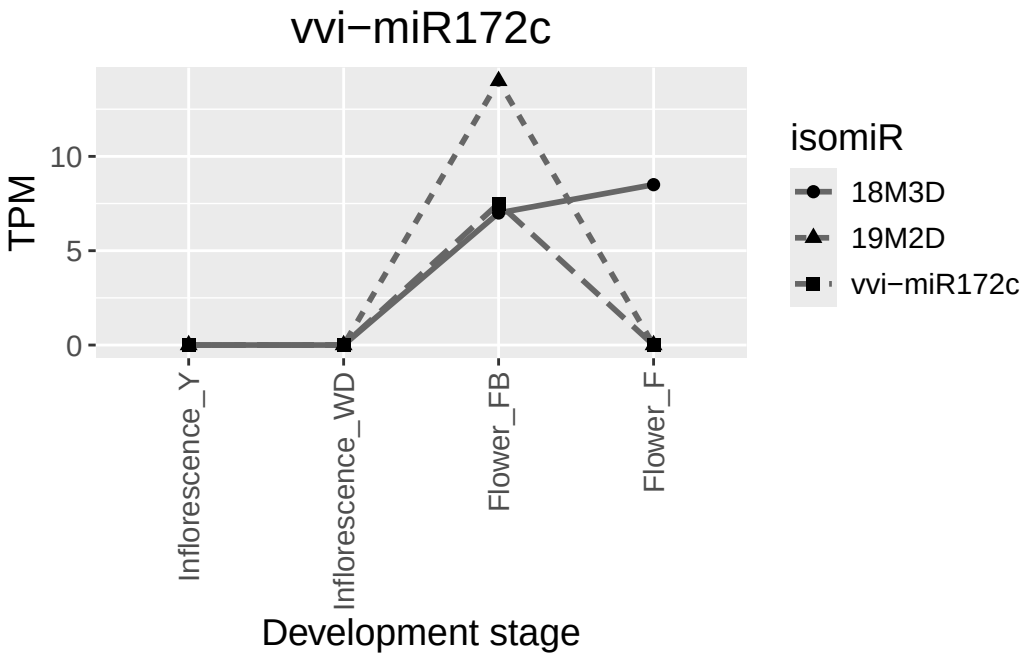

Figure 5: Number of miR172 isomiRs expressed in grapevine samples across inflorescences

Sequencing analysis revealed stage-specific expression patterns for miR172c 3'-terminal isoforms (e.g., 19M2D and 18M3D) and their canonical counterpart during inflorescence development (Figure 5). Although the canonical miR172c and the isoform 19M2D peaked in early flowering (Flower_FB), 18M3D exhibited maximal expression in flowering (Flower_F). This phased expression suggests temporal coordination of floral organ differentiation through stage-specific target regulation.

```
expr_156 <- get_expr_by_ref(isomir_dataset, "vvi-miR156f")
expr_172 <- get_expr_by_ref(isomir_dataset, "vvi-miR172c")
expr_156 <- apply(expr_156[, tissue], 2, mean)
expr_172 <- apply(expr_172[, tissue], 2, mean)

cor(as.vector(t(expr_156)), as.vector(t(expr_172)), method = "pearson")
```

```
[1] -0.6527722
```

In particular, the expression exhibited a strong negative correlation (Pearson's r = -0.65), consistent with the evolutionarily conserved miR156-miR172 cascade(Wu et al. 2009).

4.3.2 Perform differential expression analysis on isomiRs

`get_deg_data()` extracts expression data and sample information from an IsomirDataSet object for specified treatment and control groups, preparing it for differential expression analysis.

```
deg_data <- get_deg_data(isomir_dataset, treatment = "Flower_FB",
  control = "Inflorescence_WD")
```

```
deg_df <- calc_deg(deg_data)
head(deg_df)
```

```
                       baseMean log2FoldChange    lfcSE       pvalue
AAGCTCAGGAGGGATAGCGCCT 52.45515      -9.734054 2.980500 1.652628e-09
TTCCATCTCTTGCACACTGG   28.63745      -8.677853 2.877380 6.869422e-08
AAGCTCAGGAGGGATAGCGCCG 16.73975      -7.687420 2.805877 1.885081e-06
CTAAGCACAGCTCTCGCATCC  12.48853      -7.109911 2.780752 1.099824e-05
TCGATAAACCTCTGCATCCAGT 12.48853      -7.109911 2.780752 1.099824e-05
TCGATAAACCTCTGCATCCAT  11.92159      -7.015328 2.778623 1.447929e-05
                              padj
AAGCTCAGGAGGGATAGCGCCT 2.035181e-07
TTCCATCTCTTGCACACTGG   4.928810e-06
AAGCTCAGGAGGGATAGCGCCG 7.728830e-05
CTAAGCACAGCTCTCGCATCC  2.428073e-04
TCGATAAACCTCTGCATCCAGT 2.428073e-04
TCGATAAACCTCTGCATCCAT  2.968254e-04
```

4.4 Conclusions

In summary, isoFlex delivers an efficient, precise R toolkit to overcome key bottlenecks in isomiR analysis. Its innovations include:

- enhanced seed-and-extend detection of complex variants;
- integrated analytical modules with interactive visualization for workflow efficiency;
- flexible parameterization for broad applicability.

References

Bartel, David P. 2009. "MicroRNAs: Target Recognition and Regulatory Functions." *Cell* 136 (January): 215–33. https://doi.org/10.1016/j.cell.2009.01.002.

Chiang, H. Rosaria, Lori W. Schoenfeld, J. Graham Ruby, Vincent C. Auyeung, Noah Spies, Daehyun Baek, Wendy K. Johnston, et al. 2010. "Mammalian microRNAs: Experimental Evaluation of Novel and Previously Annotated Genes." *Genes & Development* 24 (May): 992–1009. https://doi.org/10.1101/gad.1884710.

Morin, Ryan D., Michael D. O'Connor, Malachi Griffith, Florian Kuchenbauer, Allen Delaney, Anna-Liisa Prabhu, Yongjun Zhao, et al. 2008. "Application of Massively Parallel Sequencing to microRNA Profiling and Discovery in Human Embryonic Stem Cells." *Genome Research* 18 (April): 610–21. https://doi.org/10.1101/gr.7179508.

Neilsen, Corine T., Gregory J. Goodall, and Cameron P. Bracken. 2012. "IsomiRs–the Overlooked Repertoire in the Dynamic microRNAome." *Trends in Genetics : TIG* 28 (November): 544–49. https://doi.org/10.1016/j.tig.2012.07.005.

Nishikura, Kazuko. 2010. "Functions and Regulation of RNA Editing by ADAR Deaminases." *Annual Review of Biochemistry* 79: 321–49. https://doi.org/10.1146/annurev-biochem-060208-105251.

Wu, Gang, Mee Yeon Park, Susan R. Conway, Jia-Wei Wang, Detlef Weigel, and R. Scott Poethig. 2009. "The Sequential Action of miR156 and miR172 Regulates Developmental Timing in Arabidopsis." *Cell* 138 (August): 750–59. https://doi.org/10.1016/j.cell.2009.06.031.

Xue, Zhihong, Haiyan Yuan, Jinhu Guo, and Yi Liu. 2012. "Reconstitution of an Argonaute-Dependent Small RNA Biogenesis Pathway Reveals a Handover Mechanism Involving the RNA Exosome and the Exonuclease QIP." *Molecular Cell* 46 (May): 299–310. https://doi.org/10.1016/j.molcel.2012.03.019.