UNIVERSIDAD DE GRANADA

miSRA manual

Version 0.0.1

Michael Hackenberg
Ernesto Aparicio-Puerta

July 12, 2023

Contents

1 Overview

miSRA is a command line tool that allows fast and easy querying of uniformly preprocessed publicly available **small RNA sequencing (miRNA-seq)** datasets. miSRA can also generate expression profiles of miRNAs and other small RNAs using sRNAbench.

miSRA connects to an online service where all the alignments are performed, making it fast and lightweight on the user side in terms of computation and space required. The database currently contains a total of 89,984 preprocessed miRNA-seq data sets from over 900 species obtained from the Sequence Read Archive (see Table 1).

miSRA is a powerful and fast tool that will allow you to reanalyze hundreds of samples in a matter of minutes and quickly ask relevant questions about a particular SRA study such as level of contamination from a particular microorganism or expression levels of potential new miRNA candidates.

Different types of input reference sequences are supported:

- **microRNAs**: sRNAbench will be used for expression profiling of the user provided microRNAs within the user selected samples (SRX,ERX and DRX accessions).
- **Longer reference sequences**: sRNAbench will align reads from the selected samples to these reference sequences providing expression matrixes and mapping frequency as a function of position.
- **Short exact reference sequences**: sRNAbench will be used in 'spike-in' mode, i.e. only exact matches will be reported.

Potential applications:

- Generate an expression matrix for known microRNAs (miRBase, MirGeneDB, pMiREN, etc) for the samples of interest.
- Explore putative novel microRNAs. The expression in tissue, cell-types or even in different species can be explored along with a visualisation of the processing pattern. This way, important biogenesis features such as a high 5' heterogeneity can be analysed.
- Compare microRNA expression patterns among different species. miSRA provides 'seed-based' expression matrixes, i.e read counts and RPM values are summed up for microRNAs that have the same seed sequence. These seed-based matrixes from different species can be combined and patterns over different tissues can be compared.

- Quantify the frequency of RNA fragments originating from longer transcripts. An example would be to explore the differences in tRNA fragments among samples.

Species	Number of studies	Number of samples
Homo sapiens	1,642	41,443
Mus musculus	814	13,683
Arabidopsis thaliana	287	2,828
Drosophila melanogaster	203	2,073
Rattus norvegicus	171	3,134
Bos taurus	147	2,602
Caenorhabditis elegans	137	1,947
Sus scrofa	137	1,435
Gallus gallus	87	742
Zea mays	82	917
Solanum lycopersicum	70	493
Danio rerio	51	508
Oryza sativa	47	343
Glycine max	43	717
Triticum aestivum	38	398

Table 1: 15 species with the most available studies

2 Installation

2.1 Dependencies

miSRA requires Python >= 3.7 and the python package *requests* (automatically installed by *pip*)

2.2 Option 1: Install with pip (recommended)

Create a virtual environment and activate it (optional but recommended)

```
python3 -m venv env
source env/bin/activate
```

Then install with *pip*

```
pip3 install miSRA
```

Test that it worked

```
miSRA --help
```

2.3 Option 2: Install manually

Alternatively, you could clone the project, install the requirements and make an alias to the script.

```
git clone https://github.com/bioinfoUGR/miSRA.git
cd miSRA
pip3 install -r requirements.txt
alias miSRA='python3 /absolute/path/to/miSRA.py'
```

If you do not want to add an alias, miSRA is a stand-alone script so it should work all the same by simply:

```
python3 /absolute/path/to/miSRA.py
```

3 Quick start

3.1 Installation

Create a virtual environment and activate it, then install with *pip*

```
python3 -m venv env
source env/bin/activate
pip3 install miSRA
```

3.2 Example run

In this example we are going to profile the expression of some human microRNAs on all RNAatlas (SRP225193) samples. Download the following files and save them into a directory (e.g., miSRA_example):

- *mature.fa*: Input mature microRNA annotations in fasta format.
- *hairpin.fa*: Input hairpin microRNA annotations in fasta format.
- *config.json*: A config *.json* file specifying all necessary parameters to perform the profiling.

```
mkdir miSRA_example
cd miSRA_example

wget https://raw.githubusercontent.com/bioinfoUGR/mi
SRA/main/examples/mirna/config.json

wget https://raw.githubusercontent.com/bioinfoUGR/miSRA/m
ain/examples/mirna/hairpin.fa

wget https://raw.githubusercontent.com/bioinfoUGR/miSRA/ma
in/examples/mirna/mature.fa
```

Or just download the files manually. Then, from that directory, run miSRA:

```
miSRA --config config.json
```

config.json contains all parameters needed to perform the profiling:

```
{
"mode":"mirna",  # There are different modes to query
# miSRA (mirna, libs and spike). The mode mirna performs
# alignments to miRNA annotations using sRNAbench

# mirna mode requires 2 miRNA annotation files
# one for mature miRNAs and one for hairpins
"mature":"mature_hsa.fa", # path to mature miRNA
   annotations in fasta format
"hairpin":"hairpin_hsa.fa", # path to hairpin miRNA
   annotations in fasta format

# you can specify which samples you want to profile
# either by specifying comma-separated SRA study
"studies":"SRP225193",
# or experiment accessions
# "experiments" :
# "SRX2349199,SRX2349197,SRX546025,SRX546026",

"localOut":"RNAatlas", # local folder where the results
   will be downloaded to
"mm": "1", # number of mismatches (optional)
}
```

Besides the paths to the files, previously downloaded, the config file should

also specify at least which SRA experiments or studies to profile. To do so, use the parameters *studies* and *experiments*.

While running, miSRA will report a job ID. You can use the job ID to retrieve your results if the connection is interrupted.

```
miSRA -j YOUR_JOB_ID_XYZ
# use this command to retrieve an example precalculated job
miSRA -j idWKJ3S5JYB7JNVM2
```

After your job is completed, results will be downloaded to the specified destination directory, in the example *RNAatlas*.

3.3 Example results

Most users will be interested in one of the following:

- *results.html*: a brief html report containing basic statistics and links the main results.
- *results.xlsx*: An Excel book containing the expression matrixes. Each of the matrixes: Read Count (RC), Reads Per Million (RPM) and Reads Per Million normalized to reads mapped to that annotation (RPMlib) can also be found as tab-separated text files in the output directory.

Other output files are generated. For a comprehensive description, please read Output files.

4 General usage

Expression profiling using miSRA comes in three 'flavours': *microRNA* (or miRNA-like) sequences, longer *libraries* to profile RNA fragments (tRNA, rRNA) or even complete microbial genomes and *exact matches* to short input sequences. To launch any job, you need to compose a config file in json format and provide its path to miSRA like this:

```
miSRA --config config.json
```

The standard config would be something like:

```
{
"mode":"mirna",
"studies":"SRP225193",
"mature":"mature.fa",
"hairpin":"hairpin.fa",
"localOut":"output_directory",
"mm": "1",
}
```

The parameter *mode* determines the profiling approach. Each of the 3 modes are explained in more detailed in the following sections.

4.1 Expression profiling: microRNA mode

For this kind of analysis, users should provide one fasta file with mature miRNA sequences and one file with miRNA precursor sequences.
The expression profiling of the mature microRNAs will be performed as described here[1]. Briefly, for a given sample, all reads will be mapped first against the precursor sequences. Those mapping within a window of (start - 3nt) and (end + 6nt) of the mature sequences will be assigned to them.

To use this mode, include the parameter *'mode': 'mirna'*. The input *config.json* should include the following parameters:

- ***mature***: path to mature miRNA annotations in fasta format
- ***hairpin***: path to precursor miRNA annotations in fasta format
- ***studies***: a comma-separated list of all SRA studies (e.g., SRP12345) to be profiled ***AND/OR***
- ***experiments***: a comma-separated list of all experiments (e.g., SRX12345) to be profiled

You can also include other optional parameters, they are explained below.

[1] https://arn.ugr.es/srnatoolbox/static/sRNAbench_article.pdf

4.2 Expression profiling: Transcript mode

In transcript mode, SRA samples reads will be aligned to user-provided references (for example tRNA, snoRNA, snRNA, rRNA, yRNA or even cDNAs to identify mRNA fragments) using sRNAbench *libs* option. Following read quantification, raw and normalized expression matrixes will be generated.

To use this mode, include the parameter *'mode': 'libs'*. A lighter faster option, that does not generate visualizations, is also available via *'mode': 'libsG'*. The input *config.json* should include the following parameters:

- ***libs***: path to reference annotations in fasta format
- ***studies***: a comma-separated list of all SRA studies (e.g., SRP12345) to be profiled ***AND/OR***
- ***experiments***: a comma-separated list of all experiments (e.g., SRX12345) to be profiled

You can also include other optional parameters, they are explained below.

4.3 Expression profiling: Exact mode

Rather than mapping SRA experiments or study reads to the reference annotation, miSRA will merely quantify exact reference matches without any mismatches. This mode uses the corresponding sRNAbench parameter *'spike-in'*, originally designed to account for spiked-in oligonucleotides.
The provided reference sequences may correspond to small RNAs like the canonical mature miRNA, as well as specific miRNA isoforms, specific tRNA fragments, etc. Potential applications include the quantification of tRNA fragments or read-level isomiRs (miRNA variants).

To use this mode, include the parameter *'mode': 'exact'*.
The input *config.json* should include the following parameters:

- ***spikeFile***: path to reference annotations in fasta format
- ***studies***: a comma-separated list of all SRA studies (e.g., SRP12345) to be profiled ***AND/OR***
- ***experiments***: a comma-separated list of all experiments (e.g., SRX12345) to be profiled

You can also include other optional parameters, they are explained below.

4.4 Expression profiling: Optional parameters

Besides the parameters required for each profiling mode, there are several optional parameters that allow users to tailor their job:

- ***localOut***: Local directory to download the profiling results. If not provided, a default directory named "miSRA_results" will be created in the current working directory.
- ***mm***: Number of mismatches allowed for the alignment. Default value is 0.
- ***minRC***: An integer number to define a minimum read count threshold to consider a read. E.g., minRC=5 would remove from the analysis all reads with fewer than 5 copies in the sample. Default value is 0.

4.5 Obtaining sequencing reads: download mode

miSRA works on reads previously preprocessed by sRNAbench. In short, the library preparation protocol is derived from a subset of reads, then the adapters are trimmed accordingly and identical reads are collapsed. These collapsed reads files are also accessible to users by means of the ***download*** mode.
The command to launch miSRA in this mode is the same as for profiling modes and the *config.json* should contain:

- ***"mode" : "download"***
- ***studies***: a comma-separated list of all SRA studies (e.g., SRP12345) to be downloaded ***AND/OR***
- ***experiments***: a comma-separated list of all experiments (e.g., SRX12345) to be downloaded

miSRA will download a zipped directory containing the corresponding collapsed reads in ***.fasta.gz*** format using sRNAbench's convention. The fasta header of each sequence includes the read count after a "***#***" symbol. Reads are sorted by abundance in decreasing order. An example collapsed file would look like this:

```
##read_id#read_count
        [...]
>18563#376755
GGCTGGTCCGATGGTAGTGGGTTATCAGAACT
```

```
>545242#210627
TGTGGGTGCTTGTGGAGTCAGACTGATAGT
>1198085#116543
TGTGGGTGCTTGTGGAGTCAGACTGATAGTCAACA
        [...]
```

4.6 Obtaining live database statistics

You can download summary statistics, namely, per sample number of studies and number of samples:

```
miSRA --db-stat
```

You can also download a list of all samples/studies for a given species using its NCBI's taxon ID:

```
miSRA --taxonID TAXON_ID
# for instance, for Homo sapiens
miSRA --taxonID 9606
```

5 Output files

The expression profiling of each sample is performed by a 'light' version of sRNAbench, so most output files are identical to sRNAbench profiling results. User-defined samples will be summarised by means of read count and expression matrixes (RPM normalised read count values). Additionally, the processing pattern will be determined by read frequency as a function of mapping position for microRNA and long reference sequences.

5.1 Summary output files

- *results.html*: a brief html report containing basic statistics and links the main results.
- *results.xlsx*: An Excel book containing the expression matrixes. Each of the matrixes: Read Count (RC), Reads Per Million (RPM) and Reads Per Million normalized to reads mapped to that annotation (RPMlib) can also be found as tab-separated text files in the output directory.

5.2 Read count matrices

Multiple-mapping adjusted read count matrices are generated for transcript and microRNA modes (*"mode:libs"* / *"mode:mirna"*).The normalization approach can be recognized by their filenames. In exact mode, files end with **RC.tsv* since multiple mapping is not allowed.

- ****RCadj.tsv***: Multiple-mapping adjusted read count.
- ****RC.tsv***: Read count (exact mode).

5.3 Normalized expression matrixes

- ****RPMlib.tsv***: Read count matrices are normalized using the total number of reads mapped to the user-provided reference sequences. For example ***matrix_mature_RPMlib.tsv*** for microRNAs.
- ****RPM_total.tsv***: Read count matrices are normalized using the total number of clean reads in the sample. For example ***matrix_mature_RPM_total.tsv*** for microRNAs.

5.4 Read level profiles

Read counts and RPM values are reported for every detected read and all samples. These output files are not provided in exact mode because files are already read level.

- ***readLevel_RC.tsv***: Raw read counts
- ***readLevel_RPMtotal.tsv***: Read Per Million normalized values

5.5 Processing pattern

Per position read coverage for each sequence can be found in the ***mappingsPerPos*** subdirectory. This subdirectory is only created in *"mode:libs"* and *"mode:mirna"*). Three position-matrix files are generated for each reference sequence:

- ****_RPMtotal.tsv***: The RPM values for each position (lines) in the reference sequence for all samples (columns)
- ****_RCperc.tsv***: The percentage of reads for each position (lines) in the reference sequence for all samples (columns)

- ****_RC.tsv***: The raw read count values (not normalized) for each position (lines) in the reference sequence for all samples (columns)

5.6 Seed expression (microRNA mode only)

Read counts and RPM values of the microRNAs are determined at a miRNA family level, i.e. expression values are groupped by the seed sequence. These values can be used for across species expression comparison.

6 Code, repository and issues

The latest version of the code can be found in our public GitHub repository. You can also provide feedback using the issues page.

7 FAQ

I launched a profiling job but the connection was lost. How can I retrieve my results?
Once a job is in the queue, you will receive a jobID. You can use this jobID to retrieve your results:

```
miSRA -j YOUR_JOB_ID_XYZ
# use this command to retrieve an example precalculated job
miSRA -j idWKJ3S5JYB7JNVM2
```

Which species are available on miSRA?
The number of different species currently available (*July, 2023*) is over 900. For a comprehensive list, please do:

```
miSRA --db-stat
```

Our goal is to profile **EVERY** miRNA-seq data set publicly available on SRA, so please if some sample/species is not present let us know.
Annotation and references of other species (including microbial genomes) not included in SRA can also be used to profile SRA samples.

How are the reads mapped?
There are 3 profiling modes, they all rely on sRNAbench. For more details, check the corresponding section.

Can I request a new dataset?
If a miRNA-seq data set from SRA is not present in miSRA, we are happy to learn about it. Please get in touch including the SRA project accession (SRPXYZ) and the NCBI's taxon ID of the organism. Our goal is to profile **EVERY** miRNA-seq data set publicly available on SRA.

I do not want or cannot use pip for some reason, can I still run miSRA?
Yes, miSRA is a stand-alone python script that sends queries to our remote database. Feel free to download it from our GitHub repository and make it work in whatever way you find more convenient.

Is querying/profiling/downloading limited?
The tool is currently limited to profile/download 500 samples at a time. For queries larger than that, please organize it in batches and merge the tables locally. If you systematically need access to larger queries get in touch and we will address your petition.
Each IP address is also limited to 100 queries/hour at the moment. This number could change at any time depending on the usage detected.

8 Related software

- ***sRNAbench***: miSRA heavily relies on sRNAbench, both to generate the preprocessed reads and to perform new profilings. sRNAbench was first described here, but has seen significant improvements since its original publication in 2019[2] and 2022[3]. You can also access sRNAbench as a web server or as a stand-alone tool distributed via docker containers.
- ***mirnaQC***: Although miSRA does not use mirnaQC[4] for quality control, its quality control process and thresholds are heavily inspired by it. If you are interested in miRNA-seq research, you should consider giving it a try.

[2] https://doi.org/10.1093/nar/gkz415
[3] https://doi.org/10.1093/nar/gkac363
[4] https://doi.org/10.1093/nar/gkaa452