



A reliable unsupervised method for detecting miRNA binding sites on mRNAs

April 2010 Release

User Guide

1. Introduction

miRiam is a fast and reliable tool for screening miRNA databases in order to identify potential interactions with a given mRNA. It is based both on thermodynamics features and empirical rules coming from known interactions.

miRiam accept a mRNA sequence as input and returns the list of possible interactions between the sequence and miRNAs from a selected database. By default, miRBase is used, but users can also choose to provide their own list of miRNAs.

Although the default settings are optimized for standard usage, miRiam allows users to specify several parameters, concerning target accessibility evaluation and empirical rules application.

2. miRiam installation

The miRiam package contains:

- miRiam files
- Vienna RNA Package 1.6.1
- miRBase DB

All these components are necessary. You will also need Python interpreter. It usually comes with every Linux distribution. You can get the latest version at www.python.org.

For a correct installation, please follow these steps:

A) Install the Vienna RNA Package.

- Change to the Vienna RNA Package directory:

```
cd ViennaRNA-1.6.1
```

- Then type:

```
./configure  
make
```

and

```
make install (as root)
```

C) Install miRiam:

- Change to miRiam directory:

```
cd miriam
```

- The directory contains several python files, needed for the correct execution of miRiam, the latest *miRBase* distribution, a directory containing some example files, and a file called *install.sh* which must be executed in order to correctly install miRiam:

```
./install.sh
```

This command will create the following directories:

- mircache
This directory will store computed secondary structure information, in order to speed up further screening on the same sequences.

- mirdb
This directory contains the whole miRBase divided per organism in different files.

- Now you are ready to use the tool!

3. miRiam standard usage

The easiest way to use miRiam is by the following command line:

```
./miriam source=... target=... *
```

where *source* is a short code that indicates the organism-specific subset of miRBase to be screened (Ex. Homo sapiens=hsa, Mus musculus=mmu, ...), and *target* is the name of a file which contains the target mRNA sequence in Fasta format.

Example

```
./miriam source=hsa target=bcl2.fasta *
```

This will perform a complete search for potentially accessible binding sites on *bcl2* 3' UTR for all the Homo sapiens miRNAs in miRBase, returning the corresponding miRNA/mRNA duplexes.

3'UTR start position will be obtained by computing the longest ORF. The output file will be *bcl2_miriam.txt*.

(The *bcl2.fasta* example file is located in the `examples` directory)

* If the command does not work, try:

```
python miriam ...
```

4. miRiam advanced usage

A number of useful options are listed below:

- mir=...** Input miR. It corresponds to an official miRNA identifier. If not provided, the whole miRBase screening (for the specified organism) will be performed.
Ex. mir=miR-15a
- mirdb=...** It allows users to use their own miRNA databases. A file in the right format must be provided. (See *Appendix A*). If provided, the source value is ignored.
- thr=...** Pairing threshold. Accepted Values:
- 'low' (1/4 of the maximum pairing score)
- 'medium' (1/2 of the maximum pairing score) (default)
- 'high' (3/4 of the maximum pairing score)
- absthr=...** Absolute Pairing threshold. If provided, *thr* is ignored.
- wlen=...** Sliding window length. It is the length of the candidate binding regions.
Accepted Values: 22...36 (Default: 32)
- out=...** Output file name. If not provided, the default filename will be:
'mRNA_name_miriam.txt'
- subseq=...** Target region to be considered for interaction search. Accepted Values:
- '3utr' (3' UTR region) (default)
- 'all' (the whole mRNA)
- utr=...** 3' UTR Start position. If not provided and *subseq=3utr*, the longest ORF rule will be considered in order to find the beginning of 3' UTR.
- filter=...** Filter type. It specifies the mismatch and G:U tolerance in the seed region. Accepted values:
- 'strict' (2 mismatches and 1 G:U allowed) (default)
- 'relaxed' (3 mismatches and 2 G:U allowed)
- 'free' (free filter. Rules are specified by *mis* and *gu* parameters)
- mis=...** Maximum number of mismatches in 5' region (free filter only). Accepted values:
0...8
- gu=...** Maximum number of G:U wobbles in 5' region (free filter only). Accepted values:
0...8
- fivelen=...** Length of 5' region for filter rules application (free filter only). Accepted values:
7...10 (Default: 7)
- first=...** It specifies if first nucleotide of miRNA has to be considered by the filter. Accepted Values:
- 'yes'
- 'no' (default)
- plotfile=...** Output file with region pairing probabilities. If not provided, the plotfile won't be generated.

Example 1

```
./miriam source=mmu target=hoxb8.fasta thr=low utr=990 filter=free mis=0 gu=1  
fivelen=9 out=hoxb8_mmu.txt
```

This will perform a complete search for most potentially accessible binding sites on *hoxb8* 3' UTR for all the Homo sapiens miRNAs in miRBase, returning the corresponding miRNA/mRNA duplexes. The 3'UTR start position is given in input. A free filter scheme is used, allowing only 1 G:U wobble and no mismatches in the seed region of 9 nucleotides.

Example 2

```
./miriam source=hsa mir=miR-15a target=bcl2.fasta filter=relaxed subseq=all
```

This will search for potentially accessible binding sites on the whole *bcl2* sequence for the Homo sapiens miRNA *miR-15a*, returning the corresponding miRNA/mRNA duplexes. A relaxed filter is used, allowing up to 3 mismatches and 2 G:U wobbles in the seed region of 7 nucleotides. The output file will be *bcl2_miriam.txt*.

Example 3

```
./miriam mirdb=my_mirnas.txt target=lin28.fasta
```

This will perform a complete search for potentially accessible binding sites on *lin28* 3' UTR for all the miRNAs in the provided list (*my_mirnas.txt* – See Appendix A for the right format) returning the corresponding miRNA/mRNA duplexes. 3'UTR start position will be obtained by computing the longest ORF. The output file will be *lin28_miriam.txt*.

(The *hoxb8.fasta*, *lin28.fasta* and *my_mirnas.txt* example files are located in the `examples` directory)

5. miRBase update

The miRBase files can be easily updated. To do so, download the latest version from <http://microrna.sanger.ac.uk/sequences/ftp.shtml>, put the *mature.fa* file in the miRiam directory (by default `miriam`), and run the *DBinit* routine:

```
./Dbinit mature.fa
```

This will cause the update of miRBase files in the *mirdb* directory. All previous files will be overwritten by the new ones.

Appendix A – Specification of *mirdb* format

A custom miRNA database can be provided via the *mirdb* option. The file must be in the following format:

```
Source Organism
# of miRNAs
miRNA1-name
miRNA1-seq
miRNA2-name
miRNA2-seq
...
miRNAn-name
miRNAn-seq
```

Here is an example:

```
Homo sapiens
4
let-7a
UGAGGUAGUAGGUUGUAUAGUU
let-7b
UGAGGUAGUAGGUUGUGUGGUU
let-7c
UGAGGUAGUAGGUUGUAUGGUU
let-7d
AGAGGUAGUAGGUUGCAUAGU
```

Credits and contacts

miRiam is developed and maintained by Ferrolab, the Bioinformatics and Data Mining Group at University of Catania.

Home Page

<http://ferrolab.dmi.unict.it>

Contacts

Dr. Alessandro Laganà: lagana@dm.unict.it - Prof. Alfredo Ferro: ferro@dm.unict.it