Objective

Look for differences between two or more treatments.

Experimental design

Two or more treatment will be compared in their sRNA content.

Biological and methodological replicates will be performed. Biological reps are represented by multiple samples per treatment, methodological reps are achieved (when possible or adequate) by using multiple analysis tools for every step.

Samples= individual libraries (.fastq files)

Treatment= set of samples

For miRNA identification:

Each sample will be analyze by 2 or 3 independent programs

Grouping and condensation of replicates:

Biological reps:

Shorstack and miRPrefer analyisis can handle multiple samples at once, all biological reps per treatment are analyzed at once, the outputs will represent composite samples. One output per treatment.

miRPlant (if used) needs to be run for every individual sample, outputs then need to be condensed per treatment.

Methodological reps:

The output from Shortstack, miRPrefer and miRPlant (if used) will be then filtered to include only miRNA detected by at least two independent tools

Activities

**Initial edit of raw Fastq file**

Objective**:**

Prepare the input file for miRNA identification software

Requirements:

* Cutadapt tool (<http://journal.embnet.org/index.php/embnetjournal/article/view/200>, <http://cutadapt.readthedocs.org/en/stable/installation.html#id1>)
* Raw sRNA FastQ file
* Adapter sequence

Code:

cutadapt -m 15 -M 35 --discard-untrimmed -a adapter\_sequence -o sample\_name#\_trimmed.fastq sample\_name.fastq &

**miRNA identification**

Objective: performed miRNA identification using three different programs and condensate the results

Tasks:

1. Identify miRNAs using ShortStack

Requirements:

* ShortStack 3.3 (<http://rnajournal.cshlp.org/content/19/6/740.short>, https://github.com/MikeAxtell/ShortStack/releases/)
* sRNA FastQ file(s) from step 1
* Genome file (A\_thal\_genome.fa, TAIR10)

1. For each set of samples within a treatment, run the tool with the following parameters:

Code:

ShortStack --outdir sample\_name --bowtie\_cores 8 --readfile sample\_name#\_trimmed.fastq sample\_name#n\_trimmed.fastq -genomefile genome.fa --sort\_mem 12G &

1. Edit results files

In directory ‘Shortstack/sample\_name’:

Change name of ‘ShortStack\_D.gff3’ to ‘ShortStack\_sample\_name.gff3’

And for that file, replace ‘Chr’ for ‘chr’

sed -i 's/Chr/chr/g' ShortStack\_sample\_name.gff3

1. Identify miRNA using miR-Prefer

Requirements:

* FastX package (<http://hannonlab.cshl.edu/fastx_toolkit/download.html>)
* miR-Prefer (<http://www.ncbi.nlm.nih.gov/pubmed/24930140>, https://github.com/hangelwen/miR-PREFeR)
* sRNA FastQ file(s) from step 1
* Config\_file
* sample.list file
* Genome file (A\_thal\_genome.fa, TAIR10)

1. Prepare the files

Sample.list= For each set of samples within a treatment create flat text file listing samples, one per line

To produce the sam files:

convert to fasta

fastq\_to\_fasta -i sample\_name\_trimmed.fastq -o sample\_name\_trimmed.fasta &

run the miRPrefer read prep tool

python scripts/process-reads-fasta.py sample.list sample\_name#\_trimmed.fasta sample\_name#n\_trimmed.fasta &

perform the mapping using the miRPrefer align tool, this step produce the SAM file to be inputted into the config\_file

python scripts/bowtie-align-reads.py -f -r A\_thal\_genome.fa -t bowtie-index/ -p 8 sample\_name\_trimmed.fasta.processed &

Config\_file\_sample\_name= create a flat text file with the following settings:

PIPELINE\_PATH = absolute path to the tool

FASTA\_FILE = absolute path to genome used to produce SAM file

ALIGNMENT\_FILE = absolute path to SAM file: sample\_name\_trimmed.fasta.processed.sam

#GFF\_FILE\_EXCLUDE =

#GFF\_FILE\_INCLUDE =

PRECURSOR\_LEN = 300

READS\_DEPTH\_CUTOFF = 20

NUM\_OF\_CORE = 8

OUTFOLDER = absolute path, named as sample\_name

TMPFOLDER = /tmp/run\_name

NAME\_PREFIX = run\_prefix

MAX\_GAP = 100

MIN\_MATURE\_LEN = 18

MAX\_MATURE\_LEN = 24

ALLOW\_NO\_STAR\_EXPRESSION = Y

ALLOW\_3NT\_OVERHANG = Y

CHECKPOINT\_SIZE = 3000

1. Run the miRPrefer miRNA identification tool

python miR\_PREFeR.py -L pipeline config\_file\_sample\_name

1. Edit the result files

In outfolder ‘ sample\_name’,

Get the lines corresponding to miRNA from “sample\_name.miRNA.gff3” and write into new file

grep -v miRNA-precursor sample\_name\_miRNA.gff3 > mirprefer\_sample\_name.gff3

1. Identify miRNAs using mirPlant

This tool can only handle one sample at a time and does not multithread.

Requirements:

* mirPlant (<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4141084/>, <https://sourceforge.net/projects/mirplant/>)
* sRNA FastQ file(s) from step 1
* genome index (A. thal TAIR10)

1. Create a genome index according to the software instructions
2. Run the tool

java -jar -Xmx12G miRPlant\_command\_line.jar -l 24 -t 20 -g genome\_index\_folder sample\_name\_trimmed.fastq

1. Edit, filter and format the output

From the ‘sample\_name\_trimmed.result’ file filter out hits with scores < 0, create a gff3 file with the hits passing the filter

Incomplete solution:

awk '{if ($2 >= 0) print $3,$7,$1}' OFS="\t" sample\_name\_trimmed.result | sed -e 's/ath-/ath\_/g;s/-/\t/g' > mirplant\_sample\_name.gff3

1. If more than one sample per treatment, condense results by accepting only miRNA hits present in at least two samples (use bedtools intersect, described below)
2. Condense results from all three (or two) programs to obtain a list of candidate known and novel list of miRNAs per treatment

Requirements:

* Bedtools 2.25.0 (<http://bioinformatics.oxfordjournals.org/content/26/6/841.short>, <https://github.com/arq5x/bedtools2/releases>)
* Know miRNA gff3 file from miRBase(ath.gff3, <ftp://mirbase.org/pub/mirbase/CURRENT/genomes/ath.gff3>)
* Gff3 result files from Shorstack, miRPrefer and miRPlant (optional)

miRNA will be analyzed in two categories: known and novel miRNAs

1. Create a new gff3 file containing entries for miRNA mature locations only.

grep –v miRNA\_primary\_transcript ath.gff3 > ath\_mature\_only.gff3

For novel miRNAS:

1. Extract all candidate miRNAs with no match in the current database

bedtools intersect –a ShortStack\_sample\_name.gff3 –b ath\_mature\_only.gff3 –v –f 0.8 > ShortStack\_sample#\_novel.gff3

bedtools intersect –a mirplant\_ sample\_name.gff3 –b ath\_mature\_only.gff3 -v –f 0.8 > mirplant\_sample#\_novel.gff3

bedtools intersect –a mirprefer\_ sample\_name–b ath\_mature\_only.gff3 –f 0.8 –v > mirprefer\_sample#\_novel.gff3

1. Find novel miRNAs identify by at least two tools per treatment

bedtools intersect -a ShortStack\_sample#\_novel.gff3 -b mirplant\_sample#\_novel.gff3 mirprefer\_sample#\_novel.gff3 -c | awk '{if ($10 >= 1) print}'| cut -f -9 > novel\_1.txt

bedtools intersect -a mirplant\_sample#\_novel.gff3 -b ShortStack\_novel.gff3 -v > novel\_2.txt

bedtools intersect -a mirprefer\_ sample#\_novel.gff3 -b ShortStack\_novel.gff3 -v > novel\_3.txt

bedtools intersect -a novel\_2.txt -b novel\_3.txt > novel\_4.txt

cat novel\_1 novel\_4.txt > sample#\_novel\_miRNAs.gff3

rm novel\_\*.txt # remove intermediate files

1. Report number of novel miRNA per treatment # check empty lines at the end of files

wc –l sample#\_novel\_miRNAs.gff3 # check empty lines at the end of files, may produce incorrect results

wc –l sample#n\_novel\_miRNAs.gff3

1. Determine shared and specific novel miRNAs across treatments

bedtools intersect -a control\_miRNA\_novel.gff3 -b sample#\_miRNA\_novel.gff3 sample#n\_miRNA\_novel.gff3 –c | awk '{if ($10 >= [# of treatments -1]) print > shared\_all\_treatments\_\_novel\_miRNA.gff3

* novel miRNAs shared by specific combinations of treatments:

control/treatment #1 (not exclusive): A ∩ B

bedtools intersect -a control\_miRNA\_novel.gff3 -b sample#\_miRNA\_novel.gff3 sample#n\_miRNA\_novel.gff3 -wa -wb -names sample# sample#n | sort -k10,10 | grep sample# | cut -f 11- > sample#\_control\_total \_novel\_miRNAs.gff3

control/treatment #2 (not exclusive): A ∩ C

bedtools intersect -a control\_miRNA\_novel.gff3 -b sample#\_miRNA\_novel.gff3 sample#n\_miRNA\_novel.gff3 -wa -wb -names sample# sample#n | sort -k10,10 | grep sample#n | cut -f 11- > sample#n\_control\_total\_novel\_miRNAs.gff3

* control/treatment #1 exclusive: (A ∩ B) – (A ∩ B ∩ C)

bedtools intersect –a sample#\_control\_total \_novel\_miRNAs.gff3 –b sample#n\_control\_total\_novel\_miRNAs.gff3 –v > sample#\_control\_specific\_novel\_miRNAs.gff3

control/treatment #2 exclusive: (A ∩ C) – (A ∩ B ∩ C)

bedtools intersect –a sample#n\_control\_total \_novel\_miRNAs.gff3 –b sample#\_control\_total\_novel\_miRNAs.gff3 –v > sample#n\_control\_specific\_novel\_miRNAs.gff3

treatment#1/treatment#2 (not exclusive): (B ∩ C)

bedtools intersect –a sample#\_miRNA\_novel.gff3 sample#n\_miRNA\_novel.gff3 > sample#\_sample#n\_total\_novel\_miRNAs.gff3

treatment#1/treatment#2 exclusive: (B ∩ C) – (A ∩ B ∩ C)

bedtools intersect –a sample#\_sample#n\_total\_novel\_miRNAs.gff3 –b shared\_all\_treatments\_miRNA.gff3 –v > sample#\_sample#n\_specific\_novel\_miRNAs.gff3

* treatment specific miRNAs

control specific: A –B-C

bedtools intersect –a control\_miRNA\_novel.gff3 -b sample#\_miRNA\_novel.gff3 sample#n\_miRNA\_novel.gff3 –v > control\_specific\_novel\_miRNAs.gff3

treatment #1 specific miRNAs: B-A-C

bedtools intersect –a sample#\_miRNA\_novel.gff3 -b control\_miRNA\_novel.gff3 sample#n\_miRNA\_novel.gff3 –v > sample#\_specific\_novel\_miRNAs.gff3

treatment #2 specific miRNAs: C-A-B

bedtools intersect –a sample#n\_miRNA\_novel.gff3 -b control\_miRNA\_novel.gff3 sample#\_miRNA\_novel.gff3 –v > sample#n\_specific\_novel\_miRNAs.gff3

1. Report results in Venn diagram (temporal: Venny, <http://bioinfogp.cnb.csic.es/tools/venny/>, BioVenn, <http://www.cmbi.ru.nl/cdd/biovenn/>, to be implemented in the tool: <http://stackoverflow.com/questions/8713994/venn-diagram-proportional-and-color-shading-with-semi-transparency>)

For known miRNA:

1. Find known miRNAs identify by at least two tools per treatment

bedtools intersect -a ath\_mature\_only.gff3 -b ShortStack\_sample\_name.gff3 mirplant\_ sample\_name.gff3 mirprefer\_ sample\_name \_miRNA.gff3 -f 0.90 –c | awk '{if ($10 >= 2) print}' | cut -f -9 > Sample\_miRNAs\_known.gff3

1. Report number of known miRNAs per treatment

wc –l Sample#\_miRNAs\_known.gff3

wc –l Sample#n\_miRNAs\_known.gff3

1. Determine shared and specific known miRNAs across treatments (Same strategy as for novel miRNA)

* Shared miRNA in all treatments: A ∩ B ∩ C

bedtools intersect -a control\_miRNA\_known.gff3 -b sample#\_miRNA\_known.gff3 sample#n\_miRNA\_known.gff3 –c | awk '{if ($10 >= [# of treatments -1]) print > shared\_all\_treatments\_known\_miRNA.gff3

* known miRNAs shared by specific combinations of treatments:

control/treatment #1 (not exclusive): A ∩ B

bedtools intersect -a control\_miRNA\_known.gff3 -b sample#\_miRNA\_known.gff3 sample#n\_miRNA\_known.gff3 -wa -wb -names sample# sample#n | sort -k10,10 | grep sample# | cut -f 11- > sample#\_control\_total \_known\_miRNAs.gff3

control/treatment #2 (not exclusive): A ∩ C

bedtools intersect -a control\_miRNA\_known.gff3 -b sample#\_miRNA\_known.gff3 sample#n\_miRNA\_known.gff3 -wa -wb -names sample# sample#n | sort -k10,10 | grep sample#n | cut -f 11- > sample#n\_control\_total\_known\_miRNAs.gff3

* control/treatment #1 exclusive: (A ∩ B) – (A ∩ B ∩ C)

bedtools intersect –a sample#\_control\_total \_known\_miRNAs.gff3 –b sample#n\_control\_total\_known\_miRNAs.gff3 –v > sample#\_control\_specific\_known\_miRNAs.gff3

control/treatment #2 exclusive: (A ∩ C) – (A ∩ B ∩ C)

bedtools intersect –a sample#n\_control\_total \_known\_miRNAs.gff3 –b sample#\_control\_total\_known\_miRNAs.gff3 –v > sample#n\_control\_specific\_known\_miRNAs.gff3

treatment#1/treatment#2 (not exclusive): (B ∩ C)

bedtools intersect –a sample#\_miRNA\_known.gff3 sample#n\_miRNA\_known.gff3 > sample#\_sample#n\_total\_known\_miRNAs.gff3

treatment#1/treatment#2 exclusive: (B ∩ C) – (A ∩ B ∩ C)

bedtools intersect –a sample#\_sample#n\_total\_known\_miRNAs.gff3 –b shared\_all\_treatments\_miRNA.gff3 –v > sample#\_sample#n\_specific\_known\_miRNAs.gff3

* treatment specific miRNAs

control specific: A –B-C

bedtools intersect –a control\_miRNA\_known.gff3 -b sample#\_miRNA\_known.gff3 sample#n\_miRNA\_known.gff3 –v > control\_specific\_known\_miRNAs.gff3

treatment #1 specific miRNAs: B-A-C

bedtools intersect –a sample#\_miRNA\_known.gff3 -b control\_miRNA\_known.gff3 sample#n\_miRNA\_known.gff3 –v > sample#\_specific\_known\_miRNAs.gff3

treatment #2 specific miRNAs: C-A-B

bedtools intersect –a sample#n\_miRNA\_known.gff3 -b control\_miRNA\_known.gff3 sample#\_miRNA\_known.gff3 –v > sample#n\_specific\_known\_miRNAs.gff3

1. Report results in Venn diagram (temporal: Venny, <http://bioinfogp.cnb.csic.es/tools/venny/>, BioVenn, <http://www.cmbi.ru.nl/cdd/biovenn/>, to be implemented in the tool: <http://stackoverflow.com/questions/8713994/venn-diagram-proportional-and-color-shading-with-semi-transparency>)