A user’s manual for miR-Island

Current as of miR-Island version 2.0

Tiantian Gao, Xin Meng, Wei Zhang

Contents

OVERVIEW

Next-generation sequencing of small RNAs have provided rich information on microRNAs profile of various plant species. However, few computational tools are presently available to analyze these data effectively. miR_Island is a local tool that analyzes plant small RNAs and quantifies small RNA genes. In this study, miR_Island performs unprecedented speed and efficient memory-usage in plant microRNA annotations.

SUMMARY OF MIR-ISLAND FUNCTIONS

Based on ultra-deep sampling of small RNA libraries by next generation sequencing, miR_Island has a lot of advantages in annotation and quantification of plant miRNA genes. miR_Island can be used to identify miRNA genes in plant species with or without annotations. miR_Island can identify miRNA genes in an ultra-fast speed, no matter at which assemble level the reference sequences are. miR_Island is properly for miRNA quantification and differentially miRNA expression. miR_Island offers user-friendly tabular outputs and publication-ready results.

IMPLEMENTATION AND ALGORITHM

miR_Island is documented by Perl (Perl 5.10 or later versions) and other fundamental packages from Perl library. All the scripts have been tested on two Linux platforms, Centos 6.4 and RedHat 5.4, and should work on similar systems that support Perl.
The core algorithm of miR_Island was developed by modifying miRDeep (Friedlander et al., 2008), which is based on a modified probabilistic model of miRNA biogenesis and a series of plant-specific filtering criteria (Meyers et al., 2008).

LICENSE AND AVAILABILITY

miR_Island is freely available under a GNU Public License (Version 3) at:
https://github.com/janeyurigao/miR-Island
The miR_Island scripts and user manual can be obtained from the web sites.

INSTALLATION OF THIRD PARTY SOFTWARE

Note: To reduce time consuming, multiple threads specificity is applied in miR_Island. Your perl version must be equal to or greater than 5.10. You can check the perl version by running perl -v, and see if it is possible to use multiple threads by running perl -V and looking at the Platform section. If you have useithreads=define you can use the specificity.
Several dependencies are required to run miR_Island.
1: bowtie (used for aligning small RNA-seq to reference genome). The bowtie package can be

downloaded from the site: http://sourceforge.net/projects/bowtie-bio/files/bowtie/. Our test version is 0.12.9.
2: Samtools (tools for alignments in the SAM format). Samtools can be downloaded from the site: http://samtools.sourceforge.net/ . Our test version is 0.1.19.
3: ViennaRNA (RNAfold is used for predicting hairpin structure). The Vienna RNA package is currently at http://www.tbi.univie.ac.at/RNA/ . Our test version is 2.0.0.
4: bioperl (used to improve the operation efficiency). Users are recommended to install bioperl according to http://www.bioperl.org/wiki/Installing_BioPerl . Our test version is 1.6.901.
5: Bio::DB::Sam (used to handle the SAM/BAM format alignment). When you install Bio::DB::Sam module, you should depress the tar.gz of samtools and change directory to the depressed folder, then run "make CXXFLAGS=-fPIC CFLAGS=-fPIC CPPFLAGS=-fPIC". At last, you install Bio::DB::Sam module.

INSTALL MIR-ISLAND

Download the zip file from https://codeload.github.com/janeyurigao/miR-Island/zip/master and unpack. Then put the scripts into your PATH. This is done by typing the following lines:

```
echo 'export PATH=/path/to/miR_Island:$PATH' >> ~/.bashrc
source ~/.bashrc
```

Note: "/path/to/miR_Island" should be changed to the real location of miR_Island folder.
You can test by typing miR_island.pl on the command line with no parameters. You should get a help message.

INPUT DATA

1. The reference genome in FASTA format is required. Any assemble level (contig level, scaffold level and chromosome level) is ok. If no genome is available, users can use EST/cDNA sequences instead.
2. RNA-seq data sets, which should be in FASTA format.
3. Known miRNAs in FASTA format is encouraged, which is not essential.
4. Tab-delimited text file that contains the location of known microRNA precursors (only separate the location and the name with a TAB), which is not essential:

```
    Chr2 - 10676451 10676573        ath-MIR156a
    Chr4 + 9888982 9889070          ath-MIR160b
```

ATTENTION: only characters of "A/T/C/G" are allowed in the above input FASTA files.

PREPROCESSING DATA SETS

If the raw reads is SRA format, you should install SRA Toolkit first. The current available version is at http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=software . Then run fastq-dump SRRxxx.sra, while xxx stands for the SRR number. You should get a file in FASTQ format named as SRRxxx.fastq. Detection and Removal of adapters need two scripts including

"find_3p_adapter.pl" and "trim_illumina_sRNA_fastq.pl" at http://axtell-lab-psu.weebly.com/tools.html (Axtell, 2013).
Detect adapters by running find_3p_adapter.pl -m ugacagaagagagugagcac < SRRxxx.fastq, while "ugacagaagagagugagcac" stands for the sequence of ath-miR156a. Users are welcome to read the script's help document first.
Remove adapters by running trim_illumina_sRNA_fastq.pl -a TCGTATGC -e 1 -q N -o SRRxxx.fa -p SRRxx_ps.fa < SRRxxx.fastq, while "TCGTATGC" is the adapter sequence. The RNA-seq data sets must be parsed into FASTA format.

PLANT MICRORNAS ANNOTATION AND QUANTIFICATION

Given that you have a tomato sRNA data set, whose cotyledon type is dicot, and there is a lot of known miRNAs referred to this species available in miRBase, miR_Island can run with the following command:

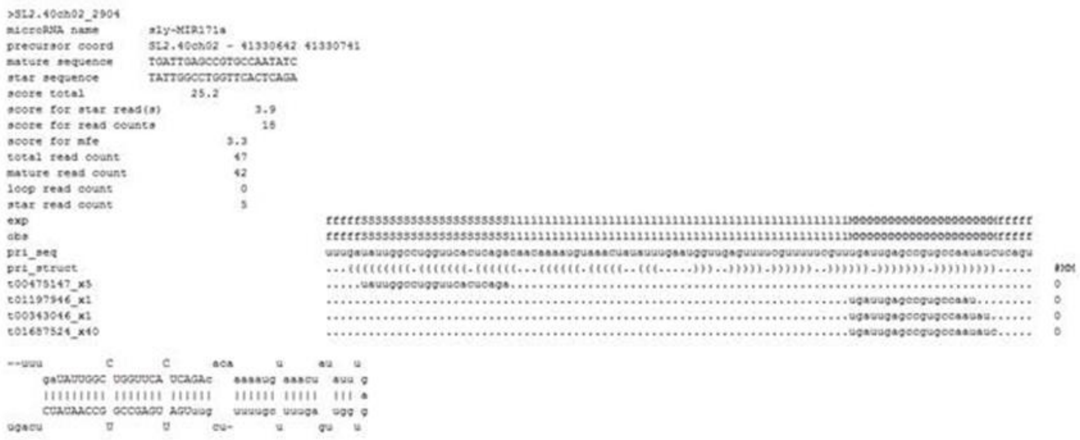
```
miR_island.pl -c dicot -g genome.fa -r sRNA_clean.fa -k miRNA.fa -p 4 -x coord.list
```

It will generate a directory names "MiR_Island_run" and a sub-directory names run_xxxxxx, while xxxxxx is a series number returned by the function of localtime in perl.
You will get two miRNA expression lists: "denovo_miRNA_expression.list"; "known_miRNA_expression.list".
In the above two expression lists, NoN_mature_exp stands for the mature sequence raw reads count, NoN_star_exp stands for the star sequence raw reads count. NoN_mature_std stands for the mature sequence RPM, NoN_star_std stands for the star sequence RPM. The character N in red color represents integer, and it is in accordance with the order of the input sRNA data sets.
When there are two small RNA data sets, "Ctl" or "Trt" will take the place of "NoN", "Ctl" stands for the first small RNA data set, and "Trt" stands for the second small RNA data set. The Log2 transformed Fold Change and P-value will be calculated (Audic and Claverie, 1997; Benjamini and Yekutieli, 2001).

However, if the known miRNAs referred to this species were not specified, miR_Island can run with the following command:

```
miR_island.pl -c dicot -g genome.fa -r sRNA_clean.fa -p 4
```

This time no known miRNA expression list will exist.
In addition, an intact flat file named "result.out" will be generated, one sample is as follows:



```
>SL2.40ch02_2904
microRNA name          sly-MIR171a
precursor coord        SL2.40ch02 - 41330642 41330741
mature sequence        TGATTGAGCCGTGCCAATATC
star sequence          TATTGGCCTGGTTCACTCAGA
score total                  25.2
score for star read(s)                3.9
score for read counts                  18
score for mfe                     3.3
total read count                   47
mature read count                  42
loop read count                     0
star read count                     5
exp                                        fffffSSSSSSSSSSSSSSSSSSSSS111111111111111111111111111111111111111111111111ooooooooooooooooooooofffff
obs                                        fffffSSSSSSSSSSSSSSSSSSSSS111111111111111111111111111111111111111111111111ooooooooooooooooooooofffff
pri_seq                                    uuugauauuggccugguucacucagacaacaaaauguaaacuauauuugaaugguugaguuuucguuuuucguuugauugagccgugccaauaucucagu
pri_struct                                 ...(((((((((.(((((((.((((((...((((((.(((((..(((.....)))..))))).))))))..)))))).))))))).)))))))))..... #MM
t00475147_x5                               .....uauuggccugguucacucaga..........................................................................  0
t01197946_x1                               ..........................................................................ugauugagccgugccaau........  0
t00343046_x1                               ..........................................................................ugauugagccgugccaauau......  0
t01687524_x40                              ..........................................................................ugauugagccgugccaauauc.....  0

--uuu         C       C       aca     u     au  u
     gaUAUUGGC UGGUUCA UCAGAc   aaaaug aaacu  auu g
     ||||||||| ||||||| ||||||   |||||| |||||  ||| a
     CUAUAACCG GCCGAGU AGUuug   uuuugc uuuga  ugg g
ugacu         U       U       cu-     u     gu  u
```

If the coordinate of a predicted microRNA precursor is in accordance with that in coord.list, microRNA name will be specified. The precursor coordinate contains four parts: chromosome, strand, begin and end. The tail is a figure of the secondary structure (Shen et al., 2012), and the mature or star region are in Upper characters.

ABOUT MIR-ISLAND PACKAGE

miR_Island package has a total of six perl scripts:
1. miR_island.pl
2. transform_genome.pl
3. diff_miRNA_expression_analysis.pl
4. excise_potential_precursors.pl
5. RNAfold_with_multi-threads.pl
6. miRNA_island_core.pl
You can get each help message by simply typing the script name.

miR_island.pl

miR_Island: an ultrafast package for annotation and quantification of miRNA and MIRNA genes with High throughput sequencing

Version = 2.0

WARNING: You did not provide enough information!

```
Usage: miRNA_island.pl -c <monocot|dicot> -g <genome.fa> -r <reads1.fa[,...,readsN.fa]> \
                    [-i <bowtie_ebwt>] [-e <mfe>] [-f <min_freq>] [-m <max_hits>] [-k <miRNA.fa>] \
                    [-l <haipin_length>] [-p <num_threads>] [-x <pre-miRNA_coord>]
```

REQUIRED:

```
-c    <monocot|dicot>              cotyledon type: monocot | dicot
-g    <genome.fa>                  reference genome file in multi-fasta format
```

```
-r   <reads1.fa[…readsN.fa] >      a comma-separated list of files with small RNA reads in
                                   multi-fasta format
OPTIONAL:
-e   <min_mfe>            maximum threshold MFE to be annotated as a potential precursor (def: -
                          17.5)
-f   <int>                minimum frequency of reads to trigger a precursor excising (def: 15)
-I   <bowtie_ebwt>        prefix of the reference genome's bowtie indexes file (def: same as option
                          "g")
-k   <miRNA.fa>           known miRNAs of the related species in multi-fasta format
-l   <int>                maximum number of sites that a read could map to related genome (def:
                          10)
-p   <int>                number of threads to use (def: all threads available)
-x   <coord>              known pre-miRNA coordinates in tab-delimited file (only seperate the
                          location and the pre-miRNA name with a TAB (such as):
                          Chr2 - 10676451 10676573 ath-MIR156a
                          Chr4 + 9888982 9889070 ath-MIR160b
-h                        print intact help message
```

transform_genome.pl

transform_genome.pl transfoms scaffold-level genome into chromosome-level and vice versa; it can also be used to transform the genome to multi-fasta with sequences in same length

```
WARNING: You did not provide enough information!

Usage:    transform_genome.pl -d <genome.fa> -i <index_file> -t <T> -o <pseudo_genome.fa>
or:       transform_genome.pl -d <data> -i <index_file> -t <F> -o <data_processed>
or:       transform_genome.pl -d <genome.fa> -t <O> -o <genome_with_same_length.fa>
OPTIONS:
-d        data to transform or restore
-I        index file that record scaffold info
-o        specify the output file
-t        function type to use: T/F/O
          T to transform scaffold-level genome to chromosome-level
          F to restore chromosome-level info to scaffold-level
          O to transform genome with same sequence length
-h        print intact help information
```

diff_miRNA_expression_analysis.pl

diff_miRNA_expression_analysis.pl aims at analyzing differential microRNA expression data. It will output a list with standard gene count, LFC (log2 transformed foldhange), pvalue, FDR and significance, etc.

```
Usage:
        diff_miRNA_expression_analysis.pl -m <miRNA.fa> -c <combined_reads.fa> \
        -r file_1.fa[,...,file_N.fa] -o <file_out>
or
        diff_miRNA_expression_analysis.pl -l <exp_list> -r file1.fa[,...,fileN.fa] \
        -o <file_out> [-m <miRNA.fa>]
Options:
-c    <reads.fa>                 a combined multi-fasta file of clean reads generated by
                                 format_clean_reads.pl
-l    <exp.list>                 a tab-delimited file contains microRNAs sequence info
-m    <miRNA.fa>                 a multi-fasta format file contains known microRNAs
-r    <file_1.fa[,...,file_N.fa> a comma-separated list of files with small RNA reads in multi-
                                 fasta format
-o    <exp.out>                  a tab-delimited output file

Note:
I.    option 'c' and option 'l' are mutually exclusive;
II.   The input reads should be formated by format_clean_reads.pl. The reads
      file should have each entry with unique sequence, the entry must be as
      CR_xN', while C represents one or more letters, R represents a non-redundant
      running number, and N represents current read count, _x is the separator.
III.  Pvalues, FDR and significance will be calculated only in the case of two samples.
```

excise_potential_precursors.pl

excise_potential_precursors.pl excises potential miRNA precursors with high throughput sequencing.

```
Version = 2.0

WARNING: You did not provide enough information!

Usage: excise_potential_precursors.pl -b <reads_vs_genome_sorted.bam> -r <genome.fa> \
           -o <precursors.fa> [-m <min_freq>]
REQUIRED:
-b    <sorted.bam>      profile of reads mapped to the related genome in a formatted, sorted and
                        indexed BAM file
-r    <genome.fa>       a related genome in multi-fasta format
-o    <precursors.fa>   an output file in multi-fasta format

OPTIONAL:

-l    <int>             maxmium length of the potential precursor to be excised (default: 277)
-m    <int>             minimum frequency of reads to trigger a precursor excising
```

RNAfold_with_multi-threads.pl

```
RNAfold_with_multi-threads.pl computes potential precursors using RNAfold with multiple threads

Version = 2.0

WARNING: You did not provide enough information!

Usage: RNAfold_with_multi-threads.pl -i <precursors.fa> -d <tmp_dir> \
                                     -o <precursor.struct> [-p <num_threads>]

REQUIRED:
-I    <precursors.fa>      a multi-fasta file with each sequence in a line
-d    <tmp_directory>      a directory to store the temp files when running
-o    <precursors.struct> an RNAfold output file with sequences and structures

OPTIONAL:
-p    <num_threads>        number of threads to use
```

miR_island_core.pl

```
miR_island_core.pl is the miR_island package's core algorithm for plant microRNA annotation.

Version = 2.0

WARNING: You did not provide enough information!

Usage: miR_island_core.pl -b <reads_vs_genome_sorted.bam> -c <monocot|dicot> -g \
<genome.fa>   -s <structure_file> -t <table_out> -r <result_out> [-m <miRNA.fa>] [-l \
<col_width>]

REQUIRED:

-b    <sorted.bam>         profile of reads mapped to the related genome in a formatted, sorted
                           and indexed BAM file
-c    <monocot|dicot>      cotyledon type: monocot | dicot
-g    <genome.fa>          reference genome file in multi-fasta format
-s    <structure_file>     secondary structure file produced by RNAfold
-r    <result_out>         an intact output file that annotates miRNA and MIRNA genes
-t    <table_out>          a brief tab-delimited text file that annotates miRNA and MIRNA genes

OPTIONAL:
```

-l	<column_width>	coloum 1 width, if identifier is long, add this value (def: 40)
-m	<miRNA.fa>	known miRNAs of the related species in multi-fasta format
-v		show the current version number
-h		print intact help message

REFERENCES

Audic, S., & Claverie, J. M. (1997). The significance of digital gene expression profiles. Genome research, 7(10), 986-995.

Axtell, M. J. (2013). ShortStack: Comprehensive annotation and quantification of small RNA genes. RNA, 19(6), 740-751.

Benjamini, Y., & Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. Annals of statistics, 1165-1188.

Friedlander, M. R., Chen, W., Adamidi, C., Maaskola, J., Einspanier, R., Knespel, S., & Rajewsky, N. (2008). Discovering microRNAs from deep sequencing data using miRDeep. Nature biotechnology, 26(4), 407-415.

Meyers, B. C., Axtell, M. J., Bartel, B., Bartel, D. P., Baulcombe, D., Bowman, J. L., ... & Zhu, J. K. (2008). Criteria for annotation of plant MicroRNAs. The Plant Cell Online, 20(12), 3186-3190.

Shen, W., Chen, M., Wei, G. & Li, Y. (2012). MicroRNA Prediction Using a Fixed-Order Markov Model Based on the Secondary Structure Pattern. PLoS One 7, e48236.