The UEA sRNA Toolkit:
A User Guide for the Perl Implementation

Martin Lott
Daniel Mapleson
Irina Mohorianu
Vincent Moulton
Simon Moxon
Frank Schwach
Contact: srna-tools@cmp.uea.ac.uk

June 15, 2011

Contents

Preface

RNA silencing [13] is a complex, highly-conserved, transcriptional and post-transcriptonal mechanism [28] that tunes gene expression. It was originally studied as a defense mechanism against pathogens in plants [4] and later studied extensively due to its ability to regulate cancer-related genes [29]. RNA silencing is mediated by molecules known as small RNAs (sRNAs) which are reviewed in detail in [26].

Recently, high-throughput sequencing has revolutionised the field of sRNA biology by making possible the identification and profiling of sRNAs in the cell. The constantly increasing number of reads facilitates the characterisation of the different pathways and reveals new classes of sRNAs. In order to process the large amount of sequences obtained from high throughput experiments, toolkits, such as the UEA sRNA toolkit [21], were developed. The UEA sRNA toolkit is available as a hosted service at `http://srna-tools.cmp.uea.ac.uk/`.

However, as high-throughput sequencing devices evolve and more reads are produced, submitting the sequence data across the internet is becoming more problematic. The UEA sRNA toolkit limits the input file size to manage UEA server resources and network bandwidth.

To overcome these restrictions, the UEA bioinformatics group has packaged an open-source, stand-alone version of the toolkit. Users can download the toolkit and run the tools locally on Unix-based desktops or servers, mitigating the data transfer limitations to a hosted service. This document describes the tools, discusses their purpose and provides details on how to use them.

Audience

This document and the toolkit are intended for bioinformaticians, who should:

- Have a working knowledge of Linux/Unix and running tools from the command line.

- Be familiar with small RNAs and their subtypes.
- Have basic knowledge of high throughput sequencing devices and the kind of output they produce, particularly FASTQ and FASTA format files.

Document Organisation

This user guide is organised as follows:

- Chapter 1, "Installation", contains details on what is provided by the UEA, what the system pre-requisities are and how to install the toolkit.
- Chapter 2, "Quickstart", contains a brief description of the tools and details how to run the tools from the command line.
- Chapter 3, "Tools", contains a detailed description of each tool in the sRNA toolkit.
- Chapter 4, "Troubleshooting and FAQ", contains answers to several commonly encountered error messages, issues and ways to resolve them.

Notational Convention

This user guide uses typefaces to identify the characteristics of text. The general-purpose typefaces and characteristics they imply are described in this table:

- `Monospaced`
  used for paths, filenames, commands and source code.
- `url`
  used for urls.

When describing tool parameters the typefaces are as follows:

- path parameters specify the location of a file or directory.
  e.g. `/local/usr/myself/tools`
- string
  e.g. `--tool adaptor`, `--adaptor_sequence_3 AGCTGGCTTC`
- numeric:integer parameters have a default value and a range of allowed values. If the input value is outside the range, the default value is automatically assigned.
  e.g. `--minsize 3`

- numeric:real parameters have a default value and a range of allowed values. If the input value is outside the range, the default value is automatically assigned.
  e.g. `--min_energy -12.5`

- boolean parameters have default value 1 (TRUE) if they are required parameters and 0 (FALSE) if they are optional parameters.
  e.g. `--trrna` [true]

If the input type does not match the parameter type, an error is produced and the execution of the script is halted.
e.g. minsize (numeric parameter) `--minsize AGTC` [ERROR].
However, in the case of a numeric mismatch (i.e. real value instead of integer value) displays no warning, all values are rounded up to nearest integer number (e.g. 18.1 → 19).

The stand-alone sRNA toolkit is a Unix-based only product. Therefore directory paths in this guide use a forward slash character (/) as the separator between a directory name and the name of a subdirectory or file in that directory. For example, the absolute path `/arabidopsis/srna_reads` indicates the `srna_reads` subdirectory of a directory named `arabidopsis` mounted off the root directory on the file system.

Disclaimer

The UEA bioinformatics group is not able to offer support for this version of the toolkit. However, a new version will be released that will offer enhancements such as additional tools, platform independence, improved performance, reduced hardware requirements and improved usability.

Acknowledgements

The sRNA toolkit was developed with support from the Biotechnology and Biological Sciences Research Council (BBSRC), `http://www.bbsrc.ac.uk`, grants BB/E004091/1 and BB/I00016X/1, and the SIROCCO consortium `http://www.sirocco-project.eu`.

Chapter 1

Installation

The toolkit can be downloaded and deployed onto machines running a Linux distribution. The deliverables simply need to be unpacked into a directory of the user's choosing, denoted in this document as `$INSTALL_PATH`. The user may find it helpful to ensure that the `srna-tools.pl` perl script is on the path. In addition, there are a number of dependencies that must be properly installed onto the system for the toolkit to function properly. The remainder of this chapter describes the deliverables provided by the UEA as well as the toolkit's dependencies that must be installed onto the system.

1.1 Deliverables

The toolkit is split into three archives that can be downloaded from

`srna-workbench.uea.ac.uk/perl_main_page.html`

and are also collectively available as a CD `iso` image.

The archive files are names as follows:

`srna-tools-cli.zip`	The sRNA toolkit software
`srna-tools-usr-local-bin.zip`	Software dependencies
`srna-tools-example.zip`	Example files
`srna-tools.iso`	CD iso image

1.2 Dependencies

1.2.1 Perl Packages

The toolkit is available for machines running a Linux distribution. For Debian based distributions the following packages, and their dependencies, are required:

- `bioperl`

- `libtemplate-perl`
- `libconfig-auto-perl`
- `libexception-class-trycatch-perl`
- `libmail-sendmail-perl`
- `libyaml-tiny-perl`

On Debian distributions, for example, on Ubuntu Linux, `BioPerl` and its dependencies can be installed with the command

`sudo apt-get install bioperl.`

For non-Debian distributions it is possible to build equivalent packages (contact with your local systems administrator).

1.2.2 Required Binaries

In addition to the Perl package dependencies,the toolkit requires a number of executable programs such as PatMaN [22] and Vienna [14]. These are provided in the `srna-tools-usr-local-bin.zip` archive which should be extracted and copied to, for example, `usr/local/bin` or alternatively to some location pointed to by your path variables. If you do not have root permissions, please contact your local systems administrator.

1.2.3 Configuration

Before using the toolkit you must edit the configuration file `$INSTALL_PATH/config/application.conf` and insert the full path of the directory where the toolkit resides on your system (see figure 1.1). This will enable the different parts of the toolkit to find each other.

```
56  ################################################
57  # Paths to directories on server and cluster
58  # where we put jobs into the queue, store
59  # genome data and third-party binaries.
60  # All paths are relative to the "doc root"
61  # i.e. the srna-tools directory
62  ################################################
63
64  # path to srna-tools "root directory" on the
65  # cluster backend and the server. The server root
66  # could be determined dynamically at run time but
67  # with this set up we can share resources between
68  # web apps by pointing them to the same root dir.
69  # It also creates a more unified interface to path data
70  root_dir => {
71    server => '/local/scratch/martin/srna-tools-cli/',
72  },
73
74  # this is where jobs are send to get queued,
75  # not the job storage area on the server
76  # (job_dir_server).
77  # We give a value for environmet "server"
78  # just for instant jobs where storage =
79  # execution dir.
80  queue_job_dir => {
81    server => '/jobs/',
82  },
```

Figure 1.1: Configuring the config file `application.conf`.

Chapter 2

Quickstart

2.1 Summary

In this document the tools are referred to and described in the following order:

Low-Level Tools (applied on raw data, e.g. FASTA sequences):

(1) **Sequence file pre-processing** tool by S. Moxon
Converts read files from FASTQ to FASTA format and removes adaptor sequences making the input file ready for use by other tools.

(2) **Filter** tool by F. Schwach
Filters sRNA sequence files in FASTA format according to user defined criteria, e.g. genome mapping reads, specific size class, t/rRNA mapping reads.

(3) **miRCat** tool − miRNA Categoriser by S. Moxon
Predicts new miRNAs from high throuput sRNA sequencing data presented as a redundant FASTA file.

(4) **miRProf** tool − known miRNA expression profiler by F. Schwach
Determines the expression profile of sRNAs (from a non-redundant FASTA file) that match known miRNAs from miRBase [15].

(5) **RNA hairpin folding and annotation** tool by F. Schwach
Produces the secondary structure of a long RNA sequence and annotates it by highlighting up to 20 short sequences on the resulting structure.

High-Level Tools (used for in depth data analysis):

(6) **FiRePat** tool − Finding Regulatory Patterns by I. Mohorianu
Identifies (positively and negatively) correlated expression profiles of sRNAs / sRNA producing loci, and genes. Receives as input two CSV files containg expression values in different samples.

(7) **SiLoCo** tool − siRNA locus comparison by F. Schwach
Finds genomic sRNA producing loci by abundance and relative position of sRNAs mapped to the reference genome [18]. The sRNA files are required in FASTA format, redundant form.

(8) **SiLoMa** tool − siRNA locus mapper by F. Schwach
Maps sRNAs (input given in FASTA format, redundant form) to a reference sequence and produces a genome browser image. The location and strand of each sRNA is represented with an arrow, and the abundance of the sRNA is proportional to the thickness of the arrow.

(9) **ta-siRNA prediction** tool by S. Moxon
Identifies ta-siRNA loci by computing the probability of phasing being significant based on a hypergeometric distribution [7].

(10) **Plant target prediction** tool by S. Moxon
Using a FASTA file containing sRNAs in non-redundant form and a FASTA file containing pairs (sRNA,transcript) are predicted based on the rules suggested in [1] and [25].

Both plants and animals use sRNAs to regulate gene expression. However, some sRNA types may not be present in both plants and animals, e.g. trans-acting short interfering RNAs (ta-siRNAs) are plant specific siRNAs and piRNAs are animal specific siRNAs. Also, some sRNA types, such as microRNA (miRNAs), while having a similar biogenesis, adjust the gene expression in slightly different ways in plants and animals [5, 17]. For this reason, some tools in the toolkit are specific for plant data sets, as shown in the following table:

Tool	Animal Data sets	Plant Data sets
Pre-processing	√	√
Filter	√	√
miRCat	√	√
miRProf	√	√
RNA folding	√	√
FiRePat	√	√
SiLoCo	√	√
SiLoMa	√	√
ta-siRNA prediction	X	√
Plant target prediction	X	√

In addition to specifying the input files and output directories, you need the following:

Tool	Sample Data	Species Data
Pre-processing	1× FASTA sRNA file	N/A
Filter	1× FASTA sRNA file	1× FASTA Genome
miRCat	1× FASTA sRNA file	1× FASTA Genome
miRProf	1× FASTA sRNA file	1× miRBase DB name
RNA folding	1× FASTA sRNA file	1× FASTA Sequence
FiRePat	2× CSV files with expression levels	N/A
SiLoCo	2× FASTA sRNA file	1× FASTA Genome
SiLoMa	1× FASTA sRNA file	1× FASTA Genome
ta-siRNA prediction	1× FASTA sRNA file	1× FASTA Genome
Plant target prediction	1× FASTA sRNA file	1× FASTA Transcriptome

Given the size of the genomes, these are not distributed with the toolkit.
Frequently used genomes can be downloaded from the following URLs:

- *Arabidopsis Thaliana*
  `ftp://ftp.arabidopsis.org/home/tair/Genes/TAIR9_genome_release/TAIR9_chr_All.fas`
- *Solanum Lycopersicum*
  `http://solgenomics.net/genomes/Solanum_lycopersicum/index.pl`
- *Oryza Sativa*
  `http://rice.genomics.org.cn/rice/link/download.jsp`
- *Human*
  `http://hgdownload.cse.ucsc.edu/downloads.html`
- *Mouse*
  `http://hgdownload.cse.ucsc.edu/downloads.html`

- *Drosophila*
  `http://www.fruitfly.org/sequence/download.html`

The running times, for each of the tools, on a Dual Quad-core Intel Xeon 2.50GHz (L5420) with 32GB RAM Linux server are:

Tool	Plant [1]		Animal [2]	
	Runtime (mins)	Memory(Mb)	Runtime (mins)	Memory(Gb)
Pre-processing	5	64	10	64
Filter	7	64	14	64
miRCat	125	256	150	256
miRProf	7	64	11	64
RNA folding	instant	64	instant	64
FiRePat [3]	instant	64	instant	64
SiLoCo	9	64	15	64
SiLoMa	8	64	11	64
ta-siRNA prediction	11	64	N/A	N/A
Plant target prediction	> 4days	1000+	N/A	N/A

Note: The run-times were computed using the default values for each too. Default values are intended to filter heavily the sRNA reads input, which reduces run-time.

The general format for calling each tool is:

```
srna-tools.pl --tool $NAME_OF_THE_TOOL [$OPTIONS_FOR_TOOL]
```

The help for each tool is available using the following command:

```
srna-tools.pl --tool $NAME_OF_THE_TOOL --help
```

2.2 Example of a Bioinformatic Analysis - Tutorial.

Given FASTA file (`files/GSM118373_Rajagopalan_leaf.fa`), a typical analysis would start with removing the adaptors. If instead of a FASTA file a FASTQ file is provided, the first step, also included in the `adaptor` tool, is converting the FASTQ file to FASTA format. The adaptor removal tool [4] is invoked with the following command:

```
srna-tools.pl --tool adaptor --adaptor_sequence_3 TCGT
--out output/a --srna_file files/GSM118373_Rajagopalan_leaf.fa
```

This generates a file `srnas_adapters_removed.fa` in `output/a`. Next, to filter all sequences that do not map to the genome and are outside a certain size range, the Filter tool is used. The size range filter can be used, for example, to focus the analysis on putative miRNA candidates.

```
srna-tools.pl --tool filter --srna_file
output/a/srnas_adapters_removed.fa --out output/f
--genome data/arabidopsis.fa --make_nr
--maxsize 26 --minsize 16
```

The results of the filter tool are contained in a zip archive which can be unzipped using the command `unzip filter_results.zip` in the directory `output/f`. The file `MyJob_filtered.fasta` contains the filtered sequence data in FASTA format. Next, to predict novel miRNA candidates, miRCat can be used as follows:

```
srna-tools.pl --tool mircat --genome data/arabidopsis.fa
--srna_file files/GSM118373_Rajagopalan_leaf.fa --out output/m
--genomehits 5 --hit_dist 100 --maxgaps 4
--max_overlap_length 50 --max_percent_unpaired 60
--max_unique_hits 4 --maxsize 24 --min_abundance 6 --min_energy -10.0
--min_gc 20 --min_hairpin_len 80 --min_paired 25 --minsize 19
--no_complex_loops --percent_orientation 80 --pval 0.2
--trrna --window_length 100
```

miRCat identifies both old and new miRNAs and creates a `.zip` archive containing the annotations of each hairpin for the miRNA candidates (structures.pdf) constructed using the hairpin annotation tool and the Vienna package [14].

[4] This example is for illustration purposes only. The `GSM118373_Rajagopalan_leaf.fa` file contains Illumina sequences with adaptors removed and is already in FASTA format.

Next, to visualise other hairpins, produced using miRCat, the RNA folding tool can be used. The input for this tool consists of the hairpin sequence and the miRNA/miRNA* which will be highlighted on the secondary structure.

```
srna-tools.pl --tool hp_tool --longSeq files/hairpin.fa --shortSeqs
files/mirna.fa --out output/h
```

In order to determine the expression of each known miRNA in the sample miRProf can be used. The results from miRProf are comparable across samples and hence miRProf is normally run once on each sample and then the results are combined.

```
srna-tools.pl --tool mirprof --mirbase_db plant_mature
--out output/mp --srna_file output/f/MyJob_filtered.fasta
--keep_best --maxsize 26 --minsize 16 --mismatches 2
```

Finally the targets for the new and old miRNAs can be checked using the target prediction tool:

```
srna-tools.pl --tool target --out output/t --pasted_srnas '>a
GCTTCTATCTTTTTCTTTCGTGCT' --transcriptome arabidopsis.fa
```

Besides identifying miRNAs we can identify all possible sRNA loci (using SiLoCo) and visualise the read distributions (using SiLoMa).

```
srna-tools.pl --tool siloco --genome data/arabidopsis.fa
--out output/si --sample_name1 S1 --sample_name2 S2
--srna_file1 files/GSM118373_Rajagopalan_leaf.fa
--srna_file2 files/GSM154370_Carrington_col0_leaf.fa
```

```
srna-tools.pl --tool siloma --genome data/arabidopsis.fa
--out output/sm --srna_file files/GSM118373_Rajagopalan_leaf.fa
--pasted_seq TAAGCTATATAGGGGGGT --region_chrom 2
--region_start 39148 --region_end 39445
```

After visually inspecting few genome browser figures we may wish to determine the ta-siRNA loci present in a given plant dataset. We can identify these loci using the ta-siRNA prediction tool.

```
srna-tools.pl --tool phasing --genome data/arabidopsis.fa
--out output/p --srna_file files/GSM118373_Rajagopalan_leaf.fa
```

Using MirProf we have obtained the expression profiles of the known miRNAs in the set. After using SiLoCo, we also have the expression levels of the loci in at most two samples. If similar gene data is available (expression levels of genes measured in similar conditions) we may use FiRePat to identify co-anti regulated pairs using both the miRNA expression levels and the loci expression levels.

```
srna-tools.pl --tool firepat --out output/fp
--gene_file files/firepat_test150_genes.csv
--srna_file files/firepat_test150_srna_loci.csv
```

The co and anti-correlated pairs formed with miRNAs will help us decide which of the targets predicted by the target prediction tool are more likely to be real, and these targets can be later validated in biological experiments. The co- and anti-regulated pairs formed with loci will provide a general overview of interactions between sRNA loci and gene at genome level.

Chapter 3

Tools

3.1 Sequence File Pre-Processing Tool

Sequencing devices produce reads with adaptor sequences at either end of the read. This tool removes those adaptor sequences making the input file ready for use by other tools in the toolkit.

The tool is able to process a FASTQ or a FASTA file. If a FASTQ file, as produced by a sequencing device, is provided as input, then this tool first converts it to FASTA, before the adaptors are removed. It can also handle zipped and gzipped archives containing files of the above mentioned formats.

Next, 5' (optional) and 3' (required) adaptors are removed, as specified below. The 5' adaptor is optional, because not all sequencing devices include it in the resulting reads. For example, for 454 datasets and conventional cloning and capillary sequencing, both the 5' and the 3' adaptors are included in the input file. In contrast, Solexa/Illumina reads start at the first base of the sRNA and contain only the 3' adaptor (see figure 3.1). The tool only looks for exact matches to the adaptor sequence(s) so it will not remove adaptors containing mismatches. For this reason it is often preferable to provide a truncated version of the adaptor sequence as input. For example, the first 8nt of the adaptor sequence are sufficient for 3' adaptor matching or the last 8nt of the adaptor sequence are sufficient for 5' adaptor matching.

Parameters:

- Required
  - adaptor_sequence_3 The 3' adaptor sequence.
  - srna_file The location of the sRNA file in FASTQ or FASTA format.

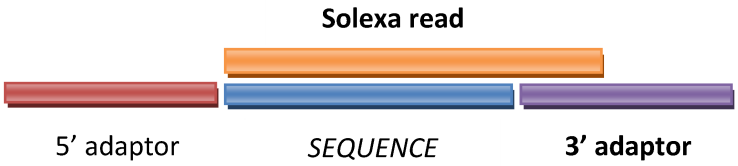

Figure 3.1: Read with adaptors. A Solexa/Illumina read starts at the first base of the sRNA and contains only the first part of the 3' adaptor.

  - out The path to the output directory.

- Optional
  - adaptor_sequence_5 The 5' adaptor sequence.
  - allow_rev_comp If used, matches to the reverse complement of adaptor sequences are allowed. This parameter is only required for classical capillary sequencing, where the orientation of the clone relative to the sequencing primers is not known.
  - minsize The minimum length of the read.
    ($16 \leq minsize \leq 35$, default $minsize = 18$)
  - maxsize The maximum length of a read.
    ($16 \leq maxsize \leq 35$, default $maxsize = 25$).

Example

```
srna-tools.pl --tool adaptor --adaptor_sequence_3 TCGT
--srna_file files/GSM118373_Rajagopalan_leaf.fa --out output/a
--adaptor_sequence_5 TGGA --allow_rev_comp --minsize 20 --maxsize 25
```

3.2 Filter Tool

This tool filters sRNA sequence files in FASTA format according to user-defined criteria. It generates a FASTA file with sequences that passed the filter(s). In addition, a comma-separated-values (csv) table is produced, which summarises the total number of sequences after each filtering step and the distribution of their lengths.

The sequences can be filtered based on their length using the optional parameters `--minsize` and `--maxsize`. This will clean the input and prepare the reads for subsequent steps, like miRNA prediction.
Next, the low complexity sequences are filtered out. This tool defines a sequence as having low complexity if it contains at most two distinct nucleotides.

In addition, the tool can filter transfer and ribosomal RNAs (t/rRNAs) using the sequences present in the `$INSTALL_PATH/data/t_and_r_RNAs.fa` file. This filtering is commonly conducted on sRNA datasets, since reads mapping to tRNA and rRNA might be degradation products. The file contains t/rRNAs obtained from RFAM, version 10 (Jan-2010) [12, 10], the Genomic tRNA Database [6] and EMBL [16], release 95 (09-Jun-2008). The file can be replaced with any FASTA file containing t/rRNAs sequences.

Note: the tool might remove some sequences that are not t/rRNA simply due to a random match to an annotated t/rRNA in another species which are present in the file `data/t_and_r_RNAs.fa`.
Then, if the user provides a corresponding genome, the sequences can be partitioned into genome-matching and not-genome-matching. Usually the reads that do not map to the genome are considered sequencing errors or minor contamination, and are generally discarded. Another application for genome filtering is the analysis of reads produced from virus-treatment experiments. For example, these sRNA reads can be partitioned into three categories: reads identified in both the host and viral genome; reads unique only to the host genome; and reads unique only to the viral genome. This can be achieved by running the filter tool several times with the different genomes.

Parameters:

- Required
  - srna_file The location of the sRNA file in FASTA format.
  - out The path to the output directory.
- Optional

- genome A FASTA file containing a genome. Sequences can be filtered according to whether they match or not the genome. By default only sRNAs matching the genome are kept.
- make_nr If specified, the resulting FASTA file is made non-redundant. The file will be smaller as there is only one entry per unique sequence. **Do not use this option if you wish to use the filtered list with other tools in this toolkit**.
- maxsize The maximum length of a read. ($16 \leq maxsize \leq 35$, default $maxsize = 25$).
- minsize The minimum length of a read. ($16 \leq maxsize \leq 35$, default $minsize = 18$).
- trrna If defined, all reads matching a sequence in `$INSTALL_PATH/data/t_and_r_RNAs.fa` are removed.
- trrna_sense By default, both sense and anti-sense t/rRNA matches are accepted. If defined, only sense matches are removed.
- discard_genome_matching Rather than keeping genome matches only sequences that don't match the genome are retained.

Example

```
srna-tools.pl --tool filter --srna_file files/GSM118373_Rajagopalan_leaf.fa
--out output/f --genome data/arabidopsis.fa --make_nr
--maxsize 26 --minsize 16 --trrna --trrna_sense
```

3.3 miRCat Tool

miRNAs are a well-studied class of sRNAs [24], that are generated from a single stranded RNA (ssRNA) that forms a stable, partially double stranded stem-loop structure (hairpin) [9]. miRCat [21] predicts miRNAs from high-throughput sRNA sequencing data without requiring a putative precursor sequence as these will be identified by the program.

The tool receives as input two FASTA files: the sRNA sequence file, with adaptors removed; and a corresponding genome for the organism that is being studied. Before processing, miRCat maps the sRNA sequences to the genome, using PatMaN [22] (PatMaN is provided in the dependencies archive for the toolkit).

Once the sequences are mapped to the input genome, miRCat will look for genomic regions covered with sRNAs (sRNA loci), containing reads with abundance at least five (this threshold can be adjusted using the `--min_abundance` parameter). These loci must match certain criteria (see figure 3.2):

- Loci must contain no more than four non-overlapping sRNAs.
- Each sRNA in a locus must be no more than 200nt away from it's closest neighbor (this threshold can be adjusted using the `--hit_dist` parameter).
- At least 90% of sRNAs in a locus must have the same orientation (this threshold can be adjusted using the `--percent_orientation` parameter).

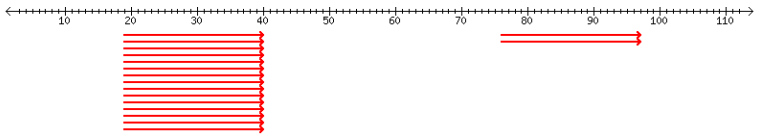

Figure 3.2: SiLoMa output showing miR164.

Once a list of loci has been produced, they are further analyzed in order to find likely miRNA candidates:

- The most abundant sRNA read within a locus is chosen as the likely miRNA.
- Flanking sequences surrounding this sRNA are extracted from the genome using varying window lengths.

- Each sequence window is then folded using RNAfold, producing a secondary structure for the putative miRNA (see figure 3.3).
- miRCat then trims the secondary structure and computes discriminative features useful for classifying miRNAs. The features are:
  - The number of consecutive mismatches between miRNA and miRNA* must be no more than 3.
  - The number of paired nucleotides between the miRNA and the miRNA* must be at least 17 of the 25 nucleotides centered around the miRNA.
  - The hairpin must be at least 75nt (for plants) or 50nt (for animals) in length.
  - The percentage of paired bases in the hairpin must be at least 50% of base-pairs in the hairpin (this threshold can be adjusted using the `--max_percent_unpaired` parameter).
- The hairpin with the lowest minimum free energy (MFE) from the sequence windows is then chosen as the precursor miRNA (pre-miRNA) candidate
- The pre-miRNA candidate is then tested using randfold using a default cutoff of 0.1 (this threshold can be adjusted using the `--pval` parameter).

Figure 3.3: RNA fold output showing miR164 precursor.

Parameters:

- Required
  - genome The location of the genome file in FASTA format.
  - srna_file The location of the sRNA file in FASTA format.
  - out The path to the output directory.
- Optional
  - genomehits The maximum number of genome hits.
    ($1 \leq genomehits$, default $genomehits = 16$).

- hit_dist The maximum distance between consecutive hits on the genome.
  ($0 \leq hit\_dist$, default $hit\_dist = 200$).
- max_gaps The maximum number of consecutive unpaired bases in miRNA region.
  ($0 \leq max\_gaps \leq 5$, default $max\_gaps = 3$).
- max_overlap_length The maximum total length (nt) of overlapping sRNAs.
  ($30 \leq max\_overlap\_length$, default= 70).
- max_percent_unpaired The maximum percentage of unpaired bases in hairpin. ($1 \leq max\_percent\_unpaired \leq 100$, default $max\_percent\_unpaired = 50$).
- max_unique_hits The Maximum number of non-overlapping hits in a locus. ($1 \leq max\_unique\_hits$, default $max\_unique\_hits = 3$).
- maxsize The maximum length of a miRNA. ($18 \leq maxsize \leq 24$, default $maxsize = 22$).
- min_abundance The minimum sRNA abundance. ($1 \leq min\_abundance$, default $min\_abundanc = 5$).
- min_energy The minimum free energy of the hairpin. Must be ≤ 0. Default $= -25$.
- min_gc The Minimum percentage of G/C in miRNA (must be ≥ 1 and ≤ 100. Default $= 10$).
- min_hairpin_len The Minmum length of hairpin (nt) (must be ≥ 50. Default $= 75$).
- min_paired The Minimum number of paired bases in miRNA region (Must be ≥ 10 and ≤ 25. Default $= 17$).
- minsize The Minimum sRNA size (Must be ≥ 18 and ≤ 24. Default $= 20$).
- no_complex_loops If defined, the hairpins with complex loops are removed.
- percent_orientation The percentage of sRNAs in locus that must be in the same orientation ($1 \leq percent\_orientation \leq 100$, default $percent\_orientation = 90$).
- pval The p-value. ($0.0 \leq pval \leq 1.0$, default $= 0.1$).
- trrna If defined, sRNAs matching sequences in the FASTA t/r RNA file (`data/t_and_r_RNAs.fa`) will be removed.
- window_length The window length. ($40 \leq window\_length \leq 400$, default $window\_length = 150$).

Example

```
srna-tools.pl --tool mircat --genome data/arabidopsis.fa
--srna_file files/GSM118373_Rajagopalan_leaf.fa --out output/m
--genomehits 5 --hit_dist 100 --maxgaps 4
--max_overlap_length 50 --max_percent_unpaired 60
--max_unique_hits 4 --maxsize 24 --min_abundance 6 --min_energy -10.0
--min_gc 20 --min_hairpin_len 80 --min_paired 25 --minsize 19
--no_complex_loops --percent_orientation 80 --pval 0.2
--trrna --window_length 100
```

miRCat returns the results as a .zip file containing the following output files:

- A .csv file showing all predicted miRNA loci - it displays the following information about each predicted miRNA:
  - Chromosome
  - Start position
  - End position
  - Strand/orientation
  - Abundance (number of times sequenced in high-throughput dataset)
  - Sequence of predicted mature miRNA
  - Representative sequence accession from input dataset
  - Length of predicted mature miRNA
  - Number of matches to genome
  - Length of predicted precursor hairpin sequence
  - G/C % content of hairpin sequence
  - Minimum free energy (MFE) of predicted hairpin sequence
  - Adjusted MFE

    $$MFE_a = \frac{MFE}{length_{hairpin}} \cdot 100.$$

    Shows MFE per 100nt making results comparable.
  - randfold p-value
  - miRNA* shows predicted miRNA* sequence(s), if any, along with abundance in input dataset shown in brackets
- A text file containing predicted miRNA precursor sequences and structures (in dot-bracket notation)
- A pdf file containing predicted miRNA precursor structures with miRNA (and miRNA* if present) highlighted

- A FASTA format file of all predicted mature miRNA sequences

Suggested parameters for the animal and plant version of miRCat are listed below:

Parameter	Plant	Animal
–extend	100	40
–withrandfold	true	true
–min_paired	17	17
–min_hits	5-20	5-20
–max_gaps	3	3
–max_genome_hits	16	16
–min_length	20	21
–max_length	22	23
–min_hairpin_len	75	50
–hit_dist	200	50
–pval	0.1	0.1
–no_complex_loops	false	true
–max_unpaired	60	40
–orientation	80	80

The following parameters can be left as default: `--minenergy`, `--max_overlap_length`, `--min_gc`.

3.4 miRProf Tool

This tool determines the expression levels of sRNAs that match known miRNAs. The expression level of a sRNA represents the number of occurences of the sequence in the sample. miRProf allows the user to group miRNAs according to different criteria e.g. organisms and/or family.

miRProf filters the sequences before the expression level is computed. Sequences shorter than 18nt (`--min_size`) or longer than 30nt (`--max_size`) will be removed. In addition, low-complexity sequences that consist of one or two bases, such as `AGAGAGAGAGAGAGA`, are removed. The user also has the option to filter against t/rRNA and a user-specified genome. Filtering will have an impact on the number of reads used for normalisation.

After building the expression levels for each sequence, miRProf generates two files: a results table in .csv format and a list of sRNAs (in FASTA format) that match known miRNAs. The results table contains a formatted list of reads that match to known miRNAs. It also contains information about redundant (total) and non-redundant (unique) sequence counts in the input set before and after every filtering step. The total abundance of reads after the final filtering step is used for normalisation [8]. Normalised counts are given in "matching reads per 1 million total reads" (RPM) to make them comparable between samples. The rest of the table lists miRNA matches and associated sequence counts. Small RNAs with matches to multiple miRNAs or miRNA hairpins receive a weighted match count that is obtained by dividing the raw count by the number of matches.

The FASTA file contains the actual sequences from your file. The ID lines contain the following information:

```
>mirnaIDs_n_c
miRNA_sequence
```

where `mirnaIDs` is the identifier obtained by concatenating the IDs of matching miRNAs, n is the consecutive number for each match and c is the raw count for the matching sequence.

Expression profiles of reads can be produced by running miRProf separately on mutliple samples and merging the results tables:

sRNA sequence,	Sample1,	Sample 2,	Sample3,	Sample4
miRNA_1,	10,	1,	100,	25
miRNA_2,	100,	10,	20,	55

Parameters:

- Required
  - mirbase_db The location of the miRBase database to use. miRBase databases can be found in `$INSTALL_PATH/data` directory. The following files should be available:

File Name	Description
mature_all.fa*	all mature miRNA sequences
mature_animal.fa*	mature miRNA sequences in Metazoa
mature_plant.fa*	mature miRNA sequences in Plants

    * for each file an `_plusX` variant is created that contains the mature sequences surrounded by XX at either end. This allows the user to match with overhangs.

    Only same-strand matches of sRNAs to the miRBase databases will be reported.
    miRBase databases can be downloaded and configured using the following command:

    ```
    srna-tools.pl --update_mirbase
    ```

  - out The path to the output directory.
  - srna_file The location of the sRNA file in FASTA format.
- Optional
  - collapse_match_groups Combines sRNAs and their counts based on their match signature. The match signature of a sRNA is formed by combining all matching miRNA IDs, i.e. a sRNA matching both miR156 and miR157 would have a match signature "miR156; miR157". Each sRNA can be unambiguously assigned to one match signature.
  - genome If a genome FASTA file is provided, sRNAs that do not have a genomic match are removed from the analysis.
  - group_family If defined, the matches to different members of the same family are combined into one.
  - group_mismatches If defined, the matches to the same miRNA are combined into groups regardless of the number of mismatches. e.g. counts for sRNAs matching miR156 exactly and with 1-3 mismatches are combined into one.
  - group_organisms If defined, matches to the same miRNA in different organisms are combined into one.
  - group_variant If defined, matches to different variants of the miRNA are combined into one, such as:

* different mature sequences that can arise from the same precursor, annotated in miRBASE as -3p, -5p, -s or -as in the ID of the miRNA and applies to mature sequences only.
* different precursors that produce the same mature sequence, annotated as -1, -2 etc. in miRBASE

See miRBASE help for more details.

- keep_best If defined, only the best matches are kept for each sRNA sequence. For example, if there are miRNAs with a perfect match for a sRNA, no miRNAs from the same organism with any mismatches would be accepted for the same sRNA. This is not applied to miRNA matches from different organisms. Often, sRNAs will match multiple members of the same miRNA family. This option helps to reduce the complexity of the output for those cases.
- maxsize The maximum sRNA length.
  ($18 \leq maxsize \leq 35$, default $maxsize = 25$).
- minsize The minimum sRNA length.
  ($18 \leq minsize \leq 35$, default $minsize = 18$).
- mismatches The maximum allowed number of mismatches.
  ($0 \leq mismatches \leq 3$, default $mismatches = 0$).
- overhangs If defined, mirProf will accept overhanging (5' or 3') bases as mismatches, providing the mirbase database has been specially prepped to do so (the `_plusX` variant of the database). If not defined, sRNAs with overhanging bases are always rejected. For example, this would be counted as 2 mismatches:

```
sRNA :  TTAAACCTAGGCAAATAACGATG
        ||||||||||||||||||x|x
miRNA:  TTAAACCTAGGCAAATAACGGT
```

- trrna If defined, sRNAs will be removed from the analysis if they match sequences in the FASTA t/r RNA file `data/t_and_r_RNAs.fa`.

There are known miRNAs in some species that have a perfect match to the other genomes but are not yet annotated as miRNAs on the newer genomes. To view matches to known miRNAs from a specific organism only, you should not use the –group_organisms option.

Example:

```
srna-tools.pl --tool mirprof --mirbase_db plant_mature
--out output/mp --srna_file files/GSM118373_Rajagopalan_leaf.fa
--collapse_match_groups --genome data/arabidopsis.fa --group_family
```

```
--group_mismatches --group_organisms --group_variant
--keep_best --maxsize 26 --minsize 16 --mismatches 2
--overhangs --trrna
```

3.5 RNA Hairpin Folding and Annotation Tool

This tool produces the secondary structure of a long (up to 1kb) RNA sequence and annotates it by highlighting up to 20 short sequences on the resulting structure.
The tool produces three files:

- PDF file showing the position of miRNA candidate sequences on a precursor hairpin (see figure 3.4).
- JPEG file showing the position of miRNA candidate sequences on a precursor hairpin.
- A text file containing the legend.

Parameters:

- Required
  - longSeq The long (hairpin) sequence in FASTA format, use quotes to give a parameter on more than one line.
  - shortSeqs The short sequence(s) in FASTA format that will be highlighted on the hairpin. These should be subsequences of the longSeq.
  - out The path of the output directory.

Example:

```
srna-tools.pl --tool hp_tool --longSeq files/hairpin.fa --shortSeqs
files/mirna.fa --out output/h
```

Figure 3.4: RNA fold output showing miR164 precursor.

3.6 FiRePat Tool

This tool identifies sRNAs (or sRNA loci) that may influence gene expression. To do this FiRePat (*Fi*nd *Re*gulatory *Pat*terns) computes the profile similarity between series of sRNA and gene expression data. Pairs of entities (sRNAs, genes) that are highly co- or anti-regulated are identified and optimally clustered. In both cases the Pearson Correlation Coefficient is used as a similarity measure. Gene/sRNA profiles with a high degree of co- or anti-regulation might indicate a functional interaction and are therefore interesting subjects for further studies.

3.6.1 Input files

In order to use FiRePat you need expression profiles of sRNAs and genes in at least two samples, e.g. a time series, different treatments or mutants.

The input of this tool consists of two CSV files containing the series. Each row in the input file should contain the same number of values, which should be the 'expression levels' of sRNAs/genes in the series. Each column should contain data coming from the same sample (e.g. a time point or treatment). The order of sRNAs/genes should be identical for all columns forming the table. The header row should indicate the name of each (time) point that will be analyzed. The default format assumes the existence of a header row, the first row will always be considered to be the header of the table.

Gene	WildType	Mutant1	Mutant2	Mutan3	Mutant4	Mutant5
Gene1	7.34	12.57	10.14	7.29	7.33	5.44
Gene2	5.32	5.53	5.12	10.45	10.39	10.47
Gene3	5.12	5.07	10.78	7.12	7.09	3.23
Gene4	6.54	14.58	9.19	6.89	7.03	5.12
Gene5	7.94	12.59	12.17	13.03	12.97	12.76
sRNA	WildType	Mutant1	Mutant2	Mutan3	Mutant4	Mutant5
sRNA1	0.12	6.23	5.59	6.12	5.78	5.46
sRNA2	3.23	3.57	3.14	3.29	3.33	12.75
sRNA3	7.64	15.12	10.76	7.29	7.53	7.44
sRNA4	12.33	12.43	12.44	12.29	12.33	12.45
sRNA5	16.43	0.21	8.44	7.99	8.58	7.78

The input files can be generated by combining expression levels of the different 'products' (sRNA loci, genes, etc) for different conditions (points). For example, the weighted and normalised expression level of sRNA loci can be obtained using the SiLoCo tool (in single sample mode). Expression profiles for microarray datasets can be obtained using standard functions in packages such as affy [11] in the Bioconductor suite for R.

Both files should contain the same time points, treatments etc. for the sRNA and gene data to be comparable. The use of different points does not raise an error, but the number of points must be identical. Also, the order of the constituent points is important because the next steps, such as differential expression analysis and the correlation analysis, are based on differences between consecutive points.

Please see our example files of 150 sRNA loci and genes in an experiment with 10 time points as a template for your own input files. They can be found in `$INSTALL_PATH/data` and are called `firepat_test150_genes.csv` and `firepat_test150_srna_loci.csv`.

3.6.2 Differential Expression Analysis

FiRePat calculates correlation only for series that exhibit differential expression. The top x% of differentially expressed products are picked for further analysis, where x is an input parameter (`--de_threshold`). Set this parameter higher to include more profiles at the cost of a reduction in clustering accuracy.

Correlation and Clustering

Pairs are created from highly co- or anti-regulated series from the two datasets (sRNAs/sRNA loci and genes). For each pair, the Pearson Correlation Coefficient is computed. If the degree of absolute similarity (absolute value of the positive or negative correlation) is above a given threshold (`--sim_threshold`) the pair is selected for further analysis. In order to filter and keep for further analysis the most significant pairs, a high similarity threshold should be given as input parameter.

The resulting pairs are sequentially clustered using two methods: first by *hierarchical* clustering and then by *k-means*. The first method suggests a putative number of clusters and then an automated procedure selects the optimal number of clusters, which is used in the k-means clustering.

Parameters:

- Required
  - gene_file The location of a gene expression file in .csv format e.g. see example in files directory.
  - srna_file The location of a sRNA expression file in .csv format e.g. see example in files directory.
  - out The path of the output directory.
- Optional

– color_int Number of color intervals for html output.
($1 \leq color\_int$, default $color\_int = 10$).

– de_threshold Differential expression threshold.
($1 \leq de\_threshold \leq 100$, default $de\_threshold = 5$).
Note that increasing this parameter will increase the number of selected series and thus the number of possible pairs, slowing down the analysis.

– sim_threshold Similarity threshold.
($85 \leq sim\_threshold \leq 100$, default $sim\_threshold = 95$).

Example:

```
srna-tools.pl --tool firepat --out output/fp
--gene_file files/firepat_test150_genes.csv
--srna_file files/firepat_test150_srna_loci.csv
--color_int 3 --de_threshold 30 --sim_threshold 95
```

In order to emphasize the changes in expression the original data is transformed to log_2 ratios relative to the first point. A positive value across the newly created series suggests an increase in expression level relative to the first point and a negative value suggests a decrease in expression level.
The output consists of two csv files, containing the positively and negatively correlated pairs, respectively, and two html files with a colored version of the tables (see figure 3.5). The last two columns in the output files represent the correlation coefficient between the series that form the pair and the identification number of the cluster to which the pair belongs to.

Figure 3.5: Firepat output on series containing 10 points.

3.7 SiLoCo Tool

This tool predicts sRNA loci using the method described in [18] and [20]. It also enables the user to compare the expression profile of sRNA loci between different samples.

In order to determine the relative position of sRNAs, the reads are mapped to the reference genome using PatMaN [22]. Only full-length, perfect matches are accepted as hits. The genome-matching reads are normalised [19] and weighted by repetitiveness. The normalisation method divides hit counts by the number of redundant reads that match the genome. The normalised count, for each distinct read, is given in "hits per 1 million matching reads". Because it is impossible to decide where a sRNA with multiple matches to the genome originated, we correct the normalised read-abundance for repetitiveness by dividing it by the number of matches to the genome. The result is a weighted hit count.

The method uses the normalised and weighted read-abundance and relative position of sRNAs on the reference genome to predict the sRNA loci. A locus must have a minimum of 3 weighted sRNA hits (this threshold can be adjusted using the `--min_hits` parameter) and no gap (absence of sRNA hits) longer than 300nt (this threshold can be adjusted using the `--max_gap` parameter).

By default SiLoCo compares two sRNA samples by computing the log_2 sRNA expression ratio and the expression average. These measures are used for ranking to help find differentially expressed loci.

Although, SiLoCo compares two samples by default, single-sample mode can also be selected. The datasets must contain sRNA sequence reads in FASTA format, in **redundant form**, i.e. with one entry for each read. Sequences shorter than 18nt (`--minsize` parameter) or longer than 30nt (`--maxsize` parameter) will be removed. Before finding sRNA loci, we remove low-complexity sequences and matches to known t/rRNAs.

Parameters:

- Required
  - genome The location of the genome file in FASTA format.
  - sample_name1 The name of the first sRNA sample e.g. `S1`.
  - srna_file1 The location of the first sRNA FASTA file.
  - out The path of the output directory.
- Optional

- sample_name2 The name of the second sRNA sample e.g. `S2` [required if `--num_samples 2`].
- srna_file2 The location of the second sRNA FASTA file [required if `--num_samples 2`].
- asrp_links If defined ASRP links to Arabidopsis small RNA database (ASRP) [2] will be added to the results file.
- max_gap The maximum gap length in a locus.
  ($1 \leq max\_gap$, default $max\_gap = 300$).
- maxsize The maximum length of a sRNA.
  ($18 \leq maxsize \leq 35$, default $maxsize = 25$).
- min_hits The minimum number of sRNAs in a locus.
  ($1 \leq min\_hits$, default $min\_hits = 3$).
- minsize The minimum length of a sRNA.
  ($18 \leq minsize \leq 35$, default $minsize = 18$).
- num_samples The number of samples.
  ($num\_samples = 1$ or $num\_samples = 2$,
  default $num\_samples = 2$).
- pseudocount The pseudocount that is added to locus expression level (to avoid division by zero errors).
  ($0 < pseudocount$, default $pseudocount = 0.1$).
- tair_links If defined links to TAIR [27] will be added to the results file.
  Links work in MS Excel and OpenOffice calc but there may be versions of these programs with which they do not work. Please note that the hyperlinks will increase the size of your result files significantly.
- trrna If defined, sRNAs matching sequences in the FASTA t/r RNA file `data/t_and_r_RNAs.fa` will be removed.
- uniq If defined, adds columns to the output table for the number of reads in each locus and from each sample, that had only a single hit to the reference genome. This count can be used to filter loci and keep those with only unique matching sRNAs.

Example:

```
srna-tools.pl --tool siloco --genome data/arabidopsis.fa
--out output/si --sample_name1 S1 --sample_name2 S2
--srna_file1 files/GSM118373_Rajagopalan_leaf.fa
--srna_file2 files/GSM154370_Carrington_col0_leaf.fa
--asrp_links --max_gap 100 --maxsize 26 --min_hits 5
--minsize 20 --num_samples 2 --pseudocount 0.2 --tair_links
--trrna --uniq
```

The results are presented in a single csv file. The header of the document contains the description of the data and read counts for sample1 (S1) and sample2 (S2). The number of non-redundant and redundant reads are listed for the input dataset and after each filtering step (if any). Valid sequences are those that passed the filter for size range, low-complexity, t/rRNA and genome matching. The number of total valid reads is used for normalisation.

Locus-data is shown in a table with the following columns:

- Chromosome, start/end position and length
  Genomic location and length of locus in nucleotides. Some incomplete genomes may not yet be assembled into chromosomes and the accessions listed here may be scaffolds or bacs instead. The list is initially sorted by chromosome and position.

- Raw count S1/S2
  Sum of read abundances in samples 1 and 2 that from the locus (not corrected for repetitiveness).

- Weighted count S1/S2
  Sum of raw read abundances divided by number of matches of each sequence to the genome.

- Normalised count S1/S2
  Sum of weighted counts divided by the total number of genome-matching reads in each sample, given in "hits per 1 million genome-matching reads". Normalised counts (abundances) are comparable between samples.

- Uniquely matching reads (optional)
  Number of sequence reads in the locus that only have a single match to the genome.

- log_2 ratio
  A measure for the difference in sRNA abundance for a given locus between the two samples, expressed as $\log_2 \frac{S_1}{S_2}$.
  When a locus is absent in one of the samples i.e. the expression level in one of the samples is 0, the ratio (S1/S2) will be either 0 or inf. It is not possible to calculate $log_2(0)$ or $log_2(\text{inf})$. To avoid this problem, a small pseudocount, with default value of 0.1, is added to all normalised and weighted hit counts. The bias introduced by the arbitrary pseudocount becomes negligible in loci with high expression levels.
  A log_2 ratio of 1 means a two-fold change in sRNA abundance. A locus with a positive log_2 ratio shows an enrichment of sRNAs in sample1, a locus with a negative ratio shows an enrichment in sample2. Unlike the linear ratio (S1/S2), log-ratios are symmetrical around zero.

Note: an increased sRNA abundance in one sample does not necessarily mean that sRNA "expression" from that locus is upregulated. Consider the case of a mutant that looses "expression" of sRNAs from all but a few loci. These loci will show an increased sRNA abundance compared to the wild type because other sRNAs are missing but sRNAs could still be produced at the same rate from these loci in-vivo. In order to rank the loci we use the following measures:

- average normalised count
  The log_2 ratio alone is not sufficient for finding differentially expressed loci, because this measure is unreliable when the sRNA abundance is low in both samples. Good candidate loci should have a high ratio of sRNA abundance and a high average count.

- log_2-ratio rank
  Each locus is given a rank according to its (absolute) log_2 ratio. Low rank numbers indicate a high degree of enrichment/depletion. Equal log_2 ratios share a rank.

- average-based rank
  Is similar to the log_2-ratio rank but based on the average normalised counts.

- weighted rank sum
  This measure can be used to identify candidate loci that show a high degree of enrichment/depletion in one of the samples at a high overall expression level. The rank sum is calculated as follows:

  $$RS = 0.5 \times RR + 0.5 \times AR,$$

  where RS is the rank sum, RR is log_2 ratio rank and AR is the average-based rank.

Example output

SiLoCo can be used to compare two Gene Expression Omnibus (GEO) [3] datasets such as the *Arabidopsis* flower and leaf sets from the Bartel lab [23]:

- Flower sample: `$INSTALL_PATH/files/GSM118372_Rajagopalan_col0_flower.fa`
- Leaf sample: `$INSTALL_PATH/files/GSM118373_Rajagopalan_col0_leaf.fa`

The CSV-formatted output file for this analysis can be found in `$INSTALL_PATH/files/???????`. This file has already been sorted by rank sum and links to the ASRPdb genome-browser are included. The top-ranking loci in this analysis show examples of loci that are highly differentially expressed in leaf and flower tissues.

3.8 SiLoMa Tool

This tool produces a map of sRNAs that match to a reference transcriptome using GMOD (Generic Model Organism Database project, `http://gmod.org/wiki/GMOD`) genome browser. Each sRNA-like read is shown using a colored arrow to indicate the precise location, orientation and abundance with respect to the transcriptome (see figure 3.6).

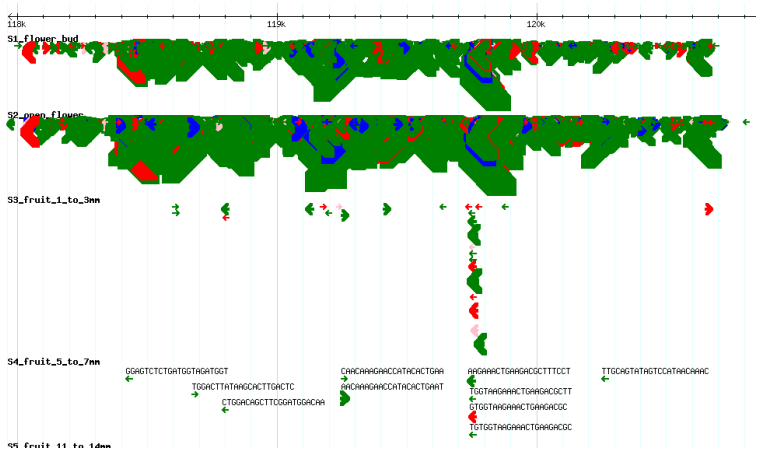

Figure 3.6: SiLoMa output showing a compact sRNA locus.

The reference sequences forming the transcriptome are comprised of either a region in a user-supplied genome or a user-supplied sequence. To match against a region within a genome, enter a chromosome (BAC/scaffold) ID (e.g. one of 1,2,3,4 or 5 for *Arabidopsis*)and a start and an end position. The maximum length of the selected region is 50kbp.

Parameters:

- Required
  - Either pasted_seq, the reference transcript, or genome, the location of the genome file, in FASTA format.
  - srna_file The location of the sRNA file in FASTA format.
  - out The path of the output directory.
- Optional
  - maxsize The maximum length of a sRNA.
    ($16 \leq maxsize \leq 35$, default $maxsize = 30$).
  - minsize The minimum length of a sRNA.
    ($16 \leq minsize \leq 35$, default $minsize = 18$).

- plot_labels Plot labels (sRNA sequences and counts).
- plot_nr Plot sRNA hits in non-redundant form.
- region_chrom The chromosome of the reference transcript in the genome.
- region_start The start position of the reference transcript in the genome.
- region_end The end position of the reference transcript in the genome.

Note: if a genome is supplied, only one transcript is created using the optional parameters `--region_chrom`, `--region_start` and `--region_end`.

Example

```
srna-tools.pl --tool siloma --genome data/arabidopsis.fa
--out output/sm --srna_file files/GSM118373_Rajagopalan_leaf.fa
--pasted_seq TAAGCTATATAGGGGGGT --region_chrom 2 --region_start 39148
--region_end 39445 --maxsize 26 --minsize 20 --plot_labels --plot_nr
```

Some parameters like `--plot_nr` control the graphical output of the tool. If sequences are plotted in non-redundant form, only one arrow is drawn for each unique sRNA sequence and the thickness of the arrow is proportional to the log_{10} of the sequence abundance. If `--plot_nr` is not present (redundant output is requested), multiple arrows are drawn to represent the abundance of each sequence in the sRNA file. Figure 3.7 shows the differences created by the `--plot_nr` parameter. Labels (`--plot_labels`) can also be included in the output, which contain the sRNA sequence and its abundance.

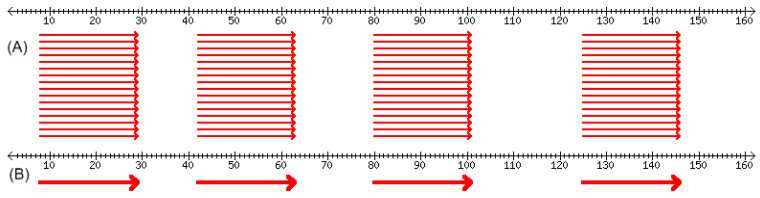

Figure 3.7: SiLoMa output in redundant (A) and non-redundant (B) format.

After processing, SiLoMa produces an archive that contains the following files:

- `JOBNAME_stats.txt`, a table in CSV format, containing an overview of the number of matches (total and unique sequences) to the reference transcript.
  For example, for a sRNA with abundance 2 which matches 4 times to

the reference transcript, the number of sequences displayed in redundant mode will be $1 \times 2 \times 4 = 8$. The number of sequences displayed in non-redundant mode will be $1 \times 4 = 4$ (see example presented in figure 3.7).
In addition, there is also a break-down by strand and sRNA size class (see figure 3.8).

- `JOBNAME_matches.fasta` is a FASTA formatted list of the matching sequences (in alphabetical order). The IDs are in the following format:

  `CONSECUTIVE-NUMBER_COUNTx_pos:MATCH-POSITIONS,`

  where `MATCH-POSITIONS` are in the format `START..END[+/-]` and `[+/-]` represents the strand.

- `JOBNAME_reference-sequence.fasta` is the reference transcript to which sRNAs were aligned.

- `JOBNAME_image.png` is the 'genome browser' figure of the reference transcript and aligned sRNAs. The ruler shows the distance along the reference transcript relative to its start. The sRNAs are shown as arrows. The direction of the arrows indicates the match to positive (pointing right) or negative (pointing left) strand and the colour indicates the sRNA size class (see figure 3.8):

  - pink : 15-19nt
  - red : 20-21nt
  - green : 22-23nt
  - blue : 24-25nt

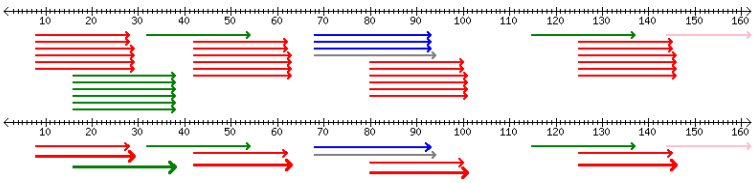

Figure 3.8: SiLoMa output showing sRNAs with different lengths.

Note: only sRNAs that have a full-length perfect match to the reference transcript are displayed. If labels are included, some labels near the edges may be cut off. To avoid this, simply expand the region of interest on the reference transcript to allow the labels to be printed in full. Also, the arrows displaying the sRNAs may be larger than the region the sRNA maps to when

using the non-redundant format with highly abundant sequences and a large region.
Examples of sRNA loci are presented in 3.9, 3.6 and 3.10.

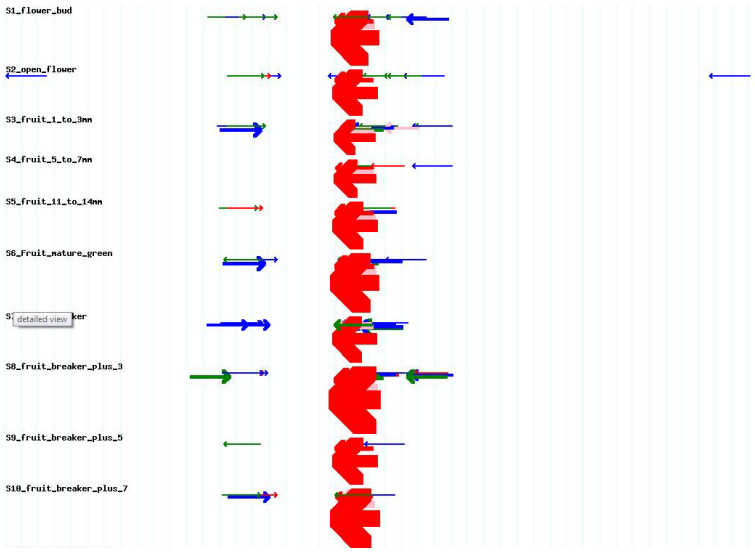

Figure 3.9: miRNA locus.

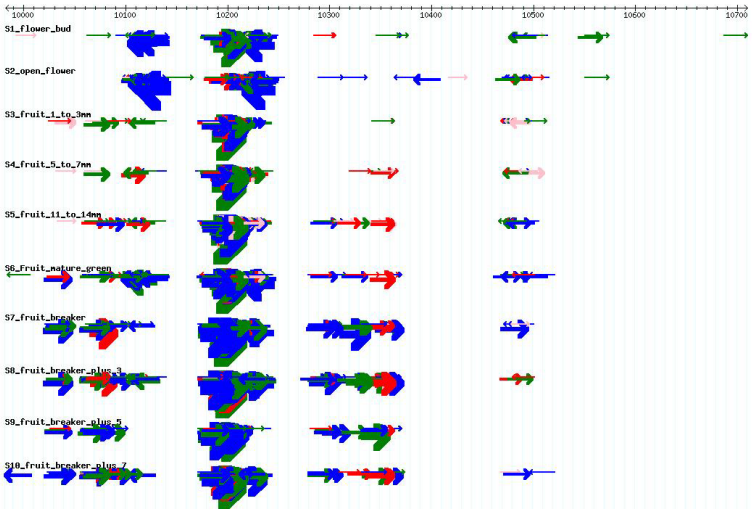

Figure 3.10: hcRNA locus.

3.9 ta-siRNA Prediction Tool

This tool identifies phased 21nt sRNAs characteristic of ta-siRNA loci. It implements the algorithm described in [7] to calculate the probability of the phasing being significant based on the hypergeometric distribution (see figure 3.11). Our implementation differs slightly as we take into account the length of the input sRNA sequences, only using 21nt sRNAs in the phasing analysis. We also require that sRNAs have a raw abundance of at least 2 in order to be included in the analysis.

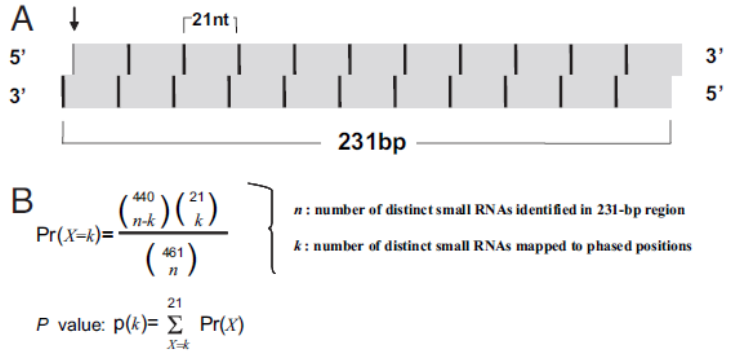

Figure 3.11: prediction of ta-siRNA loci.

Parameters:

- Required
  - genome The location of the genome file in FASTA format.
  - srna_file The location of the sRNA file in FASTA format
  - out The path of the output directory.
- Optional
  - abundance The minimum sRNA abundance.
    ($1 \leq abundance$, default $abundance = 2$).
  - pval The p-value cutoff can be adjusted to increase/decrease the number of loci returned.
    Must be either $0.001, 0.0001, 0.00001, 0.000001$ or 0.0000001, default is 0.0001.
  - minsize The minimum length of a sRNA.
    ($18 \leq minsize \leq 35$, default $minsize = 18$).
  - maxsize The maximum length of a sRNA.
    ($18 \leq maxsize \leq 35$, default $maxsize = 25$).

– trrna If defined, sRNAs matching sequences in the FASTA t/r RNA file `data/t_and_r_RNAs.fa` will be removed.

Example:

```
srna-tools.pl --tool phasing --genome data/arabidopsis.fa
--out output/p --srna_file files/GSM118373_Rajagopalan_leaf.fa
--abundance 3 --pval 0.001 --minsize 20 --maxsize 26 --trrna
```

The results consist of two files. The `locuslist.csv` file contains a list of predicted TAS loci in .csv format which contains the following information:

Chr	Start postition	End position	# sequences	# phased sequences	p-val
1	18553086	18553337	16	9	1.18e-09

- Chr: Chromosome.
- Start position: Start position of the ta-siRNA locus.
- End position: End position of the ta-siRNA locus.
- #sequences: Number of unique sRNAs mapping to this locus.
- #phased sequences: Number of unique sRNAs "in phase".
- p-val: *p*-Value showing the probability of the phasing occurring by chance.

The `srnas.txt` file contains a list of phased sRNAs from each of the predicted TAS loci along with their abundances and genomic coordinates.
For the example shown below:

Chr	Start postition	End position	# sequences	# phased sequences	p-val
2	16544875	16545126	13	8	4.833452e-09

the ta-siRNAs are:

Read	Chromosome	Start position	Strand
CCAATGTCTTTTCTAGTTCGT(19)	2	16544875	1
CGCTATGTTGGACTTAGAATA(6)	2	16544917	1
ATTTTCTAAGATCCACCGATA(12)	2	16544938	1
GAACTAGAAAAGACATTGGAC(4)	2	16544893	-1
TTCTAAGTTCAACATATCGAC(12)	2	16544914	-1
TTCTAAGTCCAACATAGCGTA(301)	2	16544935	-1
TCGGTGGATCTTAGAAAATTA(161)	2	16544956	-1
TACAAGCGAATAGACCATTTA(12)	2	16544977	-1

The first line diplays the locus coordinates as shown in the previous file. Subsequent lines show the ta-siRNA sequences with the abundance in brackets (e.g. `TTCTAAGTCCAACATAGCGTA(301)`). The sequence coordinates (chromosome, start position, orientation) are also shown for each of the predicted ta-siRNAs.

The tool has been tested using the sRNA set in `files/GSM118373_Rajagopalan_leaf.fa` described in [23]. The results obtained using default parameters are shown below:

Chr	Start	End	# seqs	# phased seqs	p-val	LocusInfo
1	18553086	18553337	16	9	1.183951e-09	TAS1b
1	23305788	23306039	6	4	4.549688e-05	PPR repeat gene
2	11729024	11729275	27	10	4.833452e-09	TAS1a
2	16544875	16545126	13	8	4.833452e-09	TAS1c
2	16546892	16547143	29	11	1.886064e-09	TAS2
3	1970346	1970597	5	4	1.563104e-05	AT3G06435.1

3.10 Plant Target Prediction Tool

This tool identifies sRNA targeted transcripts. The rules used for target prediction are based on those suggested in [1] and [25]. Specifically, miRNA/target duplexes must obey the following rules:

- No more than four mismatches between sRNA and target (G-U bases count as 0.5 mismatches).
- No more than two adjacent mismatches in the miRNA/target duplex.
- No adjacent mismatches in positions 2-12 of the miRNA/target duplex (the positions are indexed starting with the 5' end of the miRNA).
- No mismatches in positions 10-11 of miRNA/target duplex.
- No more than 2.5 mismatches in positions 1-12 of the of the miRNA/target duplex.
- $MFE_{miRNA/target} \geq 0.74 \cdot MFE_{miRNA/miRNA*}$

Parameters:

- Required
  - transcriptome The location of the transcriptome file in FASTA format.
  - out The path of the output directory.
  - Either pasted_srnas or srna_file containing sequences in FASTA format. `--pasted_srnas` should contain no more than 50 sequences. If `--srna_file` is provided, it should specify the location of the sRNA file. Allowed nucleotide symbols: `A,G,C,T,U,N`.

Example:

```
srna-tools.pl --tool target --out output/t --pasted_srnas '>a
GCTTCTATCTTTTTCTTTCGTGCT' --transcriptome arabidopsis.fa
```

A target prediction results file looks as shown below:

```
>AT4G33780.1/287-309     | Symbols:  | FUNCTIONS IN:
molecular_function unknown; INVOLVED IN: biological_process unknown;
LOCATED IN: chloroplast; EXPRESSED IN: 24 plant structures; EXPRESSED
DURING: 15 growth stages; BEST Arabidopsis thaliana protein match is:
SHW1 (SHORT HYPOCOTYL IN WHITE LIGHT1) (TAIR:AT1G69935.1); Has 20 Blast
```

```
hits to 20 proteins in 5 species: Archae - 0; Bacteria - 0; Metazoa - 0;
Fungi - 0; Plants - 20; Viruses - 0; Other Eukaryotes - 0 (source: NCBI
BLink). | chr4:16201831-16203641 REVERSE

5' AGAAGAUGAUGAUGAUCACG-AGGAAGAAGAUAGAAGCUUG 3'
                  | |||| |o|||o||||||||||||
3'                UCGUGCUUUCUUUUUCUAUCUUCG    5'

>AT4G33780.1
GAGCGTGTTGATGCATAACGAACGATGCCATTTTCCGCATCAATCTCATCGCCTTCTTCTTCTGTCGCG
CTTCTTCGATCGCCTCTCTCTTTCTTCATCTTCACTCCCAAAACCCTAATCTTCACCAGAACCAGGATC
TCTGGTTTCCCTTATCTTGCTTCCCGGCGATCCCGCGATTTCATCAACGGGAGGGATGATTTCGCTGAC
GATACGAGGAGCTGGAACCGGAAGATCAAACCGGAGTATGGGTTCGATGAGGATTACGATGGAGAAGAA
GATGATGATGATCACGAGGAAGAAGATAGAAGCTTGGATCTGTTACTTAGATTTGTAGAAAATGTTTTC
AGAAAGATTTCTAAGAGAGCAAGGAAAGCTGTCCGATCAATTTTGCCTGTTTCGATCTCTACGAAGCTC
GTGGGGTTTTCAGTGAATGGAGTACTTATTCTTGCTTTTTTGTGGATTTTGAAGGCTTTCCTCGAGGTA
GCTTGCACACTTGGAACTATTGTATTTACGAGCATTCTACTTATACGTGGACTTTGGGCCGGAGTAGCA
TACATGCAAGAGAGCCGCAACAATAGGATCAATGAACTCGCTGATGATCCTCGTGCATGGAACGGGATG
CAACCAGTTTCCTGATGAATTCGCTTTACACTTGTAGAAATCAGAATTCTGACTTTTGGGAGAGCCATA
ATTGTTTAGGTTCTTCCAAGGCAATAAAACCACAGCTGAGTTCAGAATCAGAAAGCAGTTACAGTGGAT
GTTCATTGGCAATGTCTGATGATTTAGTAAGTAAAAAAAAGTGTAATATTGTAGCATTCACCAAGTCAGC
TATGCTGGTGTGTAGCTCAACTGGGAACTAAGTCGTCGCCAATGGTGACCATGTTTTCTTAGTTTCTAA
ATAAATAAACCAAACATATAGAACATACCGTTTTCCTTCTAGTTTTGTATATATAACCAAAATTAGTAG
ACTTCAATTTTTC
```

The following information is shown:

1. sRNA ID/accession
2. Target transcript ID/accession and start-end position of the target site
3. Any information/annotation this sequence may have
4. Alignment of the miRNA (bottom sequence) to the target site (top sequence):
   - "—" represents a base pair
   - " " represents a mismatch
   - "o" represents a G-U basepair
5. Full sequence of the predicted target

In addition, a .csv file containing a summary of all potential targets is produced.

Chapter 4

Troubleshooting and FAQ

4.1 General

What is a FASTQ file?

FASTQ format is a text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores. Both the sequence letter and quality score are encoded with a single ASCII character for brevity. Our tools do not use the quality scores in the FASTQ files; only the sequence is used for downstream processing.

A FASTQ file normally uses four lines per sequence. Line 1 begins with a '@' character and is followed by a sequence identifier and an optional description (like a FASTA title line). Line 2 is the raw sequence letters. Line 3 begins with a '+' character and is optionally followed by the same sequence identifier (and any description). Line 4 encodes the quality values for the sequence in Line 2, and must contain the same number of symbols as letters in the sequence.

A minimal FASTQ file might look like this:

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

What is a FASTA file?

In these tools, we use the FASTA format for sRNA input files, genomes and transcriptomes. For short reads, a FASTA file contains two lines for each read. The first line shows the header of the read and the second, the nucleotide sequence of the read. An example file is shown below:

```
>ILLUMINA_READ_1
GGCCATCGAATATTA
>ILLUMINA_READ_2
GGTTTATGACACCTA
```

The genomes and transcriptomes may use multiple lines for the nucleotide sequence.

A FASTA file in redundant format is as follows:

```
> ILLUMINA_READ_1
GGCCATCGAATATTA
> ILLUMINA_READ_1
GGCCATCGAATATTA
> ILLUMINA_READ_1
GGCCATCGAATATTA
> ILLUMINA_READ_1
GGCCATCGAATATTA
> ILLUMINA_READ_1
GGCCATCGAATATTA
```

In a redundant file the abundance is not specified, i.e. if a sequence is present than it was sequenced during the experiment.

A FASTA file in non-redundant format represents the sequence abundance at the end of the sequence header.

```
>identifier(abundance)
sequence
```

The example above, in non-redundant form, is shown below:

```
> ILLUMINA_READ_1(5)
GGCCATCGAATATTA
```

What is a CSV file?

The comma-separated values (CSV) file format is a set of file formats used to store tabular data in which numbers and text are stored in plain textual form that can be read in a text editor. Lines in the text file represent rows of a table, and commas in a line separate what are fields in the tables row. CSV files can be opened in MS Excel or other spreadsheet programs.

sRNA sequence,	Sample1,	Sample 2,	Sample3,	Sample4
AAAGTCGTA,	10,	1,	100,	25
GCTTCGAAA,	100,	10,	20,	55
GTCAGCTCC,	34,	7,	25,	53
CCGTAGCCA,	37,	2,	64,	67
ACGTCAGAG,	27,	5,	1000,	36

I pasted in the example code and it did not work

First check if everything has been installed correctly. In particular make sure the tool and the dependencies are available in the PATH.
Potentially, the problem lies with the new line character. You must remove the new line character before pasting the command e.g. paste into a text editor of your choice first, remove then new line and then paste into the command prompt.

Cannot find error template file

If you receive an error e.g. cannot find error template file, that is because you have not set the correct paths in the `application.conf` file.

patman bin file not accessible or not executable

Make sure the PatMaN executable is in `/usr/local/bin` or, if not, on the path for your local machine. Use `which patman` to see if PatMaN can be found on the path. Use `chmod` to give executable permissions.

Can not find `organisms.txt` when updating miRBase

This is normally caused by a problem writing to the `/data` directory. Ensure you have write access.

This can be also due to the file missing in a new version of miRBase, in which case users should contact miRBase for help.

Can not move results to output directory

Ensure that the output directory exists prior to running the toolkit.

Error reading input Fasta file

Some tools (e.g. miRCat) require the sRNA sequences in redundant format. Please check the specifications of each tool prior to running it.

Error with Parameters

All parameter names must be precedeed with a '–'. Using a single dash will not work as it is interpreted as a negative value.

4.2 Tool Specific Errors

4.2.1 Sequence File Pre-Processing Tool

How many sequences are retained and how many are removed?

A summary text file is created when the tool is run containing details of how many sequences matched the adaptor and how many were not within the size limits.

If a large number of sequences do no match the adaptor it is advisable to check the input file on the command line using the Unix command `less`. Also, make sure that the adapter sequence is spelt correctly and that only a prefix, for the 3' adapter or a suffix, for the 5' adapter is given.

4.2.2 Filter Tool

Even with no filter options some sequences are removed

Low complexity sequences which, by their nature, are likely to match to the genome many times are removed by default to avoid an unnecessarily large PatMaN file being created.

A sequence of interest was removed by the t/rRNA filter

The t/rRNA file contains sequences from Rfam [12, 10], the Genomic tRNA Database [6] and EMBL [16], release 95 (09-Jun-2008) and might be out of date. This file can be replaced with a more recent file downloaded from `http://www.sanger.ac.uk/resources/databases/rfam.html`. The sequence of interest might have a random match to an annotated t/rRNA in another species. If you are not sure what sequences should be kept for further analysis and which sequences should be removed, we suggest you to leave the t/rRNA filtering for later steps.

When should I use the `--make_nr` option?

This option prepares a FASTA file for e.g. complexity analysis (number of unique sequences to number of redundant sequences).
It also represents a compact version of the redundant file, making it a suitable solution for data storage.

However, some tools require the input FASTA file in redundant format. Please check the specifications for each tool prior to running it.

4.2.3 miRCat

Does the tool predict all known miRNAs present in the sample?

Possibly not since miRNAs present at low abundance may be filtered out.

Which are the best miRCat candidates?

The best indication of a good miRNA is a high abundance and presence of a miRNA*. The randfold p-Value also provides some indication of the quality of the miRNA.

4.2.4 miRProf

miRBase files not prepared for overhanging matches

miRBase files are currently not prepared for overhanging matches, if you email us we will do our best to prepare such files.

Error reading input FASTA file

The input FASTA file must be in redundant format as miRProf will count the occurrence of each sequence.

4.2.5 RNA Hairpin Folding and Annotation Tool

Pasting the example command gave error "please correct parameter input".

Check line breaks on the command line. A new line should only be used in the nucleotide data to separate the header from the sequence data. The FASTA data must be enclosed in single quotes '...'. The following example shows where the line breaks should be:

```
srna-tools.pl --tool hp_tool --longSeq '>hairpin
GGGAGCGGGGCTTCGATGATCGCTCGGTTTGAACGGATAGAGCGAATTCTGAGTGGTGCTCCC'
shortSeqs '>mirna
GATAGAGCGAATTCTGAGTGGT' --out output/h
```

4.2.6 FiRePat

Short RNA file not recognised

Unlike other tools in the tookit, you must provide a .csv table of expression values.

sRNA sequence,	Sample1,	Sample 2,	Sample3,	Sample4
AAAGTCGTA,	10,	1,	100,	25

Where can I find gene expression data for similar samples?

We have downloaded our example data (both sRNA and gene data) from GEO `http://www.ncbi.nlm.nih.gov/geo/` [3].

Why does it take so long run?

FiRePat creates all possible correlated pairs. If the correlation threshold was low (e.g 90) then the number of pairs is large and the clustering step takes more time to complete.

How do I interpret the results?

The results are clustered on expression levels and the correlation (positive or negative) is shown in the last column of the table. If present, annotations on both genes and sRNAs can facilitate a biological hypothesis.

Is the correlation coefficient reliable?

If the expression values in both series (gene and sRNA series) are comparable (i.e. the expression ranges are comparable) then the Pearson Correlation Coefficient will accurately compute the similarity between series.

4.2.7 SiLoCo

Some input sequences are excluded from analysis

Sequences shorter than 18nt or longer than 30nt will be automatically removed. In addition, we remove low-complexity sequences that consist of one or two bases only, such as `AGAGAGAGAGAGAGA`.

4.2.8 SiLoMa

How do I know the GMOD Genome Browser is working?

You will get an error on the command line if this is the case.

How can I represent the sRNAs in different samples?

Currently the multiple-sample feature is not supported. The input files can be either merged in one file and displayed, or independent figures can be created for each sample.

4.2.9 ta-siRNA Prediction Tool

How does noise influence the accuracy of the results?

Currently we do not apply any cleaning procedure by default. However, the data can be filtered using the Filter tool before using the ta-si Prediction tool.

Some sequences are excluded from analysis

Some input sequences are excluded from analysis:

- Only sequences with a read count of two or more are included.
- Only sRNAs of 21nt are included.
- Only 21nt phase groups are identified.
- Low complexity sequences (those composed of fewer than three different nucleotides) are filtered out to limit the size of genomic match files.

4.2.10 Plant Target Prediction Tool

Can not find transcriptome

Some transcriptomes can be found in `$INSTALL_PATH/data/transcriptomes` and referred to only by the file name on the command line e.g. `'--transcriptome arabidopsis.fa'` will cause the program to look for a transcriptome at `$INSTALL_PATH/data/transcriptomes/arabidopsis.fa`.

Bibliography

[1] Edwards Allen, Zhixin Xie, Adam M Gustafson, and James C Carrington. microrna-directed phasing during trans-acting sirna biogenesis in plants. *Cell*, 121(2):207–221, Apr 2005.

[2] Tyler W H Backman, Christopher M Sullivan, Jason S Cumbie, Zachary A Miller, Elisabeth J Chapman, Noah Fahlgren, Scott A Givan, James C Carrington, and Kristin D Kasschau. Update of asrp: the arabidopsis small rna project database. *Nucleic Acids Res*, 36(Database issue):D982–D985, Jan 2008.

[3] Tanya Barrett, Tugba O Suzek, Dennis B Troup, Stephen E Wilhite, Wing-Chi Ngau, Pierre Ledoux, Dmitry Rudnev, Alex E Lash, Wataru Fujibuchi, and Ron Edgar. Ncbi geo: mining millions of expression profiles–database and tools. *Nucleic Acids Res*, 33(Database issue):D562–D566, Jan 2005.

[4] D. C. Baulcombe. In vitro replication of plant viral rna. *Curr Biol*, 1(1):53–54, Feb 1991.

[5] James C Carrington and Victor Ambros. Role of micrornas in plant and animal development. *Science*, 301(5631):336–338, Jul 2003.

[6] Patricia P Chan and Todd M Lowe. Gtrnadb: a database of transfer rna genes detected in genomic sequence. *Nucleic Acids Res*, 37(Database issue):D93–D97, Jan 2009.

[7] Ho-Ming Chen, Yi-Hang Li, and Shu-Hsing Wu. Bioinformatic prediction and experimental validation of a microrna-directed tandem trans-acting sirna cascade in arabidopsis. *Proc Natl Acad Sci U S A*, 104(9):3318–3323, Feb 2007.

[8] Noah Fahlgren, Christopher M Sullivan, Kristin D Kasschau, Elisabeth J Chapman, Jason S Cumbie, Taiowa A Montgomery, Sunny D Gilbert, Mark Dasenko, Tyler W H Backman, Scott A Givan, and James C Carrington. Computational and analytical framework for small rna profiling by high-throughput sequencing. *RNA*, 15(5):992–1002, May 2009.

[9] Marc R Friedlaender, Wei Chen, Catherine Adamidi, Jonas Maaskola, Ralf Einspanier, Signe Knespel, and Nikolaus Rajewsky. Discovering microrna from deep sequencing data using mirdeep. *Nat Biotechnol*, 26(4):407–415, Apr 2008.

[10] Paul P Gardner, Jennifer Daub, John G Tate, Eric P Nawrocki, Diana L Kolbe, Stinus Lindgreen, Adam C Wilkinson, Robert D Finn, Sam Griffiths-Jones, Sean R Eddy, and Alex Bateman. Rfam: updates to the rna families database. *Nucleic Acids Res*, 37(Database issue):D136–D140, Jan 2009.

[11] Laurent Gautier, Leslie Cope, Benjamin M Bolstad, and Rafael A Irizarry. affy–analysis of affymetrix genechip data at the probe level. *Bioinformatics*, 20(3):307–315, Feb 2004.

[12] Sam Griffiths-Jones, Alex Bateman, Mhairi Marshall, Ajay Khanna, and Sean R Eddy. Rfam: an rna family database. *Nucleic Acids Res*, 31(1):439–441, Jan 2003.

[13] Gregory J Hannon. Rna interference. *Nature*, 418(6894):244–251, Jul 2002.

[14] Ivo L Hofacker. Rna secondary structure analysis using the vienna rna package. *Curr Protoc Bioinformatics*, Chapter 12:Unit12.2, Jun 2009.

[15] Ana Kozomara and Sam Griffiths-Jones. mirbase: integrating microrna annotation and deep-sequencing data. *Nucleic Acids Res*, 39(Database issue):D152–D157, Jan 2011.

[16] Tamara Kulikova, Philippe Aldebert, Nicola Althorpe, Wendy Baker, Kirsty Bates, Paul Browne, Alexandra van den Broek, Guy Cochrane, Karyn Duggan, Ruth Eberhardt, Nadeem Faruque, Maria Garcia-Pastor, Nicola Harte, Carola Kanz, Rasko Leinonen, Quan Lin, Vincent Lombard, Rodrigo Lopez, Renato Mancuso, Michelle McHale, Francesco Nardone, Ville Silventoinen, Peter Stoehr, Guenter Stoesser, Mary Ann Tuli, Katerina Tzouvara, Robert Vaughan, Dan Wu, Weimin Zhu, and Rolf Apweiler. The embl nucleotide sequence database. *Nucleic Acids Res*, 32(Database issue):D27–D30, Jan 2004.

[17] Anthony A Millar and Peter M Waterhouse. Plant and animal micrornas: similarities and differences. *Funct Integr Genomics*, 5(3):129–135, Jul 2005.

[18] Attila Molnar, Frank Schwach, David J Studholme, Eva C Thuenemann, and David C Baulcombe. mirnas control gene expression in the single-cell alga chlamydomonas reinhardtii. *Nature*, 447(7148):1126–1129, Jun 2007.

[19] Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by rna-seq. *Nat Methods*, 5(7):621–628, Jul 2008.

[20] Rebecca A Mosher, Frank Schwach, David Studholme, and David C Baulcombe. Polivb influences rna-directed dna methylation independently of its role in sirna biogenesis. *Proc Natl Acad Sci U S A*, 105(8):3145–3150, Feb 2008.

[21] Simon Moxon, Frank Schwach, Tamas Dalmay, Dan Maclean, David J Studholme, and Vincent Moulton. A toolkit for analysing large-scale plant small rna datasets. *Bioinformatics*, 24(19):2252–2253, Oct 2008.

[22] Kay Prufer, Udo Stenzel, Michael Dannemann, Richard E Green, Michael Lachmann, and Janet Kelso. Patman: rapid alignment of short sequences to large databases. *Bioinformatics*, 24(13):1530–1531, Jul 2008.

[23] Ramya Rajagopalan, Herv Vaucheret, Jerry Trejo, and David P Bartel. A diverse and evolutionarily fluid set of micrornas in arabidopsis thaliana. *Genes Dev*, 20(24):3407–3425, Dec 2006.

[24] Brenda J Reinhart, Earl G Weinstein, Matthew W Rhoades, Bonnie Bartel, and David P Bartel. Micrornas in plants. *Genes Dev*, 16(13):1616–1626, Jul 2002.

[25] Rebecca Schwab, Javier F Palatnik, Markus Riester, Carla Schommer, Markus Schmid, and Detlef Weigel. Specific effects of micrornas on the plant transcriptome. *Dev Cell*, 8(4):517–527, Apr 2005.

[26] Frank Schwach, Simon Moxon, Vincent Moulton, and Tamas Dalmay. Deciphering the diversity of small rnas in plants: the long and short of it. *Brief Funct Genomic Proteomic*, 8(6):472–481, Nov 2009.

[27] David Swarbreck, Christopher Wilks, Philippe Lamesch, Tanya Z Berardini, Margarita Garcia-Hernandez, Hartmut Foerster, Donghui Li, Tom Meyer, Robert Muller, Larry Ploetz, Amie Radenbaugh, Shanker Singh, Vanessa Swing, Christophe Tissier, Peifen Zhang, and Eva Huala. The arabidopsis information resource (tair): gene structure and function annotation. *Nucleic Acids Res*, 36(Database issue):D1009–D1014, Jan 2008.

[28] P. M. Waterhouse, M. B. Wang, and T. Lough. Gene silencing as an adaptive defence against viruses. *Nature*, 411(6839):834–842, Jun 2001.

[29] P. D. Zamore, T. Tuschl, P. A. Sharp, and D. P. Bartel. Rnai: double-stranded rna directs the atp-dependent cleavage of mrna at 21 to 23 nucleotide intervals. *Cell*, 101(1):25–33, Mar 2000.