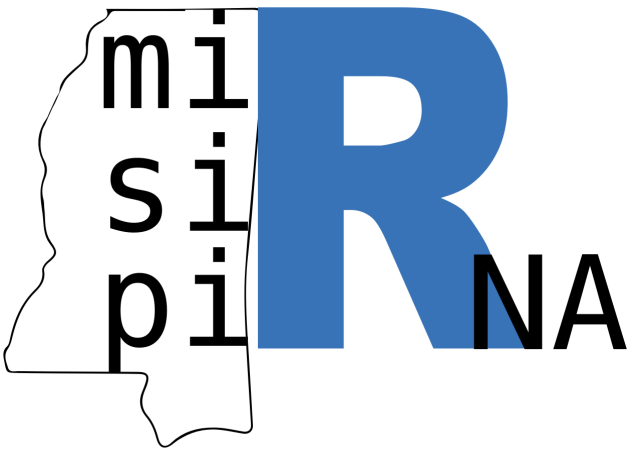
MiSiPi.RNA Documentation



Installation:

Install Devtools, MiSiPi.RNA packages and load library

```
install.packages("devtools")

devtools::install_github("stupornova33/MiSiPi.RNA")

library(MiSiPi.RNA)
```

MiSiPi.RNA is designed to run efficiently on personal laptops or high performance compute clusters. The current version supports Windows, Linux, and Mac operating systems.

Data preparation and basic usage

1. Align adapter- trimmed and quality- filtered small RNA reads to reference genome, convert to BAM format, and index.
2. If you are interested in characterizing novel or un-annotated species, you should create a BED file of regions of interest using a pipeline similar to the method shown in the Cluster Calling section of this document. Otherwise, reference annotations can be used.
3. The regions of interest should be formatted as a 3-column BED file (but if there are more columns, they will be ignored) and all lines must contain the same number of fields. Chromosome names in the BED file must match the chromosome names in the reference genome.
4. Install the ViennaRNA package on your system.

(Optional) The miRNA module outputs structure plots from RNAfold as .ps files. There is currently no built-in R way to convert these files to png or pdf. If you want to be able to convert them using MiSiPi.RNA, you can install the freely available packages Ghostscript and ImageMagick and use our included function ps2png.

5. (Optional) Prepare reference annotation BED file (useful for the hairpin RNA and/or cisNAT siRNA plots). The annotation plot is at a rough-draft stage, so we recommending extracting features (e.g. transcripts) which are most relevant to your interest and removing redundant features for clarity.

To see an example of how this file should be formatted, run:

```
annot = system.file("extdata", "processed_dmel.txt", package = "MiSiPi.RNA")
```

Once you have gathered all of these materials, you are ready to run your small RNA module/s of choice. As an example, if you are interested in miRNAs, you would run:

```
## Choose your Pokemon
## setvars just helps format and check the parameters you provide
## Descriptions of the different parameters are provided on the main Github page.

vars <- set_vars(
    roi = "path/to/bed",
    bam_file = "path/to/bam",
    genome = "path/to/genome",
    plot_output = TRUE,
    path_to_RNAfold = "path/to/ViennaRNA/RNAfold.exe",
    path_to_RNAplot = "path/to/ViennaRNA/RNAplot.exe",
    pi_pal = "BlYel",
    si_pal = "RdYlBl",
    annotate_region = TRUE,
    weight_reads = "none",
    gtf_file = "path/to/gtf",
    write_fastas = FALSE,
    out_type = "pdf"
)

misipi_rna(vars) - Default method is "all"

misipi_rna(vars, method = "siRNA")
```

Output will be stored in a directory called “miRNA_outputs” (or “siRNA_outputs” or “piRNA_outputs”).

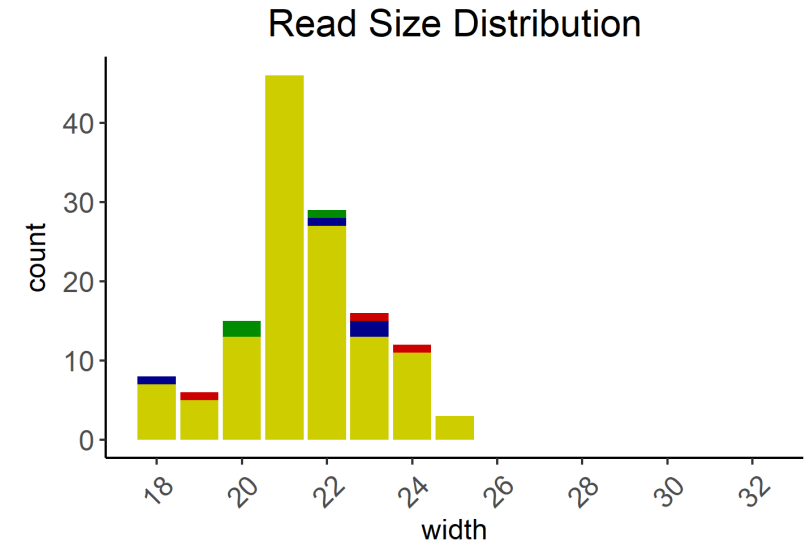
This module will produce an output directory called “run_all” in your working directory with subfolders for each type of small RNA.

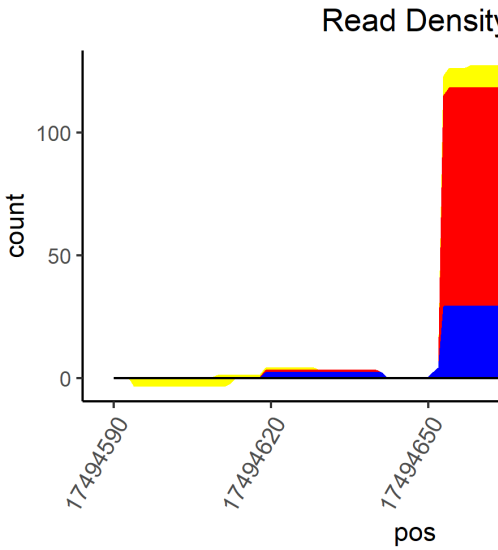
Output files from each module

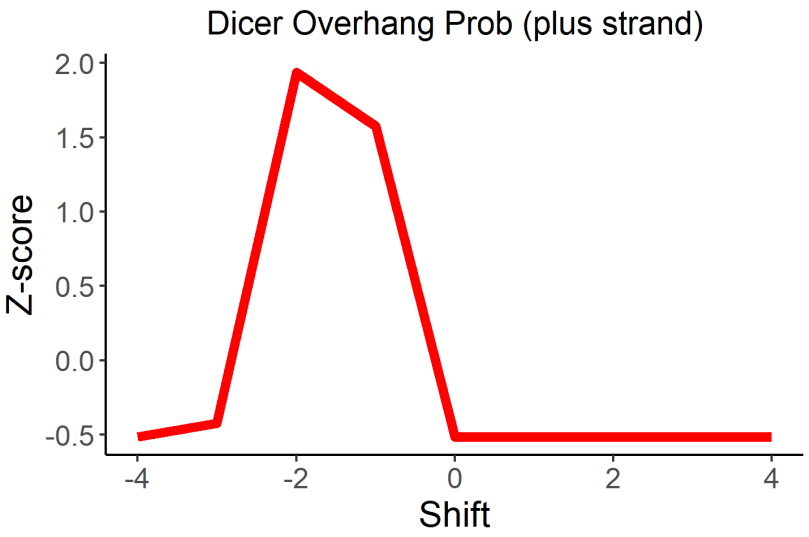
By default, a logfile is made for each module (e.g. “miRNA_logfile.txt”). This is primarily for troubleshooting issues. The program writes important messages into this file.

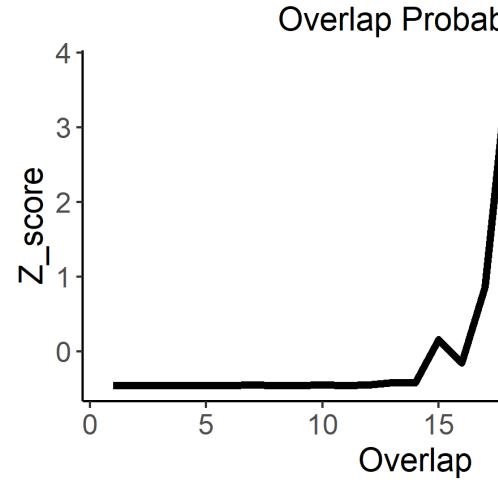
miRNA module

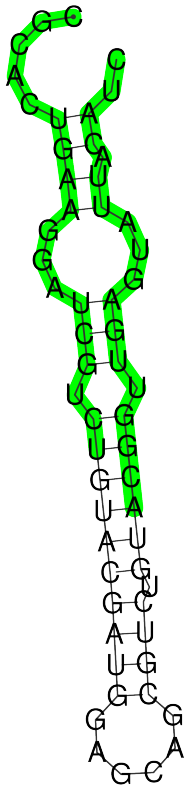
If plot_output is set to true, there will be a png or pdf file for each locus in the BED file you provided in the arguments, however it is important to note that if there are too few reads mapped for the program to progress in calculations and plotting, no plot will be made. Since this program is designed to be able to run without strand information, there will be a file ending with "+_combined" or "-_combined" for each locus (assuming adequate read coverage). This behavior may be changed in the future.









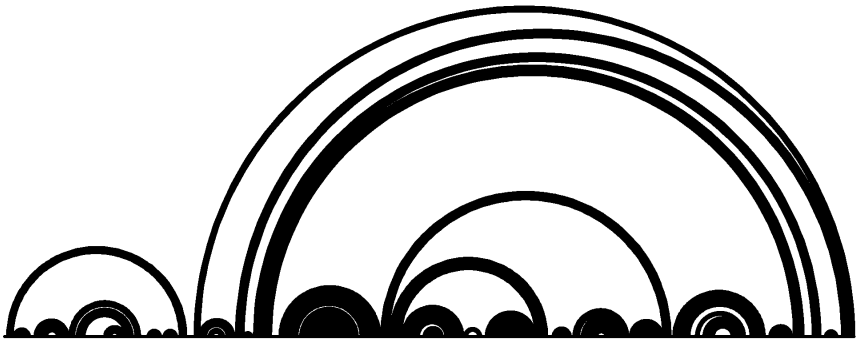


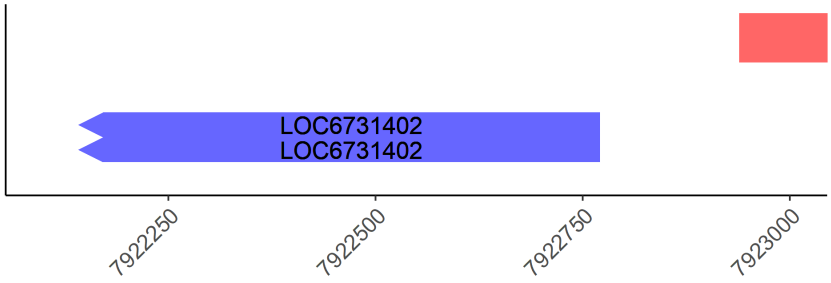
The miRNA module plots the read size distribution at the locus, including the nucleotide composition for each read size (top left), the read density by size at the locus (top middle) and the Dicer overhang probability (lower left) and read size overlap probability (lower middle). More information regarding these probabilities can be found in the bioRxiv preprint at https://www.biorxiv.org/content/10.1101/2023.05.07.539760v1. The program runs RNAfold to generate a predicted structure, and outputs a .ps file where the two most abundant reads are colored. (See Data Preparation section above regarding converting .ps files to .png in batch). In the image below, the most abundant read is colored green, while the second most abundant is red. If both reads have the same abundance, they will both be green.

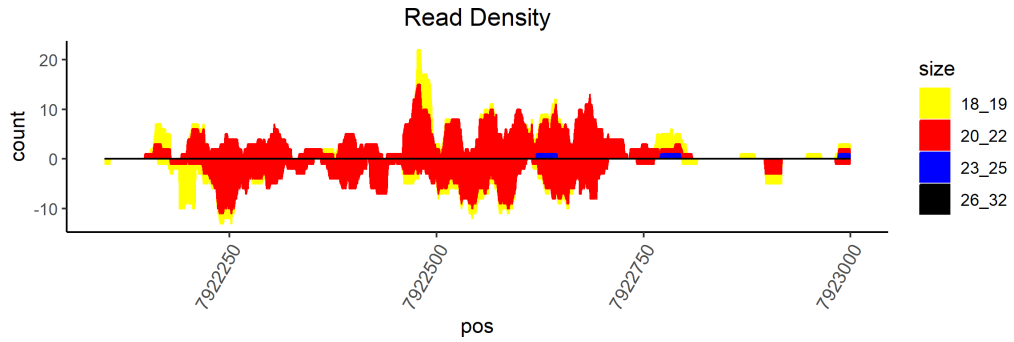
Other outputs of the miRNA module are "alt_miRNAs_coord.bed" and "miRNA_dicerz.txt". When the program encounters "isomirs" (slightly different variations in the miRNA), it chooses the most abundant variant for plotting and statistics and outputs the isomir coordinates as a BED file. If interested, you could use this file for further characterization by MiSiPi.RNA or other programs of your choosing. "miRNA_dicerz.txt" is a table containing the numbers used to generate the Dicer Probability plot above.

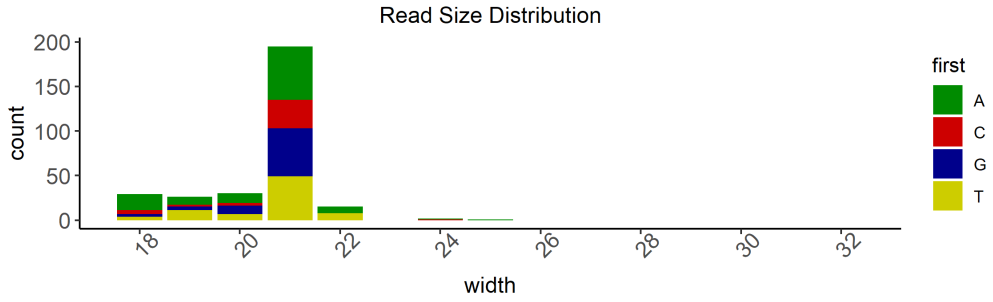
siRNA module

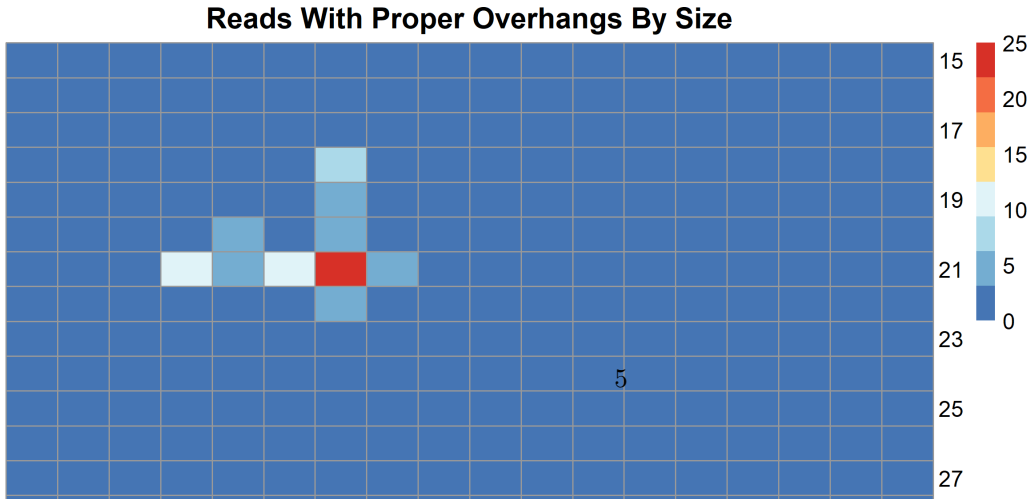
The two known primary types of endogenous siRNAs are the bidirectionally transcribed cis-natural antisense transcripts (cisNATs) and long hairpin RNAs formed from complementary inverted repeats. Both are processed from their double-stranded precursors into 21nt length small RNAs by an RNAse enzyme (in fruitflies, Dcr-2). The siRNA program first processes reads from both strands and performs various calculations, followed by hairpin-RNA specific processing (e.g. single-strand processing).

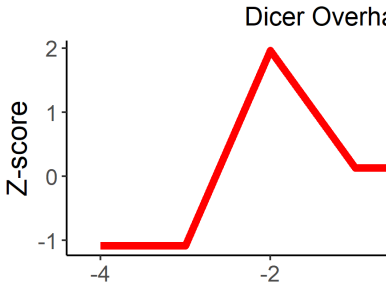


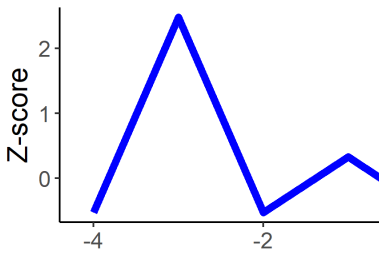

LOC6731402
LOC6731402
7922250
7922500
7922750
7923000

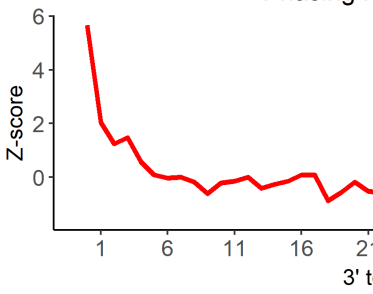

Read Density
count
pos
size
18_19
20_22
23_25
26_32

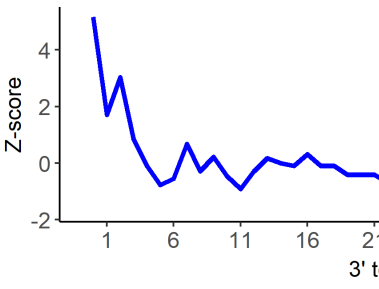

Read Size Distribution
count
width
first
A
C
G
T


Reads With Proper Overhangs By Size


Z-score


Z-score


Z-score


Z-score


Z-score

The hairpin RNA-specific part of the module outputs an "arc plot" from RNAfold (top left) which depicts the paired bases at a genomic locus as connected lines. Under the arc plot, known annotations are plotted as bars. Features on the antisense strand are colored blue, while sense features are red. Below the annotation plot, the read density is plotted by read size. Read size distribution with nucleotide composition of the first nucleotide is also shown below the read size distribution.

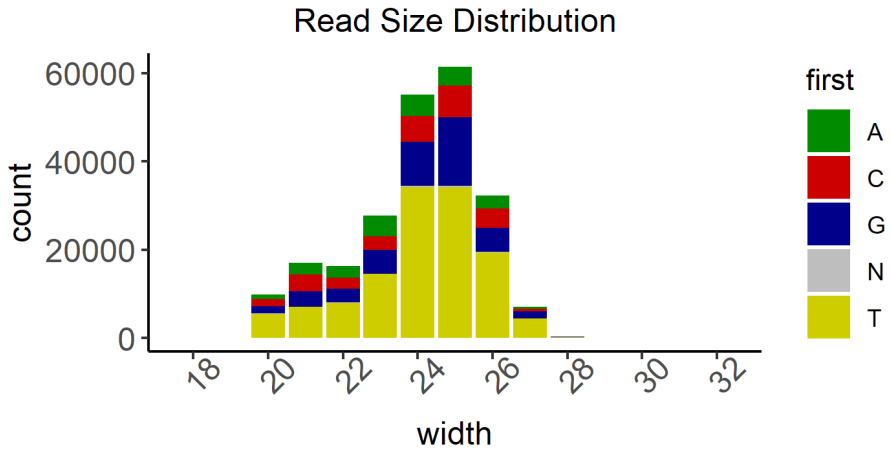
For cisNAT siRNAs, a heatmap is made. For this plot, the program identifies overlapping read pairs which have a characteristic "Dicer signature" (see bioRxiv article) from opposite strands and tabulates their sizes. This plot is useful for counting the number of cisNAT siRNA pairs at a locus. It is important to note that if the plot for a locus of interest does not have a heatmap, it is due to lack of reads on one strand.

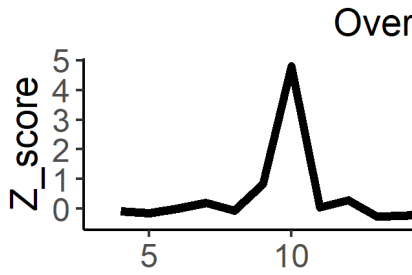
The other plots depicted are Z-scores for Dicer processing by sense (top right), antisense (middle top right) strands, phased processing by strand, and Dicer signature of sense/antisense overlapping reads (bottom right).

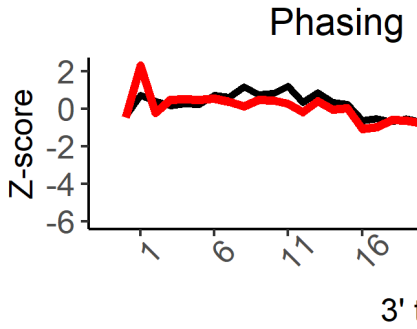
Other output files from the siRNA module are the values by locus of the Dicer and phasing probability plots for hairpin RNAs ("minus_hp_phasedz.txt", "plus_hp_phasedz.txt", "minus_hp_dicerz.txt", "plus_hp_dicerz.txt") and cisNAT siRNAs ("siRNA_dicerz.txt") as well as the raw count matrix for the siRNA heatmap, one line per locus ("siRNA_heatmap.txt"). Finally, "siRNA_read_size_distributions.txt" contains the read size distribution of each locus from 16nt-32nt.

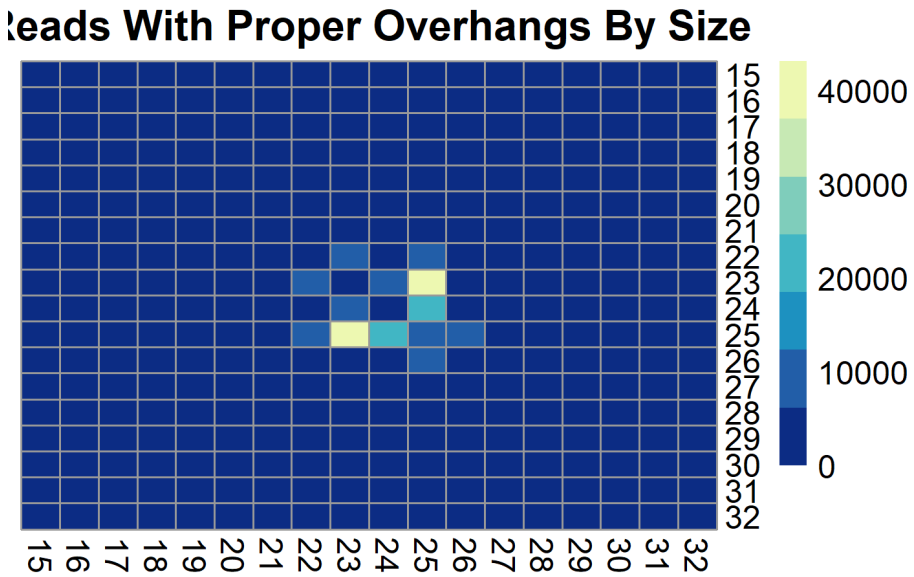
piRNA module

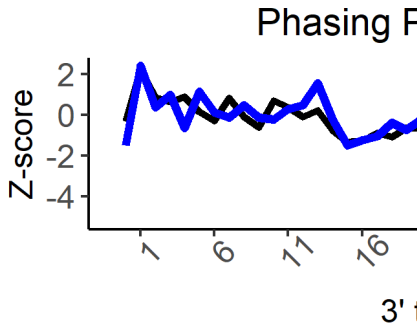
Piwi-interacting RNAs (piRNAs) are processed in two distinct pathways- the ping pong piRNA pathway and phased piRNA pathway. Ping pong piRNAs are generated by a positive feedback loop of sense/antisense complementarity, targeting, and cleavage, while phased piRNAs are processed in a phased manner from a long single-stranded precursor.

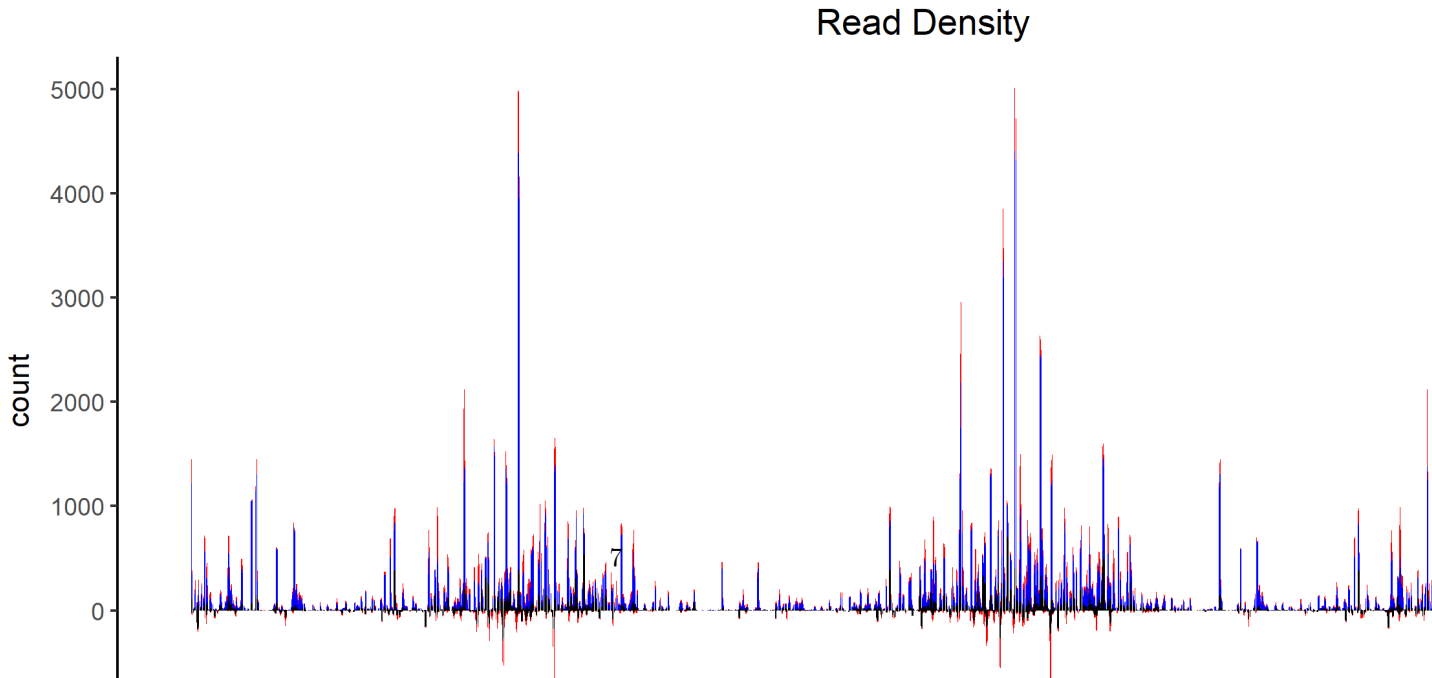

Read Size Distribution
count
60000
40000
20000
0
18
20
22
24
26
28
30
32
width
first
A
C
G
N
T


Over
Z_score
5
4
3
2
1
0
5
10


Phasing
Z-score
2
0
-2
-4
-6
1
6
11
16
3'


eads With Proper Overhangs By Size
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
40000
30000
20000
10000
0


Phasing F
Z-score
2
0
-2
-4
1
6
11
16
3'


Read Density
count
5000
4000
3000
2000
1000
0

The piRNA module plots the read size distribution with nucleotide composition (top left). The nucleotide composition is relevant for annotating piRNAs, as Uracil at the 5' end of a piRNA may be recognized by Argonaute/Piwi proteins. The module identifies sense/overlapping read pairs which overlap by exactly 10nt, a hallmark characteristic of ping pong piRNAs, and plots them by read size as a heatmap (left middle). Read density by size is shown in the bottom panel, and overlap probability, and strand-specific phasing probability is shown in the top right. For phasing, all read sizes are plotted as black lines, while reads 26nt+ are shown in either red or blue.

Other output files from the piRNA module are the phased z-scores from all reads ("all_phased_piRNA_zscores.txt"), plus and minus strand reads ("phased_plus_piRNA_zscores.txt" and "phased_minus_piRNA_zscores.txt"), phased z-scores from 26nt+ sized reads from sense and antisense, the read size distributions, the heatmap counts for each locus, and the overlap probability from 4nt-30nt.

misipi_rna

This module runs all of the other modules in addition to some other metrics and statistics and outputs them as a table.

The output table can be used if you are unsure of the nature of your small RNA loci, or if you wish to produce a summary HTML file with a sortable table and heatmap plots, or if you intend to run machine learning using the ml_probability function (see later machine learning section). It should be noted that for the summary HTML file, "plot_output" should be set to TRUE, and "output_type" should be "png".

```
# set vars as shown above

misipi_rna(vars, method = "piRNA")
```

The ML table

The misipi_rna module produces a table which is named according to the name of the input BED file, the name of the input BAM file, followed by "_ml.txt". This table can be used for characterization of your loci of interest, or as input for the machine learning module (ml_probability) tests the probability that a locus has the features of one of the major small RNA pathways.

A brief description of what is being calculated for the columns in the ML table:

1. Locus name (the chromosome, start, and end from the BED file).

2. Length The length of the locus.

3. log_shap_p Log10(Shapiro's p-value). After calculating the read size distribution, the normality of the distribution is tested using a Shapiro-Wilk's test. Small RNA-producing loci that are part of major pathways will often have skewed distributions relative to loci with reads generated from random degradation.

auc Similar to (3), the area under a curve fit on a normal distribution should be 1, whereas in skewed distributions, the curve will not capture 100% of the data.

strand_bias Strand bias. The ratio of reads aligning to sense or antisense. This can help distinguish small RNAs produced from a single stranded precursor versus a bidirectionally transcribed locus or from a foldback structure from an inverted repeat.

6. perc_GC The GC nucleotide composition of all the reads.

7. ave_size The average read size at the locus

8. perc_first_nucT The percentage of reads which have a uracil (T) at the first position in the 5' end, which is a preference for some small RNA processing enzymes.

9. perc_A10 Similar to (8) an adenine bias at the tenth position is a characteristic of ping pong piRNA biogenesis.

10. highest_si_col This number comes from the siRNA heatmap (See siRNA module). It is the average read size of the most commonly found overlapping read pairs which have a proper 2nt overhang, or in the heatmap, the cell(s) with the highest read density.

11. si_dicerz The z-score that is found at "shift 0", e.g., the probability that the read pairs have a proper 2nt overhang without shifting the read positions (see BioRxiv article for more information about how this is calculated).

12. num_si_dicer_reads The number of reads which have a proper 2nt overhang with their overlapping paired read.

13. hp_perc_paired When running the hairpin RNA- specific part of the siRNA module, RNAfold is used to fold the genomic sequence and the ratio of bases which are predicted to be paired due to complementarity is compared to the total number of bases.

14. hp_phasedz The average Z-score at nucleotides 1-3 for all overlapping read pairs (see BioRxiv for more information about phasing calculations).

15. hp_mfe From RNAfold, the minimum free energy calculated for the predicted folded structure.

16. hp_dicerz The z-score found at "shift 0" but for overlapping read pairs found by using the predicted paired positions from RNA fold (see BioRxiv article).

17. mi_perc_paired The percent of predicted paired nucleotides from RNAfold predictions. This is a separate metric from hp_perc_paired for the event that the locus may not have sufficient data to reach the folding step in one of the programs. For example, a long hpRNA would not be processed by the miRNA module if it exceeds 300nt in length.

18. mirna_dicerz The z-score found at "shift 0" but for overlapping read pairs found by using the predicted paired positions from RNA fold, but for miRNAs (see BioRxiv article).

19. mirna_mfe The minimum folding energy of the structure predicted by RNAfold from the miRNA module.

20. mirna_overlapz For all overlapping read pairs at the locus, the probability of the size of the overlap (for miRNAs, generally ~19nt).

21. pingpong_col The average read size of all read pairs overlapping by exactly 10nt (a charateristic of ping pong piRNAs).

22. max_pi_count The number of reads from (21).

23. max_piz_overlap For all overlapping read pairs at the locus, the overlap length with the highest z-score (e.g. ping pong piRNAs typically have a high probability of a 10nt overlap).

24. pi_phasedz For overlapping read pairs, the average z-score from nucleotides 1-3. This is useful for distinguishing phased piRNAs.

25. pi_phased26z The same as (24), but for only read pairs > 26nt in length.

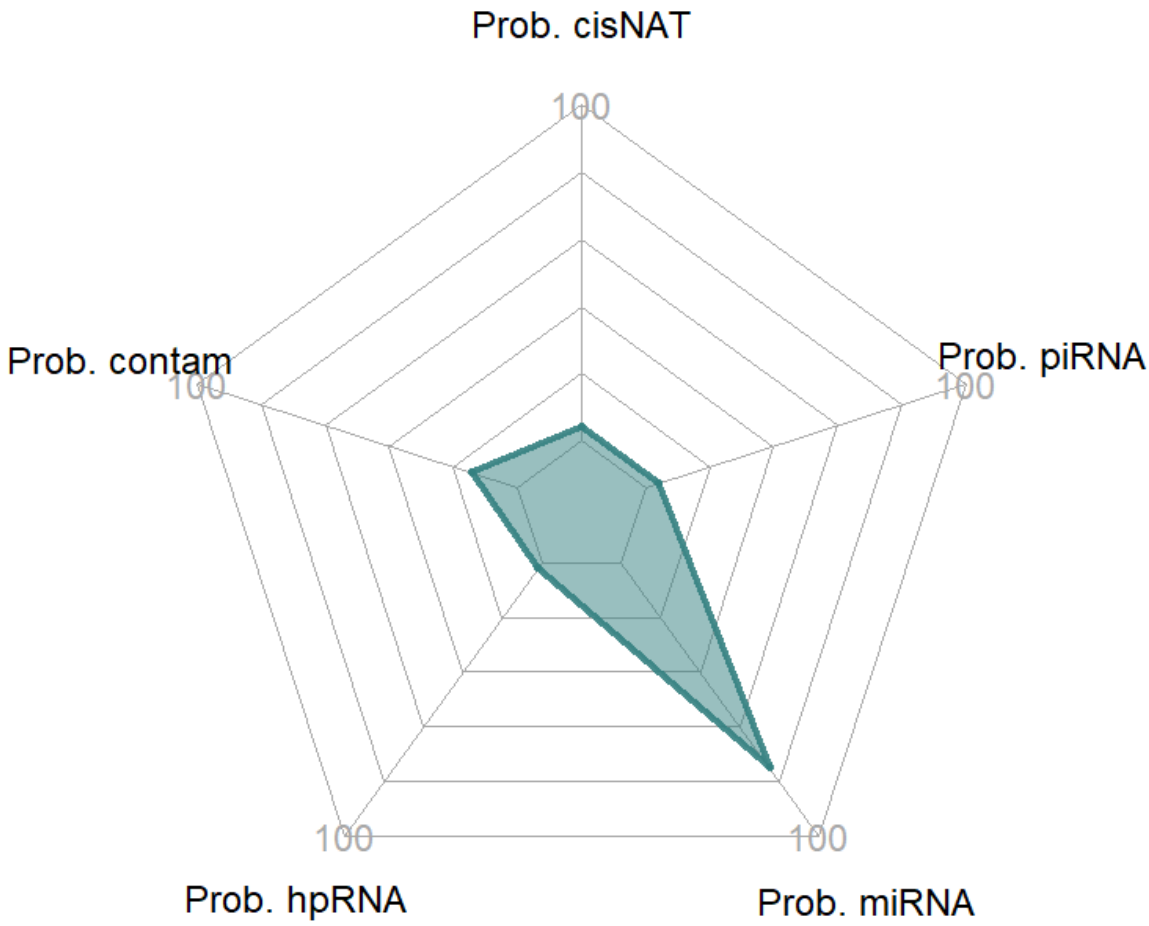
26. unique_read_bias The number of distinct read start and end positions, which may be useful for distinguishing loci such as miRNAs, simple repeats, or rRNAs.

Machine Learning

If you have a large list of loci (e.g. from calling clusters) and want to quickly determine which are more likely to be part of the small RNA pathway you are interested in (or not, such as rRNAs, snRNAs, tRNAs, etc.), you can use the output _ml table made from misipi_rna and the ml_probability module to make probability plots for each locus of interest. See the BioRxiv article for more information about how the MiSiPi.RNA machine learning model was made, or Build-A-Model section (coming soon).

```
#give the path to the directory that contains the folder and the name of the table
ml_probability("full/path/to/table/directory", "table_ml.txt")
```

The ml_probability module will create a subdirectory called “radar_plots” which will contain a probability plot for each locus.

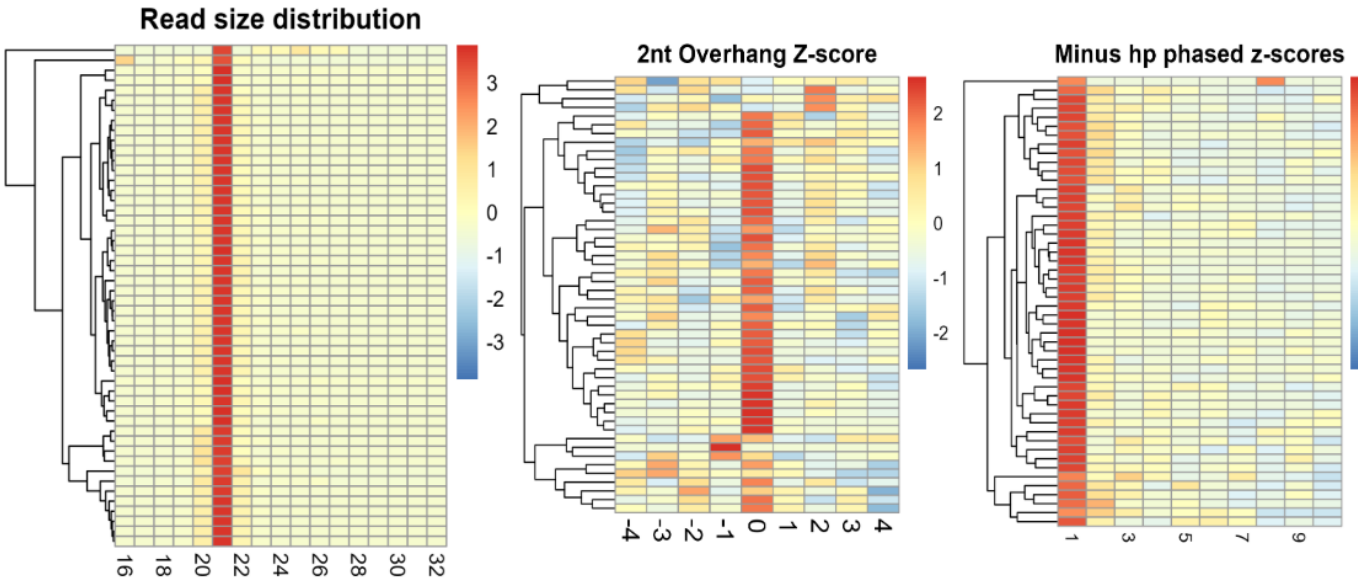


Make an html summary of all loci of interest

This module can be run after running misipi_rna over a large number of loci of interest. The output includes heatmaps of all loci from the original input BED file according to what small RNA type you choose, in addition to a sortable table of the metrics and statistics calculated for each locus. Misipi_rna produces the output tables and plots needed to generate the plots and HTML summary. misipi_rna produces a directory

called “run_all” which contains everything needed to make the summary file. For the argument “type” you should provide which small RNA pathway you are primarily interested in. The ml_plots argument is for users who have used the ml_probability function and is by default set to FALSE (see above machine learning documentation).

```
make_html_summary("full/path/to/run_all/", type = "piRNA", ml_plots = FALSE)
```



Show 10 entries

	locus	mi_plus_col	mi_minus_col	pi_col	si_col	prob_col	ave_size	perc_first_nucT	p
	All	All	All			All	Al	All	
1	chr2L-104062_104123	miRNA plus	miRNA minus	piRNA	siRNA	probability	21	0.3238095	
2	chr2L-419985_420702			piRNA	siRNA	probability	21	0.2286136	
3	chr2L-448521_448673	miRNA plus	miRNA minus	piRNA	siRNA	probability	21	0.08433735	
4	chr2L-2878578_2878933			piRNA	siRNA	probability	21	0.2369668	
5	chr2L-3021344_3021488	miRNA plus	miRNA minus	piRNA	siRNA	probability	21	0.3962264	
6	chr2L-3453706_3453753	miRNA plus	miRNA minus	piRNA	siRNA	probability	21	0.4012346	
7	chr2L-4442888_4442985	miRNA plus	miRNA minus	piRNA	siRNA	probability	21	0.1689189	
8	chr2L-4461666_4461798	miRNA plus	miRNA minus	piRNA	siRNA	probability	21	0.4927835	
9	chr2L-4948213_4948310	miRNA plus	miRNA minus	piRNA	siRNA	probability	21	0.2833333	
10	chr2L-4966284_4966389	miRNA plus	miRNA minus	piRNA	siRNA	probability	21	0.4064327	

Showing 1 to 10 of 49 entries

Troubleshooting

Each module outputs a log file of what loci are being run and when key processing steps are reached, as well as any warnings or errors that are encountered. This can be used to diagnose program failures or unexpected results.

1. What do values of -33 mean in the output tables? When the program does not have enough read data to calculate a metric, it uses -33 in place of "NA" so that the locus can still be fed into the machine learning module. This occurs often when there are too few reads meeting certain requirements for the calculation.
2. I don't see any features on the annotation plot or there is only a partial annotation. Check the GTF file used for the annotation plot to be sure there are features in the region of interest. To improve plots of partial annotations, it may be beneficial to include flanking regions of each locus (adding x bases to the stop and subtracting x bases from the start).
3. The modules run without error but the outputs are empty. This can happen if the chromosome names in the BED or BAM file do not match with the chromosome names found in the genome fasta file. Check that the names in all three files follow the same naming system.

Cluster calling pipeline

This basic pipeline can be used to create a list of regions of interest in under-annotated genomes for which small RNA is available. The suggested computational environment required to run this script is a high performance compute cluster with the following programs installed in your path:

- samtools
- bowtie (not bowtie2)
- bedtools

```
#Identifying small RNA regions of interest
#miRNAs & siRNAs: get reads of only miRNA and siRNA length
awk 'BEGIN {OFS = "\n"} {header = $0; getline seq; getline qheader ; getline qseq ; if (length(seq) >= 

#piRNAs: get reads of piRNA length
awk 'BEGIN {OFS = "\n"} {header = $0; getline seq; getline qheader ; getline qseq ; if(length(seq) >= 23
< trimmed.fq > large.fastq

#Align reads to genome
bowtie -p 10 -a -m 100 --best --strata --no-unal genome.fna small.fastq -S | samtools view -@ 10 -q 10 

samtools index small.bam

bowtie -p 20 -a -m 100 --no-unal genome.fna large.fastq -S | samtools view -@ 10 -q 10 -b | samtools so
samtools index large.bam

#Calculate coverage of all reads over genome
bedtools genomecov -bg -ibam all.bam | awk '$4 > 100' > HE.tmp.bedgraph

#Get large regions of high expression
bedtools merge -d 500 -i HE.tmp.bedgraph > HE.tmp.merge.bed
awk '{n=$2; x=$3; print $1"\t"$2"\t"$3"\t"x-n}' < HE.tmp.merge.bed | awk '$4 > 40' > HE.all.bed #all hi
```

```
#Get potential regions of high miRNA/siRNA RNA expresssion
bedtools multicov -bams small.bam all.bam -bed HE.all.bed | awk '{n=$5; x=$6; print $1"\t"$2"\t"$3"\t"n/

#Get potential regions of high piRNA expression
bedtools multicov -bams large.bam all.bam -bed HE.all.bed | awk '{n=$5; x=$6; print $1"\t"$2"\t"$3"\t"n/
```