User Manual for MTide

Last updated: 31-August-2014

Author: ZhaoZhang

From: Zhejiang University

Email: Zhangzhao87@zju.edu.cn

Introduction

This is a brief tutorial to describe the usage of MTide. MTide is an integrated pipeline designed to parse sRNA-seq and degradome data for miRNA-target identification in plant. It can quantify the known miRNA expression and identify novel miRNA from sRNA-seq data, identify the target of miRNA from degradome data signature, predict target of miRNA precisely, prioritize predicted target according to GO similarity to known or validated target, and identify the expressed miRNA between two samples.

Mtide includes four modules and eight steps for an overall survey of miRNA and its target. It is suitable for different experiment design: a) just one sample with sRNA-seq data; b) just one sample with degradome data; c) just one sample with sRNA-seq and degradome data; d) paired samples with sRNA-seq or/and degradome data.

The core algorithm consists of a modified miRDeep2 and a modified CleaveLand4. We delete some fat scripts, modify some parts of scripts affecting overall speed, add support for multiple threads, add support for plant, and report more information compared to the original ones. After refining these scripts, MTide can run very fast and precisely, and as some other tools have been added, such as TAPIR, a precise miRNA target prediction tool, DESeq, a nice R package for differential expression analysis, MTide can be a good tool for comprehensive miRNA and target analysis in plants.

Each step of Mtide can run separately for a custom design, or it can run in an integrated way by MTide.pl, which is a wrapper script for all of these eight steps.

Overview of MTide scripts:

MTide.pl: A wrapper for all of these eight steps.

1	Module 1: reads processing		
	1.1	Step 1: CleanReads.pl	Clean and collapse reads
	1.2	Step 2: filter.pl	Filter reads mapped to other non-coding RNA
	1.3	Step 3: map_parse.pl	Map reads to genome and parsed to arf format
	1.4	Step 4: miRDeep2.pl	Identification of known and novel miRNA
	1.5	Step 5: de_for_miRNA.pl	Differential expression of miRNA for two samples
2	Module 2: target identification		
	2.1	Step 6: CleaveLand4.pl	Target identification from degradome sequencing data
3	Module 3: target prediction		
	3.1	Step 7: tapir_wrapper.pl	Target prediction using TAPIR
4	Module 4: target prioritization		
	4.1	Step 8: prioritization.pl	Prioritization of predicted target

Installation

Dependencies:

Several dependencies are required to run MTide. Please make sure perl 5.10 or later version, python 2.7 or later version and R 2.5 or later version have been installed in your compute.

1. Math::CDF. CleaveLand4.pl will not compile unless this required perl module is installed.
2. PDF::API2. For pdf file generating in miRDeep2.pl.
3. cutadapt. See http://code.google.com/p/cutadapt/.
4. samtools. See http://samtools.sourceforge.net/.
5. bowtie and bowtie-build. See http://bowtie-bio.sourceforge.net/index.shtml. Note, do NOT use 'bowtie2', as it is a very different type of aligner and bowtie is used for small fragment reads.
6. RNAplex and RNAfold. These are part of the Vienna RNA package.
   See http://www.tbi.univie.ac.at/~ronny/RNA/index.html.
7. SQUID. See http://selab.janelia.org/software.html. Goto Squid and download it.
8. randfold. See http://bioinformatics.psb.ugent.be/software/details/Randfold.
9. csbl.go. This is a module of R, used for GO similarity analysis.
   See http://csbi.ltdk.helsinki.fi/csbl.go/.
   You also need to install some Bioconductor packages (Biobase, annotate, GO.db).
   It's used in prioritization.pl.
   You have no need to install this dependency if you don't want to do GO similarity analysis for prioritization of target predicted by other tools.
10. TAPIR. See http://bioinformatics.psb.ugent.be/webtools/tapir/.
    It's used in tapir_wrapper.pl.
    You have no need to install this dependency if you don't want to predict target of miRNA.
11. plyr. This is a module of R.
12. DESeq. This is a module of R for differential expression analysis.

Installation Example:

All the packages have been tested on Ubuntu 12.04 and Fedora 8 x64 platforms, and should work on similar system that support perl, python and R.

First, download all necessary packages listed here:

a) cutadapt (http://code.google.com/p/cutadapt/) Version: 1.4.1
b) bowtie short read aligner (http://bowtie-bio.sourceforge.net/index.shtml) Version: 1.0.1
c) Vienna package with RNAfold (http://www.tbi.univie.ac.at/~ivo/RNA/) Version: 2.1.17
d) SQUID library (http://selab.janelia.org/software.html) Version: 1.9g
e) randfold (http://bioinformatics.psb.ugent.be/software/details/Randfold) Version: 2.0
f) PDF::API2 (http://search.cpan.org/search?query=PDF::API2&mode=all) Version: 2.021
g) samtools (http://samtools.sourceforge.net/) Version: 0.1.19
h) csbl.go (http://csbi.ltdk.helsinki.fi/csbl.go/) Version: 1.4.1
i) TAPIR (http://bioinformatics.psb.ugent.be/webtools/tapir/). Version: 1.1

Second, install all the dependencies.

Suggest your home directory is /home/test, and all the packages will be downloaded in

/home/test/download, and installed in /home/test/soft.

cd /home/test/soft

a) tar –zxvf ../download/cutadapt-1.4.1.tar.gz
cd cutadapt-1.4.1 → python setup.py build → cd ../

b) unzip ../download/bowtie-1.0.1-src.zip
cd bowtie-1.0.1 → make → cd ../

c) tar –xvfz ../download/ViennaRNA-2.1.7.tar.gz
cd ViennaRNA-2.1.7 → ./configure –prefix=/home/test/soft/ViennaRNA-2.1.7 → make → make install → cd ../

d) tar –zxvf ../download/squid.tar.gz
cd squid1.9g → ./configure → make → cd ../

e) tar –zxvf ../download/randfold-2.0.tar.gz
cd randfold-2.0 →
edit Makefile and change line with INCLUDE=-I. to
INCLUDE=-I. –I/home/test/soft/squid1.9g -L/home/test/soft/squid-1.9g
make → cd ../

f) tar –zxvf ../download/PDF-API2-2.021.tar.gz
cd PDF-API2-2.021 → perl Makefile.pl → make → sudo make install → cd ../

g) tar –jxvf ../download/samtools-0.1.19.tar.gz2
cd samtools-0.1.19 → make → cd ../

h) enter R shell
source(“http://bioconductor.org/biocLite.R”)
biocLite(“DESeq”)
biocLite(“Biobase”)
biocLite(“annotate”)
biocLite(“GO.db”)
install.packages(“csbl.go_1.4.1.tar.gz”, repos=NULL)
install.packages(“plyr”)

i) tar –zxvf ../download/tapir-1.1.tar.gz

j) if you have cpan command in your computer
just run “cpan Math::CDF”

Third, install MTide.
tar –zxvf ../download/MTide.tar.gz

Fourth, attach the executable path to your PATH:
echo ‘SOFT=/home/test/soft’ >>~/.bashrc
echo ‘export
PATH=$SOFT/cutadapt-1.4.1/bin:$SOFT/bowtie-1.0.1:$SOFT/ViennaRNA-2.1.7/Progs:$SOFT/samtools-0.1.19:$SOFT/randfold-2.0:$SOFT/tapir-1.1:$SOFT/MTide/scripts:$PATH’ >>~/.bashrc
source ~/.bashrc

Use Example

The tutorial data can be downloaded from http://bis.zju.edu.cn/MTide/tutorial_data.tar.gz. We tested in a computer containing two Intel Xeon E5-2620 CPUs. It took about 8 hours to finish all the analysis. If your computer is more powerful, we suggest you use more threads while running MTide.

The tutorial example runs in an integrated way for one sample experiment. As the small RNA sequencing file has been cleaned and collapsed, the first step will be skipped. The configure file looks like:

MTide.conf

```
Threads_number=20
quiet_mode=no
genome_file=TAIR10_genome.fa
transcriptome_file= TAIR10_cdna.fa
go_annotation_file=ath.go
miRNA_mature_file=ath_mature.fasta
miRNA_precursor_file=ath_hairpin.fasta
nonRNA_libs_file=Rfam_for_miRDeep.fa
sample1_prefix=seq
sample1_srna_file=sRNA_cleaned.fa
sample1_deg_file=GSM278370.fasta
step1_allowed=no
step2_allowed=yes
filter_mismatches_allowed=2
step3_allowed=yes
seed_length=18
genome_mismatches_allowed=2
maximal_alignments=15
step4_allowed=yes
disable_pdf=yes
optimal_length=250
maximum_pre_number=50000
step5_allowed=no
step6_allowed=yes
t_plots_allowed=no
cleaveland_mfe_ratio=0.65
maximum_category=4
pvalue=0
step7_allowed=yes
transcriptome_from_cleaveland=yes
tapir_score_cutoff=4
tapir_mfe_ratio=0.65
ste8_allowed=yes
```

All you need is just edit this configure file, copy to the tutorial directory, and run MTide.pl with it.

The result directory tree illustration:
(The important file will be denoted with red color.)

```
MTide_run13_06_2014_t_10_15_24     (result directory with the running start time)
|-- CleaveLand_for_sample1         (like CleaveLand4 directory)
|   |-- GSM278370.fasta_dd.txt     (signature density file)
|   |-- all_mirna_sample1.fa_GSTAr.txt       (after running GSTAr.pl)
|   |-- cleaveland_sample1.result            (final result list of CleaveLand4)
|   `-- tx_for_GSTAr.tmp.fasta               (temporary transcriptome, with degradome signature)
|-- all_mirna_sample1.fa                     (all miRNA identified from small RNA-seq)
|-- bowtie_index                             (bowtie index directory)
|-- filtered_sRNA_cleaned.bwt                (bwt format of genome mapped file)
|-- filtered_sRNA_cleaned.fa                 (small RNA-seq data after filtering other non-coding
                                             RNA)
|-- filtered_sRNA_cleaned_vs_genome.arf      (arf format of genome mapped file)
|-- miRDeep_for_sample1                      (like miRDeep2 directory)
|   |-- chr_length
|   |-- error_13_06_2014_t_10_22_06.log
|   |-- expression_13_06_2014_t_10_22_06.html
|   |-- expression_analyses
|   |-- miRNAs_expressed_all_samples_13_06_2014_t_10_22_06.csv    (know miRNA expression file)
|   |-- mirdeep_runs
|   |   |-- output.mrd                       (predicted secondary structure with mapped reads)
|   |   |-- run_13_06_2014_t_10_22_06_parameters
|   |   `-- survey.csv
|   |-- result_13_06_2014_t_10_22_06.csv          (The main result file of miRDeep2, including
                                                   the novel miRNA and known miRNA)
|   `-- result_13_06_2014_t_10_22_06.html
|-- predicted_sample1.result                 (predicted target file by TAPIR)
|-- predicted_sample1.result.go              (predicted target file after prioritization)
`-- sRNA_cleaned_vs_Rfam_for_miRDeep.fa      (small RNA-seq file which could be mapped to
                                             Rfam)
```

Files Description

Input

For the input files, genome_file, transcriptome_file, miRNA_mature_file, miRNA_mature_other_file, miRNA_precursor_file, nonRNA_libs_file, srna_file and degradome sequence file, are standard fasta format files and can be set in MTide.conf.

Apart from these files, the go_annotation_file should be in format like “Gene/mRNA GOid” with each line separated by tab. Below is an example of AT1G01073 in Arabidopsis thaliana:
AT1G01073 GO:0003674 GO:0008150 GO:0009507

Output

Take the output files in tutorial data as an example:

For miRDeep2:

The files in directory miRDeep_for_sample1 are as the same as the original miRDeep2 result

1. miRNAs_expressed_all_samples_13_06_2014_t_10_22_06.csv

The reads count of all the known mature miRNAs in miRBase:

#miRNA	read_count	precursor	total seq	seq(norm)	
ath-miR156a	9835.10	ath-MIR156a	9835.10	9835.10	11936.97
ath-miR156b	9855.74	ath-MIR156b	9855.74	9855.74	11962.01
ath-miR156c	9835.10	ath-MIR156c	9835.10	9835.10	11936.97

2. result_13_06_2014_t_10_22_06.csv

This file includes four parts: the first is the overview performance of miRDeep2, the second is the detail information of novel miRNAs predicted by miRDeep2, the third is the detail information of mature miRBase miRNAs detected by miRDeep2, and the last is the information of miRBase miRNAs not detected by miRDeep2.

For CleaveLand4:

1. cleaveland_sample1.result

This is the final running result of CleaveLand4. It contains 19 columns as below:

SiteID	A general ID of miRNA and target transcript
Query	Query miRNA name
Transcript	Transcript name
TStart	Start miRNA binding site in transcript
Tstop	Stop miRNA binding site in transcript
Tslice	The site complementary to the 10^{th} miRNA site
TrueTSlice	The true cleave site
SliceType	Type of cleave. It could be 9, 10, or 11
MFEperfect	Perfect MFE value
MFEsite	Actual miRNA::transcript MFE value
MFEratio	MFE ration of MFEsite to MFEperfect

AllenScore	Allen score of miRNA::transcript
Paired	Paired sites
Unpaired	Unpaired sites
Structure	Base pair structure of miRNA and transcript
Sequence	Complementary sequence of miRNA and transcript
DegradomeCategory	Category of signal. It could be 0, 1, 2, 3, 4.
DegradomePval	P-value
Tplot_file_path	The T-plot file path of each miRNA-target pairs

2. all_mirna_sample1.fa_GSTAr.txt

The miRNA and predicted target file. The explanation of each column is the same as part of the cleaveland_sample1.result file.

3. GSM278370.fasta_dd.txt

This is the degradome density file of all the transcripts with reads mapped to them.

4. tx_for_GSTAr.tmp.fasta

It is the transcripts sequence file, but excludes transcripts having no reads mapped to them. This strategy could speed up the overall running time compared to original CleaveLand4.

For other files:

1. all_mirna_sample1.fa

It includes all the miRNAs sequences detected by miRDeep2 in step4, and is generated from "result_13_06_2014_t_10_22_06.csv".

2. filtered_sRNA_cleaned.fa

The small RNA reads file after filtering reads that could be mapped to other non-RNA libraries, like Rfam.

3. filtered sRNA_cleaned.bwt

The result bowtie format file after mapping the filtered file to genome.

4. filtered_sRNA_cleaned_vs_genome.arf

The arf format file transformed from bwt file

5. predicted_sample1.result

It is the predicting file of all the miRNAs in all_mirna_sample1.fa file using TAPIR. Some of the lines:

#miRNA	target	score	mfe	mfe_ratio	start
ath-miR156a	AT1G69170.1	1	-38.2	0.9340	1296
ath-miR156b	AT1G69170.1	1	-38.2	0.9340	1296
ath-miR156e	AT1G69170.1	1	-38.2	0.9340	1296
ath-miR156f	AT1G69170.1	1	-38.2	0.9340	1296

6. predicted_sample1.result.go

After running prioritization.pl, all the predicted target were scored based on GO similarity to the targets which could be detected from degradome sequencing file.

Mir	The name of the mature miRNA
Gene	The gene name of the transcript
GO	GO ids
Validated	V: detected from degradome seq; P: predicted by TAPIR
MF_score	Molecular function similarity score
BP_score	Biological process similarity score
CC_score	Cellular component similarity score
All_score	Add all the scores.

7. sRNA_cleaned_vs_Rfam_for_miRDeep.fa

This is the small RNA sequencing reads file that could be mapped to Rfam_for_miRDeep.fa.

Script reference

1. Integrated wrapper (MTide.pl)

Description: This script is an integrated wrapper of all the scripts listed down. It only contains one option, which is a configure file, and users can modify this file for controlling the running procedure.

Usage: MTide.pl –c mtide.conf

Options:

-c/--conf [string]	configure file of MTide.pl. [default: mtide.conf]

2. Processing the reads (CleanReads.pl)

Description: This script removes the adapters from 5’ and 3’ of the reads ends, discards the reads shorter than 18 or longer than 30 by default, and then removes the redundancy such that reads with identical sequence are represented with a single FASTA entry. Therefore, each sequence identifier must like “>rd1_read1_x15”, ‘rd1’ means the three-letter suffix of the sample, and ‘x15’ means the number of the unique read.

Usage: CleanReads.pl -q -f fasta -i reads1.fa=rd1 -i reads2.fa=rd2

Options:

-h/--help	Print help message and quit
-v/--version	Print version and quit
-q/--quiet	Quiet mode .. no log/progress information to STDERR
-i/--intput [input1.fa=rd1]	input files, should be in format like reads1.fa=rd1, 'rd1' denotes the three-letter prefix of the input file, and this option can exist many times
-a/--adap3 [string] :	3' adapter sequence (default: 'TCGTATGCCGTCTTCTGCTTG')
-g/--adap5 [string] :	5' adapter sequence (default: '^GTTCAGAGTTCTACAGTCCGACGATC')
-e/--error [float >0..1] :	Maximum allowed error rate (no. of errors divided by the length of the matching region) (default: 0.1)
-m/--min [integer] :	Discard trimmed reads that are shorter than m. (default: 18)
-x/--max [integer] :	Discard trimmed reads that are longer than x. (default: 30)
-f/--format [string] :	Input file format; can be either 'fastq', 'fasta'. (default: 'fastq')

3. Filtering reads (filter.pl)

Description: This script takes as input a file with collapsed reads. The script then processes the reads and/or maps them to the reference library, filtering the reads that mapped to exons or other non-coding RNAs, like rRNA, snRNA, snoRNA and tRNA..

Usage: filter.pl –i reads.fa –l Rfam.fa –l Repbase.fa –v 2 –m 4

Options:

-i [string]	Collapsed reads file in fasta format
-l [string]	Library for reads filtering, users can appoint one or more libraries, and the library must be in fasta format
-v [int]	Report alignments with at most <int> mismatches. [default: 2]
-k	Keep the matched alignments
-m [int]	Number of threads to launch. [default: 1]
-q	Quiet mode.. no log/progress information to STDERR
-h	Print help message and quit

4. Mapping the collapsed reads to genome and parsing to arf format (map_parse.pl)

Description: The script processes the reads and/or maps them to the reference genome, as designated by the options given. The mapped file is then converted to an 'arf' file, which used in miRDeep2.

Usage: map_parse.pl collapse_reads.fa -g genome.fa -l 18 -n 2 -q -p 8 > reads_vs_genome.arf

Options:

-g genome	The genome file where input reads file will be mapped to
-l [int]	The seed length [default: 18]
-n [int]	Mismatches allowed in the seed [default: 2]
-m [int]	Suppress all alignments for a particular read or pair if more than <int> reportable alignments exist for it. [default: 15]
-q	Quiet mode .. no log/progress information to STDERR
-p [int]	Number of threads to use for bowtie
-h	Print help message and quit

5. Identifying known miRNA and predicting new miRNA (miRDeep2.pl)

Description: This script is modified from miRDeep2. We adjust some parameters for used in plant and add the support for multiple threads to speed up the overall procedure.

Usage: miRDeep2.pl Atshoot.fa TAIR10_genome.fa reads_vs_genome.arf ath_mature.fa none ath_hairpin.fa –d –m 20 –v –P –l 250 -

6. Target identification (CleaveLand4.pl)

Description: This script is modified from CleaveLand4. We add the support for multiple threads running and report not only the 10^{th} splice site, but also the 9^{th} and 11^{th} splice site, as multiple kinds of variant miRNA exists in plant.

Usage: CleaveLand4.pl –e GSM278370.fasta –u ath_mature.fa –n TAIR10_cdna.fasta –m 4 –t –o tplot > cleaveland.result

7. Target prediction (tapir_wrapper.pl)

Description: This script predicts the target of miRNA using a modified TAPIR. As plant transcriptome are mostly large in size, the original TAPIR will take a few days for a precise prediction of target. We add the support for multiple threads to speed up this step.

Usage: tapir_wrapper.pl –i miRNA.fa –t TAIR10_cdna.fasta –m 4 –r 0.65 –b –o predicted.result

Options:

-i [string]	miRNA file in fasta format
-t [string]	Target transcriptome file in fasta format
-o [string]	Output file
-s [float]	Score cutoff in TAPIR. [default: 4]
-r [float]	mfe ratio cutoff. [default: 0.65]
-b	Tabular report of the result.
-m [int]	Number of threads to lanuch. [default: 1]
-q	Quiet mode .. no log/progress information to STDERR
-h	Print help message and quit

8. Prioritization of predicted target (prioritization.pl)

Description: This script takes two files as input, prioritizing the predicted miRNA-target pairs according to GO similarity to identified miRNA either validated by experiment or identified from degradome data.

Usage: prioritization.pl -g go_annotation_file -v validated.csv -p predicted.csv -q

Options:

-h	Print this usage
-g [string]	File containing GO annotation
-v [string]	validated miRNA-target file in degradome data
-p [string]	Predicted miRNA-target file using TAPIR or other tools
-q	Quiet mode .. no log/progress information to STDERR

9. Differential expression analysis of known miRNA (de_for_miRNA.pl)

Description: This script does differential expression analysis of known miRNA quantified by miRDeep2. It invokes DESeq package.

Usage: de_for_miRNA.pl –s sample1_expressed_miRNA=sample1 –s sample2_expressed_miRNA=sample2

Options:

-s [string]	The format must be like <expressed_miRNA_file>=<sample> And the sample denotes the experiment condition, which can be ‘treated’, ‘untreated’ or other name. The option can exist many times for experiment with replicates.