Methods in Molecular Biology

Antiviral Resistance in Plants

Chapter Title:

Computational workflow for small RNA profiling in virus-infected plants

Livia Donaire¹ and César Llave²

Affiliation:

¹Departamento de Biología del Estrés y Patología Vegetal. Centro de Edafología y Biología Aplicada del Segura (CEBAS), Consejo Superior de Investigaciones Científicas (CSIC), Murcia, Spain

²Departamento de Biología Medioambiental, Centro de Investigaciones Biológicas (CIB), Consejo Superior de Investigaciones Científicas (CSIC), Madrid, Spain.

Corresponding author: César Llave (cesarllave@cib.csic.es)

Running head: Bioinformatic toolkit for small RNA profiling

Abstract

In this chapter we describe a series of computational pipelines for the *in silico* analysis of small RNAs (sRNA) produced in response to viral infections in plants. Our workflow is primarily focused on the analysis of sRNA populations derived from known or previously undescribed viruses infecting host plants. Furthermore, we provide an additional pipeline to examine host-specific endogenous sRNAs activated or specifically expressed during viral infections in plants. We present some key points for a successful and cost-efficient processing of next generation sequencing sRNA libraries, from purification of high quality RNA to guidance for library preparation and sequencing strategies. We report a series of free available tools and programs as well as in-house Perl scripts to perform customized sRNA-seq data mining. Previous bioinformatic background is not required, but experience with basic Unix commands is desirable.

Key Words

Plant viruses, small RNAs, antiviral silencing, sRNA-seq, bioinformatic analysis, next generation sequencing

1. Introduction

RNA silencing is an important antiviral mechanism in plants and other organisms (**1**). The accumulation of virus-derived small RNAs (vsRNAs) in infected tissues and their contribution to antiviral silencing have been proven for many plant viruses (reviewed by (**2**)). Since the advent of modern next generation sequencing (NGS) technics, sequencing the small RNA (sRNA) transcriptome in the plant cell is high throughput and effortless in terms of experimental concerns (**3**). At the time of writing, at least four different massive sequencing platforms are available for sRNA profiling (454 GS FLX, Life Sciences-Roche; Illumina GA/HiSeq/MiSeq/NextSeq, Solexa; Applied Biosystems SOLiD; Ion Torrent personal genome machines/proton). All these platforms have standardized protocols for library preparation, which for sRNA profiling involves isolation of

sRNAs and attachment to specific adapter sequences. Performance comparison between different NGS platforms along with advantages and limitations for each technology has been widely discussed (4). A necessary step, and a major challenge for most biologists, is the transformation of raw data into genome-scale datasets that can be interpreted and applied at the molecular, cellular and genomic levels. The NGS computational workflow requires a number of sophisticated tools for monitoring, viewing, analyzing and storing data that are hardly manageable for scientists in the field of the biological research (5, 6). Here we outline four current analytical pipelines for computational profiling of both vsRNAs and endogenous virus-associated sRNAs in virus-infected plants. We do not aim to provide an exhaustive compilation of dedicated software and bioinformatic tools, or to dissect methods for preparation of sRNA sequencing libraries. Rather, we aim to share an elementary guideline for the analysis of sRNA-seq data using in-house and free available bioinformatic resources.

The proposed workflow consists of four multi-step pipelines. Pipeline 1 encompasses the analysis of raw data, including data quality tests, size filtering and subtraction of contaminant RNA species such as structural RNAs. Pipeline 2 and 3 are devoted to characterize vsRNA populations. When our target viral genome is available, it is possible to track the origin of the vsRNAs by mapping sRNA-seq reads onto the viral genome. Pipeline 2 involves the alignment of filtered short reads against a viral reference genome and it is used to profile sRNA derived from known viruses. As a result, specific signatures such as origin, relative abundance, size distribution, strand orientation, or 5' nucleotide identity are reported for both total and unique vsRNA sequence reads. For viruses without sequenced genomes, identification is achieved by first assembling overlapping sRNA reads into longer contiguous pieces of cDNA. Pipeline 3 is thus used for viral discovery and for resolving the genome organization of novel infecting viruses in host plants. Finally, Pipeline 4 allows the identification and quantification of virus-responsive sRNAs derived from host genomic loci.

2. Materials

2.1. RNA isolation and sRNA library construction

1. A method for high quality plant RNA isolation.
2. Agilent 2100 BioAnalyzer.
3. Qubit 3.0 fluorometer.
4. A method for sRNA library preparation.

2.2. *In silico* bioinformatic analysis

2.2.1. Hardware requirements

1. Operating system: 64-bit Linux (*see Note 1*).
2. Available RAM: at least 4 GB (recommend 16 GB or higher).
3. Free disk space: depends on the number of samples and sequencing depth (20-50 GB per sample).

2.2.2. Software requirements (*see Note 2*).

1. Java SE Runtime Environment.

To run java applications (recommended version 1.6 JREs from Oracle or higher), check the java version installed in your computer by typing in a Terminal:

```
$ java -version
```

If java is not already installed or the available version is older than the recommended one, install Oracle Java 8 by typing in the command line:

```
$ sudo add-apt-repository ppa:webupd8team/java
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install oracle-java8-installer
```

2. Perl and Bioperl.

Perl is already installed in most Linux distributions. Check that the current Perl version in your computer is 5.0 or higher by typing:

```
$ perl -version
```

To install Bioperl, type:

```
$ sudo apt-get install bioperl
```

3. SRAToolkit (7).

To install the SRA toolkit, type:

```
$ sudo apt-get install sra-toolkit
```

4. FastQC.

To download FastQC (8), type:

```
$ wget https://launchpad.net/ubuntu/+archive/primary/+files/fastqc_0.11.5+dfsg-3_all.deb
```

To install it, type:

```
$sudo dpkg -i fastqc_0.11.5+dfsg-3_all.deb
```

```
$sudo apt-get install -f
```

5. Adapter Removal.

Download AdapterRemoval (9) from GitHub by typing:

```
$ wget -O adapterremoval-2.1.7.tar.gz
```

<https://github.com/MikkelSchubert/adapterremoval/archive/v2.1.7.tar.gz>

To unpack it, type:

```
$ tar -xvzf adapterremoval-2.1.7.tar.gz
```

To compile and install it, type:

```
$ cd adapterremoval-2.1.7
```

```
/adapterremoval-2.1.7/$ make
```

```
/adapterremoval-2.1.7/$ sudo make install
```

(see **Note 3**)

6. Perl scripts from Github.

Download the Perl script “trim.fastq.length.pl” from

<https://github.com/MadsAlbertsen/miscperlscripts>.

Likewise, download from <https://github.com/ldonaire/Perl-scripts> the eight following Perl scripts:

“countfq_20to26.pl”, “countfa_20to26.pl”, “fqtofunique.pl”, “misis_table_analysis.pl”,

“nucomp_fa.pl”, “nucomp_fq.pl”, “rna_to_dna.pl” and “srna_count_comparison.pl”.

Save all of them into a new folder called “perl_scripts” in your working directory (see **Note 4**).

7. Python and Python packages.

Python is already installed in most Linux distributions. To check the version installed in your computer type:

```
$ python --version
```

First, install pip package management system:

```
$ sudo apt-get install python-pip
```

Then, install numpy and pyfasta packages as follows:

```
$ sudo python -m pip install numpy
```

```
$ sudo python -m pip install pyfasta
```

8. Bowtie.

Install Bowtie (**10**) by typing:

```
$ sudo apt-get install bowtie
```

9. SAMools.

To install SAMtools (**11**), type:

```
$sudo apt-get install samtools
```

10. MISIS.

Download MISIS (**12**) from:

<https://www.fasteris.com/apps/misis.php>.

Unpack it into your working directory:

```
/working_directory$ tar -xvzf Misis.tar.gz
```

11. Integrated Genomics Viewer (IGV).

Download the binary distribution of IGV (**13**) from:

<http://software.broadinstitute.org/software/igv/download>.

Unpack it into your working directory:

```
/working_directory$ unzip IGV_2.4.3.zip
```

12. eXpress

Download the Linux 64-bit executable application of eXpress (**14**) from:

<https://pachterlab.github.io/eXpress/index.html>.

Unpack it into your working directory: `/working_directory$ tar -xzvf express-1.5.1-linux_x86_64.tgz`

Give executable permissions by typing:

```
/working_directory$ chmod +x express-1.5.1-linux_x86_64/express
```

2.2.3. Local databases

1. Creating a local Rfam index (*see* **Note 5** and **Note 6**):

Download Rfam data in fasta format from version 11.0 (or previous):

```
/working_directory$ wget  
ftp://ftp.ebi.ac.uk/pub/databases/Rfam/11.0/Rfam.fasta.gz
```

Unpack it into a new folder called “Rfam_index” in your working directory:

```
/working_directory/Rfam_index$ gunzip Rfam.fasta.gz
```

Convert RNA sequences into DNA by using the Perl script “rna_to_dna.pl” (Materials, 2.2.2, step 6):

```
/working_directory/Rfam_index$ perl perl_scripts/rna_to_dna.pl  
Rfam_index/Rfam.fasta
```

The output file will be a fasta file called Rfam-DNA.fasta.

If you want to search for miRNAs in your datasets, it is necessary to first remove pre-miRNA and miRNA sequences from the Rfam fasta file. To do so, create a file containing the fasta IDs of all pre-miRNAs and miRNAs included by using the Unix command “grep” and remove the first “>” using the Unix command “sed”:

```
/working_directory/Rfam_index$ grep -E 'miR|MIR|mir' Rfam-DNA.fasta | sed  
's/^> //' > miR_headers.txt
```

The output file “miR_headers.txt” contains 27394 fasta IDs of miRNAs and pre-miRNA sequences from different organisms.

Then, use pyfasta (Materials 2.2.2, step 7) to eliminate those sequences from the fasta file as follows:

```
/working_directory$ pyfasta extract --header --fasta Rfam_index/Rfam-DNA.fasta -  
-exclude --file miR_headers.txt > Rfam-modif.fasta
```

The output file “Rfam-modif.fasta” contains 35,5610 fasta sequences.

To create a Bowtie index, type:


```
/working_directory$ bowtie-build Rfam_index/Rfam-modif.fasta Rfam_index/Rfam
```

The output should be six files with ebwt extension.

2. Creating local plant virus indexes

For known viruses, virus reference genomes can be downloaded in fasta format from the NCBI sequence repository (nucleotide database). As an example, our reference genome consists of the *Tobacco rattle virus* isolate PpK20, a positive, single-stranded RNA virus with a bipartite genome. The sequences of TRV1 and TRV2 with the accession numbers NC_003805 and NC_003811, respectively, are downloaded into our working directory and referred to as TRV1.fasta and TRV2.fasta.

To create a Bowtie index of TRV1, type:

```
/working_directory$ mkdir virus_index  
/working_directory$ mv TRV1.fasta virus_index  
/working_directory$ bowtie-build virus_index/TRV1.fasta virus_index/TRV1
```

The output should be six files with ebwt extension.

3. Creating a local plant host index

Genomic or transcriptomic fasta files of the plant hosts can be downloaded from different plant genomic databases such as Solgenomics, TAIR, PlantGDB, NCBI. For instance, our test experiment is run on the Arabidopsis transcriptome (CDS+UTRs, version 10) available at the TAIR website:

https://www.arabidopsis.org/download_files/Sequences/TAIR10_blastsets/TAIR10_cdna_20101214_updated

Create a new folder called “plant_index” and save the Arabidopsis cDNAs into a file called “TAIR10_cdna_20101214_updated.fasta”.

It is highly recommended to shorten the fasta headers to avoid later execution problems. To remove the description and leave only the gen ID use the Unix “awk” command:

```
/working_directory$ awk '{print /^>/ ? $1 : $0}' plant_index/  
TAIR10_cdna_20101214_updated.fasta > plant_index/TAIR10cdna.fasta
```

To create a Bowtie index, type the following:

```
/working_directory$ bowtie-build plant_index/TAIR10cdna.fasta  
plant_index/TAIR10cdna
```

The output should be six files with ebwt extension.

3. Methods

3.1. Viral-small RNA library construction and sequencing

In the following section we briefly provide some clues and tricks to optimize our sequencing performance. It is not our goal to offer a detailed description of current protocols for library preparation and sequencing, which have been extensively reviewed by others (*15, 16*).

1. Purification of High-Quality RNA

Several methods are suitable for high-quality RNA isolation including guanidine isothiocyanate and acid-phenol methods such as TRIzol Reagent (Thermo Fisher Scientific) or TRI reagent (Sigma-Aldrich/Merck). If using commercial kits, ensure that short RNAs (< 200 nts) are retained.

Recommended kits are *mirVana*[™] miRNA Isolation Kit (Thermo Fisher Scientific) or Macherey-Nagel NucleoSpin miRNA kit (Clontech Laboratories).

2. RNA quality test

To improve NGS library construction and further increase sequencing performance, sRNAs should be size-fractionated from total RNA using gel electrophoresis or silica spin columns. RNA integrity and degradation are determined with the RNA integrity number (RIN) calculated using the Agilent 2100 BioAnalyzer (Agilent Technologies, CA). For plant samples a RIN value higher than eight is

usually recommended, but lower values can be acceptable provided that discrete RNA bands are visualized in the gel-like image displayed by BioAnalyzer (17).

The amount of starting material is important so the RNA source has to be measured preferentially using highly sensitive quantitation assays such as the Qubit 3.0 fluorometer (Thermo Fisher Scientific). The required amount of high quality RNA is 1 to 3 µg. Treatment with DNase is advisable as DNA contamination may produce inaccurate measurements of RNA concentration.

3. Library preparation

Most RNA-seq companies offer fast and cost-effective preparation of libraries for ultra-deep sequencing with low input requirements for purified total RNA. Different library preparation options include individual libraries and/or pooled barcoded libraries according to customer preferences and further downstream applications (16). Multiplexed-sequencing is the preferred option to process a large number of samples in a single flow cell lane, without drastically increasing cost or time. Multiplexing consists in the incorporation of individual barcode sequences to each sample during library preparation so they can be easily distinguished and sorted during data analysis.

If the NGS libraries are prepared in the lab, standard protocols for preparing directional sRNA libraries are described elsewhere (15, 16). The directional method is advisable since the sequencing always reads from the 5' to the 3' end of the original sRNA species. Alternatively, numerous kits for making sequencing oriented libraries are available commercially. For instance, TruSeq Small RNA Library Prep Kit (Illumina) or SMARTer smRNA-Seq Kit (Clontech Laboratories) allows size selection and multiplexing and are highly recommended. Libraries can be done from microgram to picogram of starting material but problems arise when the amount of input RNA is too low (200 ng of total RNA) as adapter dimers compete with the desired products in the RT-PCR reaction. Thus, quality assessments of libraries are of primary importance when PCR-based cloning techniques are used and should be done thorough the whole process. The quality of an sRNA library

can be assessed by a number of criteria. For instance, gel-based size selection of amplicons is critical due to the presence of adapter dimers and/or higher molecular RNA products created during the ligation steps.

4. Sequencing

Here we report the analysis of sRNA datasets obtained using massively parallel sequencing from Illumina, which utilizes an optimized four-color DNA sequencing by synthesis (SBS) technology (HiSeq 2500 System). In our test, we pooled four libraries in a multiplex format to obtain approximately 42 M of clean reads per sample. The recommended run type is 50 bp and single-ends (SE) that allow enough read length for accurate adapter trimming. In sRNA-seq, an important factor is the library size, which is the number of clean sequence reads for a given sample. For accurate profiling of sRNAs, the deeper the library is sequenced, the more sRNA species will be detected, including those expressed at very low levels, and more precise their quantification will be (4).

3.2. Pipeline 1: Raw-data analysis

This analysis is intended to parse small RNA reads from the raw sequence data, and to classify and characterize sRNA reads. sRNA reads from 18 to 26 nucleotides in length are parsed and trimmed from raw sequences upon identification of 3' adaptor sequences. Then low-quality, contaminated or background sequence reads are filtered out. Samples in our test were obtained from healthy, buffer (mock)-inoculated Arabidopsis plants (MOCK8) or from Arabidopsis plants infected with TRV (TRV8).

1. Obtaining small RNA-seq data in fastq format.

Download raw data directly from the website of the sequencing supplier or from public databases (see **Note 7**). The example data can be obtained from the Short Read Archive (SRA) under the accession numbers SRR6156924 and SRR6156925.

To directly download it, type from the working directory (here called "At_TRV"):

```
/At_TRV$ wget ftp://ftp-trace.ncbi.nih.gov/sra/sra-  
instant/reads/ByRun/sra/SRR/SRR615/SRR6156924/SRR6156924.sra
```

or

```
/At_TRV$ wget ftp://ftp-trace.ncbi.nih.gov/sra/sra-  
instant/reads/ByRun/sra/SRR/SRR615/SRR6156925/SRR6156925.sra
```

To transform SRA to fastq format use SRA toolkit (Materials, section 2.2.2, step 3):

```
/At_TRV$ fastq-dump SRR6156924.sra
```

The standard output should be:

```
Read 43279723 spots for SRR6156924.sra  
Written 43279723 spots for SRR6156924.sra
```

or

```
/At_TRV$ fastq-dump SRR6156925.sra
```

The standard output should be:

```
Read 44951890 spots for SRR6156925.sra  
Written 44951890 spots for SRR6156925.sra
```

Rename the new created fastq files by typing:

```
/At_TRV$ mv SRR6156924.fastq MOCK8.fq
```

or

```
/At_TRV$ mv SRR6156925.fastq TRV8.fq
```

2. Quality-control checkpoints

Specific controls should be applied at different stages of the analysis to ensure reproducibility and reliability of the results. Quality control for the raw data involves the analysis of sequence quality, the presence of adaptors, sequence content, GC content, N content, overrepresented k-mers, PCR

artifacts or contaminations by degraded RNA. This can be done using FastQC (**8**), which is a popular tool to perform quality control on Illumina reads (*see* **Note 7**). To run FastQC (Materials, section 2.2.2 step 4) from the command line, type:

```
/At_TRV$ fastqc MOCK8.fq TRV8.fq
```

The output delivered by this program includes several files for each fastq file analyzed. Quality results are showed in the html file that can be opened using a common web browser.

For sRNA reads it is important to pass the per base sequence quality, which should be higher than 30 in the full length of the sRNA sequence.

3. 3' adapter trimming.

Adaptor sequences should be trimmed from the raw data to obtain clean reads. This is a critical checkpoint in the workflow when a read length of 50 bp is used in the sequencing setting. In our example, the adapter sequence was already removed by the sequencing supplier, however, whenever required, adaptor sequences can be trimmed out using AdapterRemoval (**9**) (Materials, section 2.2.2 step 5 and **Note 8**).

4. Filtering by sRNA size.

vsRNAs are mostly of 20 to 26 nts in length and derive from Dicer-like-mediated cleavage of double-stranded RNA substrates. To filter reads by length, use the Perl script “trim.fastq.length.pl” (Materials, section 2.2.2 step 6):

```
/At_TRV$ perl perl_scripts/trim.fastq.length.pl -i MOCK8.fq -o MOCK8_20to26.fq -m 20 -x 26
```

Options:

-i <file.fq>	input file in fastq format
-o <file.fq>	output file in fastq format
-m <integer>	minimum length
-x <integer>	maximum length

The standard output (*see Note 9*) should be:

```
43279723 sequences evaluated
36331025 sequences within 20 bp to 26 bp saved
```

or

```
/At_TRV$ perl perl_scripts/trim.fastq.length.pl -i TRV8.fq -o TRV8_20to26.fq -m
20 -x 26
```

The standard output should be:

```
44951890 sequences evaluated
41393135 sequences within 20 bp to 26 bp saved
```

The output is a fastq file containing sRNA sequences from 20 to 26 nts in length for each input file (“MOCK8_20to26.fq” and “TRV8_20to26.fq”).

5. Removing structural sRNAs sequences.

A significant fraction of sRNA reads in the sequenced set maps to structural non-coding RNA such as ribosomal RNA, transfer RNA, small nucleolar RNA or small nuclear RNA. It is highly recommended to filter out these reads before proceeding with the analysis. To do so, map your reads against a local Rfam database using Bowtie (Materials, section 2.2.2, step 8 and section 2.2.3, step 1) by typing:

```
/At_TRV$ bowtie Rfam_index/Rfam -q MOCK8_20to26.fq -k 1 -v 1 --best -t --un
MOCK8noRfam.fq >MOCK_Rfam_maps.csv
```

Options:

-q <file.fq>	input file in fastq format
-k <integer>	reports <integer> good alignment per read
--best	hits guaranteed best stratum; ties broken by quality
-v <integer>	report end-to-end hits with <integer> mismatches
-t	print wall-clock time taken by search phases
--un <file.fq>	write unaligned reads to a new file

```
> <file.csv>          redirect the mapping results to a file
```

The standard output should be:

```
Time loading forward index: 00:00:01
Time loading mirror index: 00:00:01
Time for 1-mismatch full-index search: 00:05:40
# reads processed: 36331025
# reads with at least one reported alignment: 2429303 (6.69%)
# reads that failed to align: 33901722 (93.31%)
Reported 2429303 alignments to 1 output stream(s)
Time searching: 00:05:42
Overall time: 00:05:42
```

or

```
bowtie Rfam_index/Rfam -q TRV8_20to26.fq -k 1 -v 1 --best -t --un TRV8noRfam.fq
>TRV_Rfam_map.csv
```

The standard output should be:

```
Time loading forward index: 00:00:01
Time loading mirror index: 00:00:00
Time for 1-mismatch full-index search: 00:06:34
# reads processed: 41393135
# reads with at least one reported alignment: 680324 (1.64%)
# reads that failed to align: 40712811 (98.36%)
Reported 680324 alignments to 1 output stream(s)
Time searching: 00:06:35
Overall time: 00:06:35
```

(see **Note 10**)

The output files “MOCK_Rfam_maps.csv” and “TRV_Rfam_maps.csv” contain information of sRNA reads that aligned to Rfam sequences in tab-delimited format and can be removed to save

disk space. “MOCK8noRfam.fq” and “TRV8noRfam.fq” contain sRNA sequences without matching to Rfam database in fastq format.

The results of this pipeline are summarized in Table 1.

3.3. Pipeline 2. Sample infected with known viruses.

This pipeline is devoted to classify and characterize vsRNAs derived from viruses for which a reference genome is available.

1. Mapping against viral reference genomes.

To find the occurrences of a sRNA read along the viral reference genome, if no mismatches are allowed, reads are aligned against the genome sequence and mapping results are saved in a SAM format file. For viruses whose genomes are not fully resolved, EST or partial sequences can be used. In our test we confront our filtered sRNA reads against the TRV1 genome by typing:

```
/At_TRV$ bowtie virus_index/TRV1 -q MOCK8noRfam.fq -k 1 -v 0 --best -t --al  
MOCK8mapTRV1.fq -S MOCK8mapTRV1.sam
```

Options:

-q <file.fq>	input file in fastq format
-k <integer>	reports <integer> good alignment per read
--best	hits guaranteed best stratum; ties broken by quality
-v <integer>	report end-to-end hits with <integer> mismatches
-t	print wall-clock time taken by search phases
--al <file.fq>	write aligned reads to a new file
-S <file.sam>	write hits in SAM format

The standard output should be:

```
Time loading forward index: 00:00:00  
Time for 0-mismatch search: 00:01:37  
# reads processed: 33901722  
# reads with at least one reported alignment: 6871 (0.02%)
```

```
# reads that failed to align: 33894851 (99.98%)
Reported 6871 alignments to 1 output stream(s)
Time searching: 00:01:37
Overall time: 00:01:37
```

or

```
/At_TRV$ bowtie ./virus_index/TRV1 -q TRV8noRfam.fq -k 1 -v 0 --best -t --al
TRV8mapTRV1.fq --un TRV8noTRV1.fq -S TRV8mapTRV1.sam
```

The standard output should be:

```
Time loading forward index: 00:00:00
Time for 0-mismatch search: 00:04:23
# reads processed: 40712811
# reads with at least one reported alignment: 17450229 (42.86%)
# reads that failed to align: 23262582 (57.14%)
Reported 17450229 alignments to 1 output stream(s)
Time searching: 00:04:23
Overall time: 00:04:24
```

(see **Note 10** and Table 1)

The output files “MOCK8mapTRV1.fq” and “TRV8mapTRV1.fq” contain vsRNA sequences derived from TRV1 in fastq format. “MOCK8mapTRV1.sam” and “TRV8mapTRV1.sam” contain the alignment results in SAM file format.

2. Graphical representation of vsRNAs along viral genomes.

The spatial distribution of vsRNAs of different size and strand orientation across the viral reference genome is used to visualize vsRNA diversity. *MISIS (12)* shows sRNA reads along the viral genome as an histogram where the x-axis represents the nucleotide position of the 5' end (for sense vsRNAs) or 3' end (for antisense vsRNAs) and y-axis represents the number of reads starting at each genomic position.

To display MISIS (Materials, section 2.2.2, step 10) type:

```
/At_TRV$ java -jar appMISIS.jar
```

Click “No” to reject loading the demo data.

First, create a folder in the working directory called “misis_TRV” to save the project. To load the vsRNA alignment file, click on “File” in the toolbar menu and then click on “Load bam/sam”. In the emerging window called “Files selection”, select the path to the SAM file created in the previous step (/At_TRV/TRV8mapTRV1.sam). Save the project in “misis_TRV” and select the path to the fasta file with the virus reference genome (At_TRV/virus_index/TRV1.fasta). Finally, select the sizes of the vsRNAs (minimal size 20 and maximal size 26), and press “run” to start the analysis.

The output of this analysis comprises eight files. The four files with the “pm” annotation contain data of vsRNA mappings with no mismatches to the reference TRV1 genome and are the following:

“TRV1_table_pm_20_26.txt” is a table with the counting of mapping reads at each nucleotide position and for each specific size class and strand orientation.

“TRV1_project_pm_20_26.b2btf” contains the whole project information.

“TRV1_pm_table_logo_20_26.txt” contains information to create the logos of all mapped nucleotides.

“TRV1_pm_table_logo_first_nt_20_26.txt” contains information to create the logos of the first mapped nucleotides.

The remaining four files contain the same information to find the mappings of a vsRNA read, but up to two mismatches are allowed. In our example, these files should be identical to the corresponding “pm” files because we only allowed perfect matches to the reference viral genome in the Bowtie alignment.

To visualize the results click on “File” in the toolbar menu, then “Load Project” and select the file “TRV1_project_pm_20_26.b2btf“. The resulting histogram is shown in Fig. 1.

To compare vsRNA distribution between different samples, an additional table/project can be loaded in the same MISIS window. MISIS allows a broad range of visualization and analysis options. It is thus possible to magnify a certain area of the viral genome or modify the Y-axis scale if required, create a bar graph summary, display and export the reference viral sequence, load the logos (i.e. the nucleotide frequency at each genomic position), load and export the consensus reference viral sequence, display only specific vsRNA size classes or create graphical representation of the frequency of each nucleotide at the 5’ end. Furthermore, the counting table file generated by MISIS (“TRV1_table_pm_20_26.txt”) can be opened using a spreadsheet application such as Excel or Apache OpenOffice Calc for a thorough quantitative analyses.

3. Analysis of specific characteristics of the vsRNA dataset.

The study of distinctive attributes of the vsRNA dataset provides useful information to understand the origin and function of these molecules during the infection. For instance, the size distribution of total and unique sequence reads revealed the contribution of different DCL protein to vsRNA production. The distribution in strand polarity reflects the origin of vsRNAs from positive or negative viral strands and let us infer the secondary structure of the potential RNA precursors. And the identity of the 5’ nucleotide position of the vsRNAs is used to predict association with multiple effector silencing complexes that mediate different effects on RNA and chromatin (*18*).

In this section, we applied a number of analyses that can be used to illustrate specific features of the vsRNAs population.

3.1. Size distribution of total vsRNAs.

To obtain the frequency of total vsRNAs of different size classes, execute the Perl script “countfq_20to26.pl” (Materials, section 2.2.2 step 6):

```
/At_TRV$ perl perl_scripts/countfq_20to26.pl TRV8mapTRV1.fq
```

The standard output should be:

```
Parsing fastq file, please wait...
Total sequences analyzed = 17450229

length      reads      percentage
20 nts      1072073    6.14360418995074
21 nts      11379993   65.2140037818415
22 nts      3827160    21.9318611807329
23 nts      407482     2.33510975701236
24 nts      508438     2.9136465773601
25 nts      147687     0.846332732940066
26 nts      107396     0.615441780162312
```

This result can be copied to a spreadsheet application for graphical representation as shown in Fig.

2a.

3.2. Collapsing reads and obtaining size distribution of unique vsRNAs.

To collapse reads and obtain a fasta file containing unique vsRNA sequences, execute the Perl script “fqtofaunique.pl” (Materials, section 2.2.2 step 6):

```
/At_TRV$ perl perl_scripts/fqtofaunique.pl TRV8mapTRV1.fq
```

The standard output should be:

```
Parsing sequences, please wait...
17450229 total sequences evaluated
41670 unique sequences
```

The resulting file “TRV8mapTRV1-unique.fa” contains 41,670 unique sequences in fasta format and in alphabetical order where the header of each sequence is a random name followed by “_” and an integer that indicates the number of reads found (counts) for this specific vsRNA sequence.

For example, the first two fasta sequences of this file are:

```
>SRR6156925.419430_9
```

```
AAAAACTATGTGTATGTTT
>SRR6156925.3321940_4
AAAAAAGAATCAACCGTCTTC
```

These two vsRNAs have been sequenced nine and four times, respectively.

To obtain the size distribution of unique vsRNA sequences from this fasta file use the Perl script “countfa_20to26.pl” (Materials, section 2.2.2 step 6):

```
/At_TRV$ perl perl_scripts/countfa_20to26nts.pl TRV8mapTRV1-unique.fa
```

The standard output should be:

```
Parsing fasta file, please wait...
Total sequences analyzed = 41670

length      reads      percentage
20 nts      6694        16.0643148548116
21 nts      10161       24.3844492440605
22 nts      8215        19.7144228461723
23 nts      4715        11.3150947924166
24 nts      5481        13.1533477321814
25 nts      3318        7.9625629949604
26 nts      3086        7.40580753539717
```

This result can be copied to a spreadsheet application for graphical representation as shown in Fig. 2a.

3.3. Polarity distribution of total and unique vsRNAs.

The dataset generated by MISIS in step 2 is then used to examine the sense or antisense polarity distribution of total and unique vsRNAs of each size. To extract data from this file execute the Perl script “misis_table_analysis.pl” (Materials, section 2.2.2 step 6):

```
/At_TRV$ perl perl_scripts/misis_table_analysis.pl
misis_TRV/TRV1_table_pm_20_26.txt
```

The standard output should be:

sense vsRNAs

length	total	percentage	unique	percentage
20 nts	568554	6.62549577249614	3439	16.0505927377952
21 nts	5203849	60.6416972709862	5115	23.8728647437693
22 nts	2204903	25.6942621198058	4155	19.3923270792495
23 nts	191034	2.22616490149226	2476	11.5560533930738
24 nts	273287	3.18467878720078	2850	13.301596191543
25 nts	84392	0.983440164403899	1758	8.20498459815178
26 nts	55286	0.644260983614963	1633	7.62158125641744
Sum	8581305		21426	

antisense vsRNAs

length	total	percentage	unique	percentage
20 nts	503519	5.67734034027127	3255	16.0788381742739
21 nts	6176144	69.6380304984009	5046	24.9259039715471
22 nts	1622257	18.2914748170128	4060	20.0553250345781
23 nts	216448	2.44052153339007	2239	11.0600671803991
24 nts	235151	2.65140393581003	2631	12.9964433906343
25 nts	63295	0.713671692304501	1560	7.70598695909899
26 nts	52110	0.587557182810451	1453	7.17743528946848
sum	8868924		20244	
both	17450229		41670	

This result can be copied to a spreadsheet application for graphical representation as shown in Fig. 2b.

3.4. 5' end nucleotide composition of total and unique vsRNAs.

MISIS generates bar graphs in which the relative frequency of each 5' terminal nucleotide of different categories of vsRNAs is represented. This analysis can be done using full-length reference or in user-defined specific viral regions, including sense or antisense viral strands. However, it

could be interesting to have the raw data of the 5' end nucleotide composition of vsRNAs if further statistical analyses are required or for a custom graphical representation with a user-defined format using a spreadsheet application. This data can be extracted with the Perl script “nucomp_fq.pl” (Materials, section 2.2.2 step 6):

```
/At_TRV$ perl perl_scripts/nucomp_fq.pl TRV8mapTRV1.fq
```

The standard output should be:

```
Checking sequences...
```

```
Total sequences analyzed: 17450229
```

5'_end_nt	counts	percentage
A	6542630	37.4930896322335
T	5561653	31.8715187061442
C	2817851	16.1479313537948
G	2528095	14.4874603078275
N	0	0

This result can be copied to a spreadsheet application for graphical representation as shown in Fig. 2c.

Moreover, it is possible to report the nucleotide composition of vsRNAs of each size class by combining the output of the Perl script used to filter sRNAs by length in Pipeline 1 (section 3.2, step 4) (“trim.fastq.length.pl”) and the above Perl script. For example, type the following command to obtain the 5' nucleotide composition of total vsRNAs of 21 nts:

```
/At_TRV$ perl perl_scripts/trim.fastq.length.pl -i TRV8mapTRV1.fq -o  
TRV8_21nt.fq -m 21 -x 21 | perl perl_scripts/nucomp_fq.pl TRV8_21nt.fq
```

The output file “TRV8_21nt.fq” is a fastq file containing TRV1-derived vsRNAs of 21 nts in length.

The 5' nucleotide composition of vsRNAs of 21 nts should be (Fig. 2c):

```
Checking sequences...
```



```
Total sequences analyzed: 11379993
```

5'_end_nt	counts	percentage
A	4392230	38.5960694351921
T	3383110	29.7285771616907
C	1747991	15.3602115572479
G	1856662	16.3151418458693
N	0	0

To obtain the nucleotide composition of unique vsRNA reads use the Perl script “nucomp_fa.pl” (Materials, section 2.2.2, step 6):

```
/At_TRV$ perl ./perl_scripts/nucomp_fa.pl TRV8mapTRV1-unique.fa
```

The result, as shown in Fig. 2c, should be:

```
Checking sequences...
```

```
Total sequences analyzed: 41670
```

5'_end_nt	counts	percentage
A	12602	30.2423806095512
T	12886	30.9239260859131
C	8955	21.4902807775378
G	7227	17.3434125269978
N	0	0

3.4. Pipeline 3. Sample infected with unknown viruses

This pipeline is focused on the characterization of vsRNAs derived from viruses for which a previous reference genome is not available or from unknown viruses present in our plant sample. Given that vsRNAs produced in plants upon viral infections are overlapping in sequence, they can be assembled into long contiguous fragments of the viral genome and be used for virus identification.

1. Detection of viruses in a plant sample.

VirusDetect is a software package that analyzed large-scale sRNA datasets to perform reference-guided assembly as well as de novo assembly using the sRNA assembly program Velvet (**19**). The VirusDetect online tool can be used by following the next simple steps (*see Note 11*):

1. Register a user account in VirusDetect webpage: http://virusdetect.feilab.net/cgi-bin/virusdetect/vd_help.cgi.
2. Login at VirusDetect Online tool.
3. Upload the input fastq file. Use as input file the fastq file generated after removing non-coding RNA contaminants in Pipeline 1 (section 3.2, step 5 “TRV8noRfam.fq”).

Alternatively, fastq files can be uploaded via ftp:

```
/At_TRV$ ftp virusdetect.feilab.net  
Name: your_username  
Password: your_password  
ftp> put TRV8noRfam.fq
```

4. Upload a fasta file containing sequences of your plant reference transcriptome/genome in case it is not already loaded (named the file with the prefix “host_”). In our example, Arabidopsis is already in the list of pre-defined host, thus select it.
5. Analyze the data using the plant virus database at either 95, 97 or 100% of sequence identity at the nucleotide level with respect to the reference viral sequences (95% in this example).
6. Once the analysis is completed, download the resulting folder “result_TRV8noRfam.fq” and unpack it.

Output files:

“blastn.html” lists the genome reference viruses, identified by BLASTN against the National Center for Biotechnology Information (NCBI) database, that align with virus contigs generated upon assembly of vsRNA reads. In our test, the presence of TRV was confirmed in the TRV-infected sample named TRV8 with normalized sequencing depths (reads per million, RPM) of 162.6 and

152.4 for TRV1 and TRV2, respectively (Table 2). In addition, BLASTN identified several groups of contigs that were identical or highly homologous to the nucleotide genome sequence of an unexpected virus named *Turnip mosaic virus* (TuMV). Deep sequencing of the vsRNA library revealed TuMV-derived vsRNAs but at much lower normalized depth (0.7) (see **Note 12** and Table 2).

“blastx.html” lists the reference viruses, identified by BLASTX, that align with viral contigs generated upon assembly of the vsRNA sequences. In our example, viruses other than TRV or TuMV were not identified by BLASTX and consequently this file was not created.

“undetermined.html” lists contigs with no homologies to viral reference genome entries of the non-redundant databases of the GenBank. In this file, clusters of potential vsRNA contigs are highlighted in green. In our example, three contigs were selected as potential viral contigs with 21- and 22-nt vsRNA enrichment higher than 53% (data not shown)

“undetermined_blast.html” lists assembled contigs with hits in the virus reference database but that don’t fully satisfy the default filter options. This includes a coverage cutoff of a virus contig with respect to the reference virus sequences higher than 0.75, a coverage cutoff of a reported virus reference sequence with respect to the assembled virus contigs higher than 0.1 or a depth cutoff of a reported virus reference lower than 5. Potential novel virus contigs formed by high percentage of mapping vsRNAs of 21 and 22 nts are highlighted in green. In our analysis, three contigs with sizes ranging from 84 nts to 604 nts are found, of which two of them share sequence identity the RNA2 of TRV and *Pea early browning tobnavirus*, respectively (data not shown).

“blastn.reference.fa”, “blastx.reference.fa”, “contig_sequences.blastn.fa”,

“contig_sequences.blastx.fa”, “contig_sequences.fa” and “contig_sequences.undetermined.fa” are fasta files of the reference virus sequences identified in the analysis, contigs of assembled reads or contigs with hits to virus reference sequences, respectively.

“blastn.sam”, “blastx.sam”, “blastn.xls” and “blastx.xls” are sam format files and Excel files containing the alignment information of each contig.

Folders “blastn_references” or “blastx_references” contains html files with detailed alignment information of the virus contig to the reference virus sequences.

2. Analysis of vsRNA populations

Once all viruses infecting any given sample have been identified, proceed as described in Pipeline 2 to characterize the vsRNA population.

3.5. Pipeline 4. Analysis of virus-associated endogenous small RNAs

A significant fraction of the sRNA transcriptome is altered in plants exposed to viral attacks. This includes the massive accumulation of virus-associated endogenous siRNAs that reflects widespread silencing of host genes during viral infections (**20**). Pipeline 4 performs a comparative analysis of the populations of host-specific endogenous sRNAs between infected and non-infected plants.

1. Mapping sRNA reads against the plant reference transcriptome

Use TAIR (version 10) as cDNA reference sequence to search for Arabidopsis sRNAs that map to protein coding genes using Bowtie (see Materials, section 2.2.2, step 7 and section 2.2.3, step 3):

```
/At_TRV$ bowtie plant_index/TAIR10cdna -q MOCK8noRfam.fq -k 1 -v 0 --best -t --  
al MOCK8mapAt.fq -S MOCK8mapAt.sam
```

Options:

-q <file.fq>	input file in fastq format
-k <integer>	reports <integer> good alignment per read
--best	hits guaranteed best stratum; ties broken by quality
-v <integer>	report end-to-end hits with <integer> mismatches
-t	print wall-clock time taken by search phases
--al <file.fq>	write aligned reads to a new file

```
-S <file.sam>          write hits in SAM format
```

The standard output should be:

```
Time loading forward index: 00:00:00
Time for 0-mismatch search: 00:05:18
# reads processed: 33901722
# reads with at least one reported alignment: 19055755 (56.21%)
# reads that failed to align: 14845967 (43.79%)
Reported 19055755 alignments to 1 output stream(s)
Time searching: 00:05:18
Overall time: 00:05:18
```

or

```
/At_TRV$ bowtie plant_index/TAIR10cdna -q TRV8noRfam.fq -k 1 -v 0 --best -t --al
TRV8mapAt.fq -S TRV8mapAt.sam
```

The standard output should be:

```
Time loading forward index: 00:00:00
Time for 0-mismatch search: 00:05:44
# reads processed: 40712811
# reads with at least one reported alignment: 5212900 (12.80%)
# reads that failed to align: 35499911 (87.20%)
Reported 5212900 alignments to 1 output stream(s)
Time searching: 00:05:44
Overall time: 00:05:44
```

See Note 10.

The mapping results are summarized in Table 1.

2. Collapsing reads to unique sRNA sequences.

To collapse reads and obtain fasta files containing unique sRNA sequences use the Perl script

“fqtofaunique.pl” (Materials, section 2.2.2, step 5) as follows:

```
/At_TRV$ perl perl_scripts/fqtofaunique.pl MOCK8mapAt.fq
```

The standard output should be:

```
Parsing sequences, please wait...
19055755 total sequences evaluated
151497 unique sequences
```

or

```
/At_TRV$ perl perl_scripts/fqtofaunique.pl TRV8mapAt.fq
```

The standard output should be:

```
Parsing sequences, please wait...
5212900 total sequences evaluated
144491 unique sequences
```

The results are summarized in Table 1.

3. Size distribution of total and unique sRNA sequences.

Use the Perl scripts used in Pipeline 2 to obtain the size distribution of total reads or unique sequences (section 3.2, steps 1 and 3, respectively).

```
/At_TRV$ perl perl_scripts/countfq_20to26nts.pl MOCK8mapAt.fq
```

The standard output should be:

```
Parsing fastq file, please wait...
Total sequences analyzed = 19055755
length      reads      percentage
20 nts      3915193    20.5459872883546
21 nts      9296099    48.7836824098547
22 nts      1526113    8.00867244567324
23 nts      897877     4.71184164573904
24 nts      2463915    12.9300308489483
```

25 nts	567734	2.97933091604085
26 nts	388824	2.04045444538933

or

```
/At_TRV$ perl perl_scripts/count_fq_20to26nts.pl TRV8mapAt.fq
```

The standard output should be:

Parsing fastq file, please wait...

Total sequences analyzed = 5212900

length	reads	percentage
20 nts	810496	15.5478908093384
21 nts	3014825	57.8339312091159
22 nts	403928	7.74862360682154
23 nts	220534	4.23054345949472
24 nts	604162	11.5897485085077
25 nts	106014	2.03368566440945
26 nts	52941	1.01557674231234

or

```
/At_TRV$ perl perl_scripts/countfa_20to26nts.pl MOCK8mapAt-unique.fa
```

The standard output should be:

Parsing fasta file, please wait...

Total sequences analyzed = 151497

length	reads	percentage
20 nts	6363	4.20008316996376
21 nts	21260	14.0332811870862
22 nts	16779	11.0754668409275
23 nts	24990	16.4953761460623
24 nts	72465	47.8326303491158

25 nts	6303	4.16047842531535
26 nts	3337	2.20268388152901

or

```
/At_TRV$ perl perl_scripts/countfa_20to26nts.pl TRV8mapAt-unique.fa
```

The standard output should be:

```
Parsing fasta file, please wait...
```

```
Total sequences analyzed = 144491
```

length	reads	percentage
20 nts	6672	4.61758863873875
21 nts	27449	18.9970309569454
22 nts	18872	13.0610211016603
23 nts	22307	15.4383317992124
24 nts	61062	42.2600715615505
25 nts	5386	3.72756780699144
26 nts	2743	1.89838813490114

Data regarding size distribution of total reads and unique plant-derived sRNA sequences can be copied to a spreadsheet application for graphical representation as shown in Fig. 2d.

4. Graphical representation of sRNAs mapping to the reference transcriptome.

Use the IGV (**13**) (Materials, section 2.2.2 step 11) desktop application to visualize sRNA mappings against the reference transcriptome by typing:

```
At_TRV/$ java -Xmx750m -jar IGV_2.3.97/igv.jar
```

Options:

```
-Xmx(integer)m Start IGV with (integer) MB of memory.
```


In the “Genomes” tab of the menu tool bar, click on “Load Genome from file” and search for the Arabidopsis TAIR10 cDNA fasta file used to create the Bowtie index (see Materials, section 2.2.3, step 3) (“/At_TRV/plant_index/TAIR10_cdna_modif.fasta”). Then, open it.

IGV requires sRNA alignment files in BAM format. Prior to loading the BAM files into IGV it is necessary to sort them by creating index files using SAMtools (*11*) (Materials, section 2.2.2 step 9) as follows:

To transform SAM files to BAM format, type:

```
At_TRV/$ samtools view -b -h -S MOCK8mapAt.sam -o MOCK8mapAt.bam
```

or

```
At_TRV/$ samtools view -b -h -S TRV8mapAt.sam -o TRV8mapAt.bam
```

Options:

```
-b    output file in BAM format
-h    print header for the SAM file
-S    input is SAM format
-o    output file name
```

To sort the BAM files, type:

```
/At_TRV$ samtools sort MOCK8mapAt.bam MOCK8mapAt.sort
```

or

```
/At_TRV$ samtools sort TRV8mapAt.bam TRV8mapAt.sort
```

To create an index of the BAM files, type:

```
/At_TRV$ samtools index MOCK8mapAt.sort.bam
```

or

```
/At_TRV$ samtools index TRV8mapAt.sort.bam
```

Finally, to upload the sRNA-mapping BAM files select “File” in the toolbar menu and then “Load from file”, search for the sorted bam files generated previously

(“/At_TRV/MOCK8mapAt.sort.bam” or “/At_TRV/TRV8mapAt.sort.bam”) and open them.

Loading both samples at the same time is useful to compare sRNA abundance between non-infected and infected samples for a specific Arabidopsis transcript. As an example, Fig. 3 illustrates the output alignment of sRNAs from both MOCK8 and TRV8 samples along the Arabidopsis locus AT2G24850.1.

5. Quantification of endogenous sRNA abundance for each locus.

To count the number of total sRNAs produced at any given gene locus, use the eXpress program

(14) (Materials, section 2.2.2 step 12) as follows:

```
/At_TRV$ express-1.5.1-linux_x86_64/express plant_index/TAIR10_cdna_modif.fasta  
MOCK8mapAt.sam -o express MOCK8
```

or

```
/At_TRV$ express-1.5.1-linux_x86_64/express plant_index/TAIR10_cdna_modif.fasta  
TRV8mapAt.sam -o express TRV8
```

Options:

-o Sets the name of the directory in which eXpress will write all of its output

The following files will be created in the directories “express MOCK” and “express TRV”:

“params.xprs” and “results.xprs”.

The file “results.xprs” has a tab-delimited format and can be opened using a spreadsheet

application. This file contains the counting of sRNAs mapping for each locus, among other data.

Data from the second (“target_id”), third (“length”) and fifth (“tot_counts”) columns are used in

this analysis. “target_id” is the ID given to the transcript locus in the input fasta file, “length” is the

length in base pairs of each transcript sequence, and “tot_counts” is the number of sRNA mapping (uniquely or ambiguously) to each transcript.

Next, the Perl script “srna_count_comparison.pl” (Materials, section 2.2.2, step 6) is used to integrate the sRNA abundance from both samples in a new tab-delimited file. This script also calculates the fold change (FC) or the ratio between the sRNA counts per transcript in infected versus non-infected samples and the log₂FC (FCs lower than 1 become negative, while FCs higher than 1 become positives for graphical representation).

To execute the script, type the following in the command line:

```
/At_TRV$ perl perl_scripts/srna_count_comparison.pl express MOCK8/results.xprs  
express_TRV8/results.xprs MOCK TRV
```

The standard output should be:

```
#####  
Number of transcripts analyzed: 41671  
  
Number of transcripts without sRNAs mapping any sample, MOCK and TRV: 17863  
  
Number of transcripts with sRNAs mapping only to sample MOCK: 4402  
  
Number of transcripts with sRNAs mapping only to sample TRV: 5636  
  
Number of transcripts with sRNAs mapping to both samples: 13770  
  
#####
```

The job finished without any problem, congratulations!

The output file “MOCK&TRV_comparison.csv” has a tab-delimited format and can be opened with a spreadsheet application. It contains the following columns: “transcript_id”, “transcript_length”, “#_of_sRNAs_mappings_in MOCK”, “#_of_sRNAs_mappings_in TRV”, “Fold_change:TRV:MOCK” and “log₂FC”.

In our dataset, there are 186 Arabidopsis transcripts for which the abundance of sRNAs in the infected sample is higher ($\log_2FC > 2$) than in the non-infected sample, whereas 5,636 transcripts produce sRNAs in virus-infected plants but not in mock-inoculated ones (*see Note 14*).

Notes

1. All the analyses described in this chapter have been done using Ubuntu version 16.04. However, operating systems different to Linux can also be used by installing and executing a Virtual Machine (i.e. Virtualbox) in a Windows PC or MacOS X environment. Alternatively, MacOS versions are available for most of the programs used here, although the execution procedure and program installation may differ.
2. Generally, programs or applications are installed and executed using text-based commands in a Terminal (also called command line or command prompt). Press Ctrl+Alt+t to open a Terminal in Ubuntu 16.04. The Linux user must have root (or superuser) access to run administrative tasks (like installing programs). Here, “\$” or “`working_directory/$`” are used to indicate what has to be written in the command line in order to run a program or execute a Unix command. Knowledge of Linux/Unix command usage is not required, albeit advisable, to perform these analyses.
3. If you get any bug during compilation you may need to install the zlib and bzlib2 libraries by typing:


```
$sudo apt-get install libz-dev  
$sudo apt-get install libbz2-dev  
$sudo apt-get install zlib1g-dev
```
4. The system requires the complete location (or path) of the executable files to run the corresponding programs. To save time and circumvent potential setbacks during the execution of a program, it is highly recommended to install programs in the so-called working directory.

Alternative, it is possible to add a specific directory to \$PATH variable, which is an environment variable where the system search for the executable, by typing:

```
$ export PATH=/path_to_your_executable_file:$PATH
```

5. Non-coding RNA sequences from different organisms can be downloaded from other publicly available databases such as RNACentral (<http://rnacentral.org/>), Incrnadb (<http://lncrnadb.org/>), tRNADB-CE (<http://trna.ie.niigata-u.ac.jp/cgi-bin/trnadb/index.cgi>) or ncRNA database (<http://biobases.ibch.poznan.pl/ncRNA/>), among others.

6. From Rfam version 12 onwards the file “Rfam.fasta.gz” has been replaced with a folder called “fasta_files” containing fasta sequence files. To download all files of the current version (version 13, acceded October 2017), type:

```
/working_directory$ wget  
ftp://ftp.ebi.ac.uk/pub/databases/Rfam/CURRENT/fasta_files/RF*.fa.gz
```

To merge all into one unique file, type:

```
/working_directory$ gunzip RF*.fa.gz  
/working_directory$ rm RF*.fa.gz  
/working_directory$ cat RF*.fa > Rfam-CURRENT.fasta  
/working_directory$ rm RF*.fa
```

7. NGS QC toolkit can be used for data obtained from any other platform **(21)**.

8. Illumina adapter sequences are available at https://support.illumina.com/content/dam/illumina-support/documents/documentation/chemistry_documentation/experiment-design/illumina-adapter-sequences-1000000002694-02.pdf (Acceded on October 2017). The 3’ adapters RPI1 to RPI48 used by TruSeq Small RNA Library Prep Kit (Illumina) for library preparation are listed in the “RNA PCR Index Primers” section. The 3’ adapters D701 to D712 used by SMARTer smRNA-Seq Kit for Illumina (Clontech Laboratories) are listed in the “Illumina TruSeq HT index” section. To remove

the 3' adapter sequence using AdapterRemoval (Materials, section 2.2.2, step 5) type the following in the command prompt:

```
$ AdapterRemoval --file1 fastq_file.fq.gz --adapter1 "your_adapter_sequence"
```

The output of this program includes three files:

“your_output.discarded” contains sRNA sequences smaller than 14 nts in length.

“your_output.settings” contains log data, trimming statistics and sRNA length distribution.

“your_output.truncated” contains sRNA sequences larger than 15 nts in length.

Change the name of the files with another fastq file name before executing the program again in order to avoid overwriting.

Other software tools such Quadapt **(22)** or Trimmomatic **(23)** can be used to trim adaptor sequences or to discard low-quality reads.

9. The Standard output (or stdout) in Unix refers to the standardized data flow that is produced by command line programs. The default destination is the displayed screen of the computer terminal that executes the program. Here, the standard output of the programs used gives some information about the current status of the running program or directly the result of a specific command line program. The standard output can be easily redirected to a file by adding to the command line options “> file_name”.

10. The execution time may be different according to the computer capacity. Here we use a desktop computer with Core i7-6800K CPU, 32 GB of RAM and 6 threads.

11. Detailed information of the workflow followed by VirusDetect online tool is shown here:

http://virusdetect.feilab.net/cgi-bin/virusdetect/vdo_help.cgi. Alternatively, VirusDetect standalone program can be used to analyze your samples. See http://virusdetect.feilab.net/cgi-bin/virusdetect/vd_help.cgi, for information of system requirements, dependences, VirusDetect installation and usage.

Other tools such as VirFind (24) or ViRome (25) are also designed for virus discovery through vsRNA assembly or alignment against virus reference sequences, respectively.

12. Our sample referred to as TRV8 was harvested from plants that had been inoculated in our lab with an infectious TRV clone and maintained in the growth chamber until collection. We envisioned two possible scenarios to explain the presence of sRNAs with sequence homology to the TuMV genome in this sample: 1) wrong assignment to the TRV-infected library during sorting of multiplexed samples (26), 2) contamination during plant manipulation including RNA extraction and/or library preparation.

13. The eXpress software is available online, albeit updated versions are not expected in the future. Other tools such as Kallisto (27) can be used for quantification of sRNAs mapping to target transcripts. However, the output file of Kallisto should be modified to have exactly the same format as the output file of eXpress when using the Perl script “sRNA_count_comparison.pl”.

14. Introducing sample replicates is strongly recommended to minimize variability in the sRNA counts due to technical manipulation or biological variation and to increase reproducibility and robustness of the statistical tests (4).

Acknowledgments

This work was supported by National Research Grants BIO2012-39973, PCIN-2013-064 and BIO2015-70752 from Ministerio de Economía y Competitividad, Spain.

References

1. Ding SW and Voinnet O (2007) Antiviral immunity directed by small RNAs. *Cell* 130:413–426
2. Llave C (2010) Virus-derived small interfering RNAs at the core of plant-virus interactions. *Trends Plant Sci* 15:701–707
3. Head SR, Kiyomi Komori H, LaMere SA, et al (2014) Library construction for next-

generation sequencing: Overviews and challenges. *Biotechniques* 56:61–77

4. McCormick KP, Willmann MR, and Meyers BC (2011) Experimental design, preprocessing, normalization and differential expression analysis of small RNA sequencing experiments. *Silence* 2
5. Fahlgren N, Sullivan CM, Kasschau KD, et al (2009) Computational and analytical framework for small RNA profiling by high-throughput sequencing. 15:992–1002
6. Farazi TA, Brown M, Morozov P, et al (2012) Bioinformatic analysis of barcoded cDNA libraries for small RNA profiling by next-generation sequencing. 58:171–187
7. Leinonen R, Sugawara H, and Shumway M (2011) The sequence read archive. *Nucleic Acids Res* 39:2010–2012
8. Andrews S FastQC: A quality control tool for high throughput sequence data, <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/> (Accessed 27 Oct 2017)
9. Lindgreen S (2012) AdapterRemoval: easy cleaning of next generation sequencing reads. *BMC Res Notes* 5:337–343
10. Langmead B, Trapnell C, Pop M, et al (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol* 10:R25
11. Li H, Handsaker B, Wysoker A, et al (2009) The Sequence Alignment/Map format and SAMtools. 25:2078–2079
12. Seguin J, Otten P, Baerlocher L, et al (2014) MISIS: a bioinformatics tool to view and analyze maps of small RNAs derived from viruses and genomic loci generating multiple small RNAs. *J Virol Methods* 195:120–122
13. Robinson JT, Thorvaldsdóttir H, Winckler W, et al (2011) Integrative genomics viewer. *Nat Biotechnol* 29:24–26
14. Roberts A and Pachter L eXpress: Streaming quantification for high-throughput sequencing.

<https://pachterlab.github.io/eXpress/overview.html> (Accessed 27 Oct 2017)

15. Donovan WP, Zhang Y, and Howell MD (2011) Large-scale sequencing of plant small RNAs. *Methods Mol Biol* 744:159–173
16. Hafner M, Renwick N, Farazi TA, et al (2012) Barcoded cDNA library preparation for small RNA profiling by next-generation sequencing. *58*:164–170
17. Babu C. V. S and Gassmann M (2016) Assessing integrity of plant RNA with the Agilent 2100 Bioanalyzer System. *Agil Appl Note* 5990–8850E
18. Donaire L, Wang Y, Gonzalez-Ibeas D, et al (2009) Deep-sequencing of plant viral small RNAs reveals effective and widespread targeting of viral genomes. *Virology* 392:203–214
19. Zheng Y, Gao S, Padmanabhan C, et al (2017) VirusDetect: An automated pipeline for efficient virus discovery using deep sequencing of small RNAs. *Virology* 500:130–138
20. Cao M, Du P, Wang X, et al (2014) Virus infection triggers widespread silencing of host genes by a distinct class of endogenous siRNAs in Arabidopsis. *Proc Natl Acad Sci* 111:14613–14618
21. Patel RK and Jain M (2012) NGS QC toolkit: A toolkit for quality control of next generation sequencing data. *PLoS One* 7:e30619
22. Martin M (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. *17*:10–12
23. Bolger AM, Lohse M, and Usadel B (2014) Trimmomatic: A flexible trimmer for Illumina sequence data. *30*:2114–2120
24. Ho T and Tzanetakis IE (2014) Development of a virus detection and discovery pipeline using next generation sequencing. *Virology* 471–473:54–60
25. Watson M, Schnettler E, and Kohl A (2013) ViRome: An R package for the visualization and analysis of viral small RNA sequence datasets. *29*:1902–1903

26. Illumina, Inc. (2017) Effects of index misassignment on multiplexing and downstream analysis
27. Bray NL, Pimentel H, Melsted P, et al (2016) Near-optimal probabilistic RNA-seq quantification. Nat Biotech 34:525–527

Figure Captions

Fig. 1. Distribution of total vsRNA sequenced reads along the TRV1 genome. vsRNAs derived from the sense (blue) and antisense (red) strands are shown.

Fig. 2. **a**, size distribution of total and unique vsRNA reads. **b**, distribution of total and unique vsRNA reads of each size class derived from positive and negative viral strands. **c**, 5' end nucleotide composition of vsRNAs. **d**, size distribution of total and unique endogenous sRNA reads from mock-inoculated (MOCK8) and TRV-infected (TRV8) plants.

Fig. 3. sRNA mapping from mock-inoculated (MOCK8) and TRV-infected (TRV8) plants along the Arabidopsis transcript locus AT4G24690.1 using the IGV snapshot viewer. sRNAs derived from the sense (red) and antisense (blue) strands are shown.

Table Captions

Table 1. Summary of raw data analysis and mapping against viral and plant genomes.

Table 2. VirusDetect online tool results summary.

Tables

Table 1. Summary of raw data analysis and mapping against viral and plant genomes.

	MOCK8			TRV8		
	Total	%	Unique	Total	%	Unique
Sequence tags	43,279,723			44,951,890		
Size filtering (20-26 nts)	36,331,025	83.94%		41,393,135	92.08%	
Mapping with Rfam						
Mapped	2,429,303	6.69%		680,324	1.64%	
Mo mapped	33,901,722	93.31%		40,712,811	98.36%	
Mapping with TRV1						
Mapped	6,871	0.02%		17,450,229	42.86%	41,670
No mapped	33,894,851	99.98%		23,262,582	57.14%	
Mapping with TAIR10						
Mapped	19,055,755	56.21%	151,497	5,212,900	12.80%	144,491
No mapped	14,845,967	43.79%		35,499,911	87.20%	

Table 2. VirusDetect online tool results summary.

Reference	Length	Coverage (%)	Depth (Norm)	% Identity	Description
AF169561	9,835	7,617 (77.4)	0.7	99.93	Turnip mosaic virus
AF166084	6,791	6,771 (99.7)	162.6	99.59	Tobacco rattle virus RNA1
Z36974	3,855	1,642 (42.6)	152.4	99.43	Tobacco rattle virus RNA2

Figure 1

Figure 2

Figure 3