miARma-Seq: **mi**RNA-Seq **A**nd **R**NA-Seq **M**ultiprocess **A**nalysis tool.

mRNA detection from RNA-Seq Data User's Guide

Eduardo Andrés-León, Rocío Núñez-Torres and Ana M Rojas.

Instituto de Biomedicina de Sevilla (IBIS), Hospital Universitario Virgen del Rocío/CSIC/Universidad de Sevilla, Computational Biology and Bioinformatics Group, Seville, Spain.

First edition 1.1 March 2016

Table of Contents

1. INTRODUCTION

miARma-Seq is a comprehensive pipeline analysis for RNA-Seq and miRNA-Seq data suited for mRNA, miRNA and circRNA identification and differential expression analysis of any organism with a sequenced genome. Briefly miARma-Seq integrates quality-control analysis of raw data (fastqc), trimming of the reads, with adapter sequence prediction if necessary, alignment of the reads with the correspondent genome reference, entities quantification, differential expression analysis, miRNA-mRNA target prediction, miRNA-mRNA inverse expression pattern analysis and functional analysis to detect the enrichment of metabolic pathways and gene ontologies for mRNAs. All these steps can be executed as a whole pipeline or as separated steps. To make easier the execution of single steps, miARma-Seq has been implemented with a Perl based module structure.

This guide gives a tutorial-style introduction for the practical use of miARma-Seq but does not describe every feature of the pipeline. A full description of every feature is given by the individual function help documents available in our website (http://miarmaseq.cbbio.es/Documentation/). It includes explanations of command-line options for each type of analyses to give an idea of basic usage. Input and output file formats are also detailed. Also, many examples of use are given.

This document does not try to explain the underlying algorithms or data-structures used in miARma-Seq. For these issues, we recommend to consult the information available in the webpages of the software integrated in miARma-Seq.

2. PRELIMINARIES

2.1. Pre-requisites

miARma-Seq is a tool that provides an easy and common interface to various analysis software. It also intends to reduce to the minimum the number of dependencies. Nevertheless, some basic programs listed below must be correctly installed:

1. Perl v5.6.0 or higher**.** http://www.cpan.org/src/5.0/perl-5.6.1.tar.gz
2. R environment v.3.0 or higher. http://www.r-project.org/
3. Java v.1.6. or higher. http://www.java.com/.
4. Bioconductor v.1.3 or higher. http://www.bioconductor.org/install/
5. Compilers:
    a. Xcode for Mac:
    https://itunes.apple.com/es/app/xcode/id497799835?l=en&mt=12
    b. For Linux:
        i. Gcc: https://ftp.gnu.org/gnu/gcc/
        ii. make: http://ftp.gnu.org/gnu/make/

2.2. How to get help

This user guide will hopefully answer most questions about miARma-Seq. Note that each module in miARma-Seq has its own help page (http://miarmaseq.cbbio.es/Documentation). If you have a question about any particular function, reading the module's help page will often answer the question very quickly. Nevertheless, if you've run into a question, which isn't addressed by the documentation, or you've found a conflict between the documentation and software itself, then you can visit our help & contact web page at http://miarmaseq.cbbio.es/help.

In addition, the authors of miARma-Seq always appreciate receiving reports of bugs in the pipeline modules or in the documentation. The same goes for well-considered suggestions for improvements. For these issues please contact at: miARma-devel@cbbio.es.

3. QUICK START

3.1. miARma installation instructions

Latest installation instruction for Linux, Mac and Windows, can be found in our web page at http://miarmaseq.cbbio.es/installation. If you are using a Unix system, the recommended procedure is the following:

1. Create a directory to install miARma, (eg : NGS) and download the software :

```
$> mkdir NGS
$> cd NGS/
NGS> curl -L -O https://bitbucket.org/cbbio/miarma/get/master.tar.bz2
```

2. Extract miARma binaries and libraries:

```
NGS>tar -xjf master.tar.bz2
NGS>cd cbbio-miARma-*
cbbio-miarma>ls -l
  Examples
  README.md
  bin
  lib
  miARma
```

3.2. mRNA Example installation instructions

1. Inside miARma folder, download data:

```
2.  miARma>  curl -L -O
    https://sourceforge.net/projects/miarma/files/Examples/Examples_miARma_mRNAs.tar.bz2
```

3. Uncompress it :

```
miARma>tar -xjf Examples_miARma_mRNAs.tar.bz2
```

4. Check the parameters (optional step):

```
miARma>perl miARma Examples/basic_examples/mRNAs/1.Quality/1.Quality.ini --check
```

5. Execute the examples:

```
miARma>perl miARma Examples/basic_examples/mRNAs/1.Quality/1.Quality.ini
```

3.3. Other needed data for miARma execution

miARma uses topHat tool for read alignment, which can use bowtie1 or bowtie2 algorithms. For the mRNA example included in miARma, human hg19 genome index for Bowtie 1 and Bowtie 2 are needed.

3.3.1. Bowtie1 index installation:

1. Downloading from miARma folder:

```
miARma>curl -L -O http://miarmaseq.cbbio.es/download/Genome/Index_bowtie1_hg19.tar.bz2
```

2. Extracting:

```
miARma>tar -xjf Index_bowtie1_hg19.tar.bz2
```

3.3.2. Bowtie2 index installation:

1. Downloading from miARma folder:

```
miARma>curl -L -O http://miarmaseq.cbbio.es/download/Genome/Index_bowtie2_hg19.tar.bz2
```

2. Extracting

```
miARma>tar -xjf Index_bowtie2_hg19.tar.bz2
```

4. mRNA-SEQ ANALYSIS

miARma-Seq presents a highly flexible modular structure to perform the different stages of the mRNA analysis. In this section, each module will be extensively described, including the description of the input and output files, the different parameters for the analysis and the creation of the configuration file to execute it.

In order to better explain mRNA-Seq analysis, data from GEO (GEO code: GSE37376) will be used (this data can be downloaded from GEO). For testing purposes, miARma provides a reduced version of raw files from this experiment in order to illustrate how it works. Briefly, this experiment contains HAEC mRNA profiles of TSA treated and control samples in triplicate generated by deep sequencing. Examples installation is described in section 3.2.

A complete example of the pipeline can be executed using:

```
perl miARma Examples/basic_examples/mRNAs/miARma_mRNAs_pipeline.ini
```

4.1. General features

4.1.1. Configuration file

In order to execute miARma-Seq, a configuration file in INI format is mandatory with information about your experiment setup. General information must be provided using the heading **[General]** at the beginning of the configuration file. Although general section is required for any analysis with miARma-Seq a configuration file only with this section will not perform any analysis. See below a detailed explanation in order to configure the different steps of the analysis. This information is mainly oriented to the path of input files and output directories.

The parameters included in this section are:

Mandatory parameters:	
type	Type of analysis to perform with miARma-Seq. Allowed values for this parameter are: miRNA, mRNA or circRNAs. Example: type=mRNA
read_dir	Folder for input files where raw data from high thorughput sequencing in fastq format are located. Example: read_dir=Examples/basic_examples/mRNAs/reads/
label	Name to identify the analysis. This name will appear in the output files and plots. Example: label= TSA
miARmaPath	Folder where miARma-Seq has been installed. Example: miARmaPath=/opt/miARma/
output_dir	Folder to store the results. Example: output_dir= Examples/basic_examples/mRNAs/results/
organism	Organism analysed in the experiment. Example: organism=human
Optional parameters:	
verbose	Parameter to show the execution data on the screen. Value of 0 for no verbose, otherwise to print "almost" everything. Example: verbose=0
threads	Number of process to run at the same time. The maximum value of this parameter is defined for user's computer. Example: threads=4
stats_file	File where stats data will be saved. Example: stats_file=stats.log
logfile	File to print the information about the execution process. Example: logfile=run_log.log
seqtype	Sequencing procedure of RNA-Seq experiment. Allowed values: Paired/Paired-End or Single/Single-End (by default). Please note that paired-end analysis samples must be named with the final end of "_1" and "_2" before file extension to correctly identify paired samples. Example: SRR488566_1.fastq and SRR488566_2.fastq. Example: seqtype=Single
strand	Parameter to specify whether the data is from a strand-specific assay. The allowed values are: yes (by default), no or reverse. Example: strand=no

4.1.2. Examples of the general information in the configuration file

1) General information of mRNA analysis.- In this example, user defines general parameters to execute miARma from its own directory, the pipeline input files are fastq files from human located in the input directory (Examples/basic_examples/mRNAs/reads/ in the example) and the results will be saved in results directory (Examples/basic_examples/mRNAs/results/ in this case) including the name of the experiment (TSA in this example). The example is composed of six single-end unstranded samples. The analysis will be performed with 4 threads and the execution information will be showed in the screen.

```
[General]
type=mRNA
```

```
verbose=1
read_dir= Examples/basic_examples/mRNAs/reads/
threads=4
label= TSA
miARmaPath=.
output_dir= Examples/basic_examples/mRNAs/results/
organism=human
seqtype=Single
strand=no
```

4.2. Quality module

The aim of the Quality module is to provide a simple way to check the quality of our sequenced samples and avoid the inclusion of outliers. This analysis will be performed with FASTQC software.

4.2.1. Input/Output files

Input: Raw data from high throughput sequencing in fastq format (compressed files are allowed).

Output:

1. HTML report with different plots and statistics of the quality of the data. These files will be saved inside a folder called Pre_fastqc_results under the path specified in output_dir. For each fastq file, an independent quality analysis process will be performed and stored in a folder with the same name of the fastq file. In order to examine the results, a html file called fastqc_report.html is included. Please visit FastQC help page to better understand the FastQC report.

2. Summary results report (xls format) with the main statistics of the analysis. This report will be generated in the output directory provided by the user. In this report, "Quality" section with the path of the quality results can be founded, together with a summary table below with the columns:

-[Filename]- Name of the sample
-[Number of reads] –Number of reads contained in the fastq files.
-[%GC Content]- Proportion of GC content of the reads
-[Read Length]- Length of the reads.
-[Encoding]- Type of encoding of the fastq files.

An example of the summary report can be consulted in the following link.

4.2.2. Configuration file

To execute this analysis, the heading **[Quality]** must be included in the configuration file. The parameters included in this analysis are:

Mandatory parameters	
prefix	Parameter to define when miARma will perform the quality analysis. Use "pre" to perform a quality analysis for unprocessed reads and "post" for processed reads (after adapter trimming step). miARma also accepts the keyword "both"

	in case you want the analysis twice: before and after the pre-processing of the reads. Since mRNA analysis do not includes adapter trimming step, only prefix=pre is recommended. Example: prefix=pre

4.2.3. Examples of configuration file to run Quality analysis

1) Quality analysis of mRNA analysis. In this example, user will perform the quality analysis executing miARma from its own directory, the pipeline input files are fastq files from human located in the input directory (Examples/basic_examples/mRNAs/reads/ in the example) and the results will be saved in results directory (Examples/basic_examples/mRNAs/results/ in this case) including the name of the experiment (TSA in this example). The example is a single-end un-stranded experiment. The analysis will perform with 4 threads and the execution data will be showed in the screen.

```
[General]
type=mRNA
verbose=1
read_dir= Examples/basic_examples/mRNAs/reads/
threads=4
label= TSA
miARmaPath=.
output_dir= Examples/basic_examples/mRNAs/results/
organism=human
seqtype=Single
strand=no

[Quality]
prefix=Pre
```

This configuration file can be founded in the examples folder of the miARma downloaded directory and can be executed using:

```
perl miARma Examples/basic_examples/mRNAs/1.Quality/1.Quality.ini
```

To inspect results for a Fastq file named SRR488566 , please check Examples/basic_examples/mRNAs/results/SRR488566_fastqc/fastqc_report.html

4.3. Aligner module

The aim of the aligner module is to align sequenced reads against the reference genome. For mRNA analysis, miARma-Seq has included topHat tool, which has been implemented with Bowtie1 and Bowtie2 algorithms. These aligners can be used as a single option or combined.

The reference genome to align the reads against is mandatory and is different for Bowtie1 and Bowtie 2. Theses reference genomes are used as pre-built Bowtie indexes, which can be downloaded here for most of organism from for Bowtie1 and here for Bowtie2. Note that, Bowtie1 and Bowtie 2 use different indexes format. Bowtie 1 index must have .ebwt extension and is only allowed for Bowtie1 analysis, while Bowtie 2 index must have .bt2 extension and is only allowed for Bowtie2 analysis. To download bowtie indexes of human genome 19 used in the examples, please see section 3.3.1. and 3.3.2.

4.3.1. Input/Output files

Input: Raw data or pre-processed data from high throughput sequencing in fastq format.

Output:

1. Aligned files in SAM/BAM format saved in the output directory provided by the user in the "bowtie1_results" or "bowtie2_results" folder.

2. Summary results report (xls format) with the main statistics of the analysis. This report will be generated in the output directory provided by the user. In this report, "Alignment" section with the path of the aligner results can be founded, together with a summary table below with the columns:

-[Filename]- Name of the sample.
-[Processed Reads]- Initial number of reads contained in the fastq file (trimmed file).
-[Aligned reads]- Number of aligned reads against the reference genome provided.
-[Multimapping reads]- Number of reads with multiple alignment.
-[Overall alignment]- Proportion of aligned read/total number of reads.
-[Fail to align]- Number of reads that fail to align.

An example of the summary report can be consulted in the following link.

4.3.2. Configuration file

To execute this analysis the heading **[Aligner]** must be included in the configuration file. The parameters included in this analysis are:

Mandatory parameters	
aligner	Specific software to perform the alignment against the corresponding index. As state above for mRNA-Seq analysis the tool topHat is implemented in miARma-Seq. Others aligners are available for miRNA analysis (Bowtie1, Bowtie2, miRDeep) and circRNAs (BWA). Please see the specific documentation of these analyses to deep in their use. Example: aligner=tophat.
tophat_aligner	Tophat uses bowtie2 by default, bowtie1 can be also specified. Specifying both, means to repeat the analysis one with each aligner (Allowed values: Bowtie1, Bowtie2 and Bowtie1-Bowtie2/Bowtie2-Bowtie1). Please note that each aligner needs the specific index for its execution. Example: tophat_aligner=Bowtie1
bowtie1index	Path of the pre-built Bowtie 1 index to the alignment of the reads. This parameter is mandatory when tophat_aligner=Bowtie1. This index can be downloaded from Bowtie 1 web page as stated above. The index is composed by various files with the same name, followed by a number and the .ebwt extension (i.e. bw1_homo_sapiens19.1.ebwt, bw1_homo_sapiens19.2.ebwt, bw1_homo_sapiens19.3.ebwt, etc). See the example below to define

	bowtie1index parameter supposing that index would be placed in Genomes/Indexes/bowtie1/human/. Example: bowtie1index=Genomes/Indexes/bowtie1/human/bw1_homo_sapiens19
bowtie2index	Path of the pre-built Bowtie 2 index to the alignment of the reads. This parameter is mandatory when tophat_aligner=Bowtie2. This index can be downloaded from Bowtie 2 web page as stated above. The index is composed by various files with same index name, followed by a number and the .bt2 extension (i.e. bw2_homo_sapiens19.1.bt2, bw2_homo_sapiens19.2.bt2, bw2_homo_sapiens19.3.bt2, etc). See the example below to define bowtie2index parameter supposing that index would be placed in Genomes/Indexes/bowtie2/human/. Example: bowtie2index=Genomes/Indexes/bowtie2/human/bw2_homo_sapiens19
gtf	File in GTF format used to extract information about gene structure (exons, introns, genes, …) Example: gtf=Examples/basic_examples/mRNAs/data/Homo_sapiens_GRCh37.74_chr.gtf
Optional parameters	
bowtiemiss	Maximum number mismatches in seed alignment in bowtie analysis. Allowed values are: 0-3 for Bowtie1 analysis (2 by default) and 0-1 for Bowtie 2 analysis (0 by default). Example: bowtiemiss=1
bowtielength	Length of the seed substrings to align during multiseed alignment. Smaller values make alignment slower but more senstive. Allowed values are comprised between 5-32. Example: bowtielength=19
tophat_multihits	Instructs TopHat to allow up to this many alignments to the reference for a given read, and choose the alignments based on their alignment scores if there are more than this number. The default is 20 for read mapping. Example: tophat_multihits=5
tophat_read_mismatches	Final read alignments having more than these many mismatches are discarded. The default value is 2. Example: read_mismatches=2
tophat_seg_mismatches	Read segments are mapped independently, allowing up to this many mismatches in each segment alignment. Default value is 2. Example: tophat_seg_mismatches=1
tophat_seg_length	Each read is cut up into segments, each at least this long. These segments are mapped independently. The default value is 25. Example: tophat_seg_length=20
tophatParameters	Other parameters to perform the alignment with topHat using the topHat recommended syntaxis in topHat user guide. Example: tophatParameters= --fusion-ignore-chromosomes chrX

4.3.3. Examples of configuration file to run Aligner module

1) Alignment with TopHat using Bowtie2: In this example, user will perform the alignment with Bowtie 2 of the input fastq files located at the input directory (Examples/basic_examples/mRNAs/reads in the example) against a pre-built Bowtie 2 index downloaded as stated above and located in index directory

(Genomes/Indexes/bowtie2/human/ in this example) and the human gtf file (located in Examples/basic_examples/mRNAs/data/ in this example). The results will be saved in results directory (Examples/basic_examples/mRNAs/results/ in this case) including the name of the experiment (TSA in this example). The example is based on single-end unstranded samples. User will execute miARma from its own directory.

```
[General]
type=mRNA
verbose=0
read_dir= Examples/basic_examples/mRNAs/reads/
threads=4
label= TSA
miARmaPath=.
output_dir= Examples/basic_examples/mRNAs/results/
organism=human
seqtype=Single
strand=no


[Aligner]
aligner=tophat
tophat_aligner=Bowtie2
bowtie2index=Genomes/Indexes/bowtie2/human/bw2_homo_sapiens19
gtf=Examples/basic_examples/mRNAs/data/Homo_sapiens_GRCh37.74_chr.gtf
```

This configuration file can be founded in the examples folder of the miARma downloaded directory and can be executed using:

```
perl miARma Examples/basic_examples/mRNAs/2.Aligner/2.1.TopHat_Read_Alignment_Bowtie2.ini
```

2) Alignment with TopHat using Bowtie1: In this example, user will perform the alignment with Bowtie 1 of the input fastq files located at the input directory (Examples/basic_examples/mRNAs/reads in the example) against a pre-built Bowtie 1 index downloaded as stated above and located in index directory (Genomes/Indexes/bowtie1/human/ in this example) and the human gtf file (located in Examples/basic_examples/mRNAs/data/ in this example). The results will be saved in results directory (Examples/basic_examples/mRNAs/results/ in this case) including the name of the experiment (TSA in this example). The example is a single-end with no stranded samples. User will execute miARma from its own directory. The alignment will be performed allowing 0 mismatches in seed alignment, with 19 of length of seed substrings, choosing the alignments with scores >5, discarding alignments with >2 mismatches, allowing 1 mismatch in each segment alignment and cutting up each read into segments of at least 20 nucleotides.

```
[General]
type=mRNA
verbose=0
read_dir= Examples/basic_examples/mRNAs/reads/
threads=4
label= TSA
miARmaPath=.
output_dir= Examples/basic_examples/mRNAs/results/
organism=human
seqtype=Single
strand=no


[Aligner]
aligner=tophat
```

```
tophat_aligner=Bowtie1
bowtie1index=Genomes/Indexes/bowtie1/human/bw1_homo_sapiens19
gtf=Examples/basic_examples/mRNAs/data/Homo_sapiens_GRCh37.74_chr.gtf
tophat_multihits=5
tophat_read_mismatches=2
tophat_seg_mismatches=1
tophat_seg_length=20
bowtiemiss=0
bowtieleng=19
```

This configuration file can be founded in the examples folder of the miARma downloaded directory and can be executed using:

```
perl miARma Examples/basic_examples/mRNAs/2.Aligner/2.2.TopHat_Read_Alignment_Bowtie1.ini
```

3) Alignment with TopHat using Bowtie1 and Bowtie2: In this example, user will perform the alignment with Bowtie 1 and Bowtie 2 of the input fastq files located at the input directory (Examples/basic_examples/mRNAs/reads in the example) against the specific pre-built Bowtie indexes for Bowtie 1 and Bowtie 2downloaded as stated above and located in index directory (Genomes/Indexes/bowtie1/human/ and Genomes/Indexes/bowtie2/human/ in this example) and the human gtf file (located in Examples/basic_examples/mRNAs/data/ in this example). The results will be saved in results directory (Examples/basic_examples/mRNAs/results/ in this case) including the name of the experiment (TSA in this example). The example is based on single-end unstranded samples. User will execute miARma from its own directory.

```
[General]
type=mRNA
verbose=0
read_dir= Examples/basic_examples/mRNAs/reads/
threads=4
label= TSA
miARmaPath=.
output_dir= Examples/basic_examples/mRNAs/results/
organism=human
seqtype=Single
strand=no


[Aligner]
aligner=tophat
tophat_aligner=Bowtie1-Bowtie2
bowtie1index=Genomes/Indexes/bowtie1/human/bw1_homo_sapiens19
bowtie2index=Genomes/Indexes/bowtie2/human/bw2_homo_sapiens19
gtf=Examples/basic_examples/mRNAs/data/Homo_sapiens_GRCh37.74_chr.gtf
```

This configuration file can be founded in the examples folder of the miARma downloaded directory and can be executed using:

```
perl miARma Examples/basic_examples/mRNAs/2.Aligner/2.3.TopHat_Read_Alignment_Bowtie2_Bowtie1.ini
```

4.4. ReadCount module

The aim of the ReadCount module is the summarization of mapped reads into genomic features such as genes, exons, promoter, gene bodies, genomic bins and chromosomal locations. For mRNA analysis, miARma-Seq has implemented featureCounts .

4.4.1. Input/Output files

Input: Aligned files in SAM/BAM format.

Output:

1. Tabulated text file with the entities and the correspondent counts in the output directory provided by the user within "Readcount_results" folder. In this file, each row corresponds to an mRNA or gene identifier and each column to the number of reads of that selected feature in each sample. The names of the columns are the name of each sample. Example:

	SRR488566	SRR488567	SRR488568	SRR488569	SRR488570	SRR488571
ENST00000016171	387	380	448	275	226	331
ENST00000212015	5	6	13	1	3	18
ENST00000224756	855	398	1347	205	302	453
ENST00000224950	3	5	15	4	1	2
ENST00000225171	15	15	13	3	5	3
ENST00000225174	96	136	126	366	323	533

2. Summary results report (xls format) with the main statistics of the analysis. This report will be generated in the output directory provided by the user. In this report, "ReadCount" section with the path of the readcount results can be found, together with a summary table below with the columns:

-[Filename]- Name of the sample.
-[Processed Reads]- Initial number of reads contained in the fastq file (trimmed file).
-[Assigned reads]- Number of assigned reads using the database in gtf format provided.
-[Strand]- Type of experiment.
-[Number of identified entities]- Number of identified entities.

An example of the summary report can be consulted in the following link.

4.4.2. Configuration file

To execute this analysis the heading **[Readcount]** must be included in the configuration file. The parameters included in this analysis are:

Mandatory parameters	
database	File in GTF format used to calculate the number of reads. Example: database=Examples/basic_examples/mRNAs/data/Homo_sapiens_GRCh37.74_chr.gtf
Optional parameters	
featuretype	Feature type (3rd column in GTF file) to be used, all features of other type are ignored (default:exon) for featureCounts analysis. Example: featuretype=exon
seqid	GTF attribute to be used as feature ID. Allowed values are presented in the

	GFT file, for instance: gene_id (for EnsEMBL gene identifiers), gene_name (for gene symbols) and transcript_id (for EnsEMBL transcripts ids). Example: seqid=transcript_id
quality	Quality value threshold to avoid counting low quality reads. Example: quality=10
parameters	Other featureCounts parameters to perform the analysis using the featureCounts recommended syntaxis. Example: parameters= -d 50 -D 600

4.4.3. Examples of configuration file to run ReadCount module

1) Quantification of mRNAs by Readcount: In this example, user will perform the read summarization corresponding to mRNAs taking as a reference the GTF from human genome (to download the gtf file used in this example see section 3.2.). User will execute miARma from its own directory. The input files are aligned sam files from example 2.1. located in the input directory (Examples/basic_examples/mRNAs/results/ in the example) and the results will be saved in results directory (Examples/basic_examples/mRNAs/results/ in this case). The analysis will be performed with a minimum quality value of 10.

```
[General]
type=mRNA
verbose=0
read_dir= Examples/basic_examples/mRNAs/reads/
threads=4
label= TSA
miARmaPath=.
output_dir= Examples/basic_examples/mRNAs/results/
organism=human
seqtype=Single
strand=no

[ReadCount]
database= Examples/basic_examples/mRNAs/data/Homo_sapiens_GRCh37.74_chr.gtf
seqid=transcript_id
quality=10
featuretype=exon
```

This configuration file can be founded in the examples folder of the miARma downloaded directory and can be executed using:

```
perl miARma Examples/basic_examples/mRNAs/3.ReadCount/3.1.ReadCount.ini
```

4.5. Differential Expression module

The aim of this module is to perform the differential expression analysis between different experimental conditions. For this purpose, miARma-Seq implements NOISeq and EdgeR software. Both are valuable tools to identify differentially expressed (DE) elements, which covers different requirements. edgeR is a widely employed tool for differential expression analysis that allows not only the identification of DE elements between two experimental conditions but more complicated comparisons in the same analysis process. On other hand, Noiseq allows the simulation of technical replicates to increase the reliability of the results, when no replicates are available for the analysis.

4.5.1. Input/Output files

Input: Tabulated file with the counts of the reads. In this file, each row corresponds to a feature and each column to the number of reads of that feature. The names of the columns are the name of each sample. Example:

	SRR488566	SRR488567	SRR488568	SRR488569	SRR488570	SRR488571
ENST00000016171	387	380	448	275	226	331
ENST00000212015	5	6	13	1	3	18
ENST00000224756	855	398	1347	205	302	453
ENST00000224950	3	5	15	4	1	2
ENST00000225171	15	15	13	3	5	3
ENST00000225174	96	136	126	366	323	533

Output:

1. <u>Tabulated results files</u> (excel compatible) with the entities differentially expressed (DE) and the statistical values of the analysis for any of the comparison between the different experimental conditions. According to the tool selected for the analysis, the format of the results differs. Specific format will be detailed explained below.

- **EdgeR results**- EdgeR results will be located in the "EdgeR_results" directory in the output_dir directory defined by the user. The results with the DE entities of each condition will be saved in different files. The name of the results files will be constructed as follows:

(Label_defined_by_user)_(Aligner_tool)_EdgeR_results_(label_of_the_comparison).xls

Example: For the comparison of TSA experiment performed with tophat using Bowtie2, the resultant file will be: TSA_top_bw2_EdgeR_results_Comp.xls

EdgeR result file contains 5 columns:

- [Entity]- Name of the DE entity, which according to the experiment could be the name of miRNAs, mRNAs or circRNAs.
- [logFC]- Log2-fold- change value.
- [logCPM]- Log2 counts-per-million.
- [Pvalue]- Probability value.
- [FDR]- False discovery rate obtained by Benjamini and Hochberg's algorithm.

Example:

	logFC	logCPM	PValue	FDR
ENST00000369849	7.619099461	11.85573339	1.06E-89	2.12E-86
ENST00000430998	-5.870096638	10.75890654	2.22E-66	2.22E-63
ENST00000369583	3.769556218	13.92140279	1.90E-43	1.27E-40
ENST00000370602	6.627294947	9.080539331	3.72E-38	1.86E-35
ENST00000369209	5.863248361	8.655767131	3.63E-37	1.45E-34
ENST00000302424	-3.541871217	12.30165857	8.71E-36	2.90E-33

- **Noiseq results**- Noiseq results will be located in the "Noiseq_results" directory in the output_dir directory defined by the user. Noiseq generates a results file with the statistical values of every expressed entity for each condition. The name of this file will be constructed as follows:

(Label_defined_by_user)_(Aligner_tool)_Noiseq_results_(label_of_the_comparison).xls

Example: For the comparison of TSA experiment performed with tophat using Bowtie2, the resultant file will be: TSA_top_bw2_Noiseq_results_Comp.xls

Both files contain 7 columns:

- [Entity]- Name of the DE entity, which according to the experiment could be the name of miRNAs, mRNAs or circRNAs.
- [Condition1_mean]- Expression values for condition 1.
- [Condition2_mean]- Expression values for condition 2.
- [M] - log2-ratio of the two conditions.
- [D] - value of the difference between conditions.
- [prob] - probability of differential expression.
- [ranking] – summary statistic of "M" and "D" values.

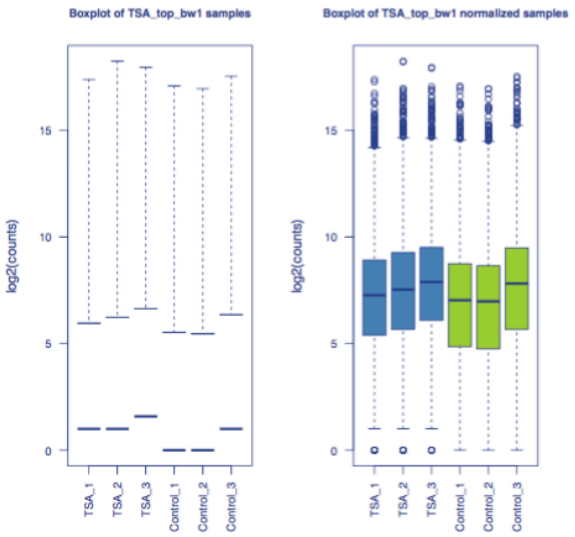
Example:

	Treated_mean	Untreated_mean	M	D	prob	ranking
ENST00000369583	2177.517325	27307.59397	-3.64854	25130.076	0.9971613	-25130.07691
ENST00000369849	38.20205833	7007.774842	-7.51916	6969.572	0.9971613	-6969.57684
ENST00000302424	9947.815988	759.4912734	3.71127	9188.324	0.9962489	9188.325464
ENST00000430998	3506.948954	54.92580282	5.99658	3452.023	0.9928021	3452.02836
ENST00000373115	10141.37308	1462.162751	2.79407	8679.210	0.9804339	8679.210783

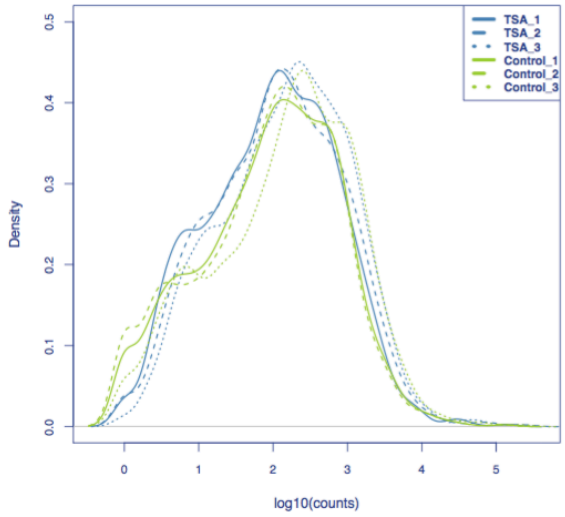
2. Exploratory plots of the analysis. miARma-Seq provides a exhaustive PDF report with different plots in order to make easier to the user the interpretation of the data. This report contains:

2.1. Analysis of the distribution of the reads in the samples. The detailed inspection of the distribution of the reads in the different samples allows to the user identify samples with abnormal distribution of the reads. These samples are recommended to be removed from the analysis since may introduce noise or affect to the final results. In order to inspect the distribution of the reads in the different samples, miARma-Seq generates two kinds of plots:

- Boxplot of the distribution of the counts. The first page of the report contains 2 boxplots with the distribution of the counts, before (left) and after (right) the normalization process. The log2(number of counts) is represented for each sample. Boxplot of non-normalized data usually will have a lower limit near to – infinite due to the mRNAs with no counts. The different replicates will be represented with the same colour. The expected boxplot will look like this:

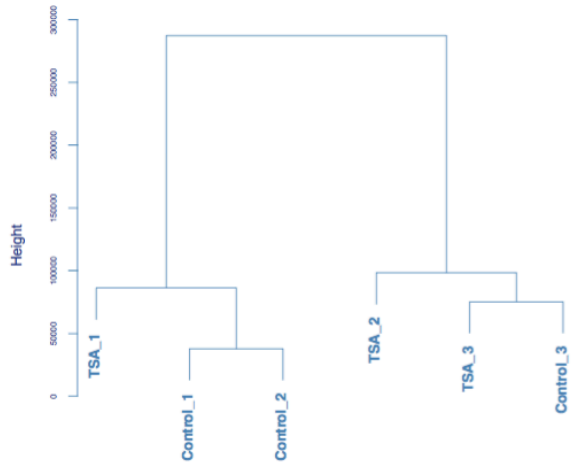


- Density plot of the distribution of the counts. The second and third page of the report contains 2 density plots with the distribution of the counts, before (second page) and after (third) the normalization process. The plot represents the density of the log10 of the counts for each sample. The different replicates will be represented with the same colour. These plots will look like this:

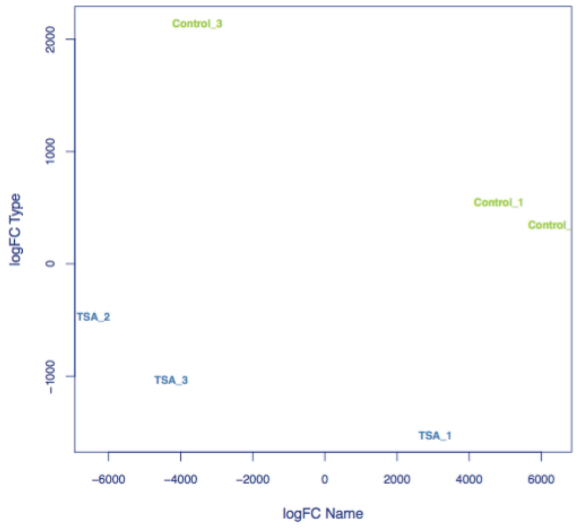


2.2. Analysis of the samples similarity- In order to examine the quality of the data obtained in the experiment, miARma-Seq has implemented different plots, which allows the inspection of the diversity between the samples. For a good quality experiment, the samples belonging to the same experimental conditions should present more similarity between them than with the samples of others experimental conditions. Thus, with these analyses user can identify samples with low quality to remove from the analysis.

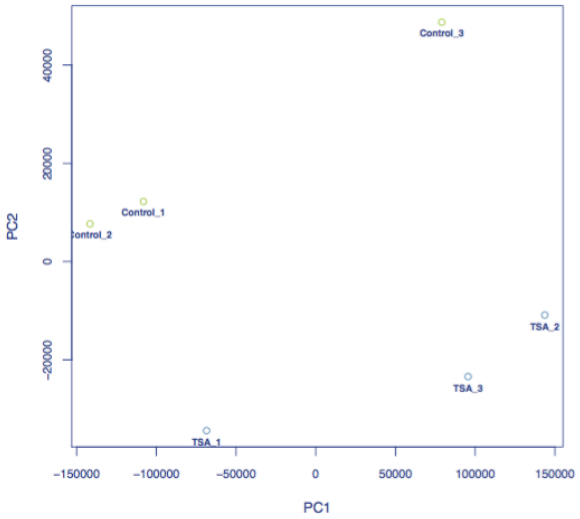
-Hierarchical clustering of the samples: The hierarchical clustering plots, before and after normalization process, classify the samples according to their similarity. The distance of the branch is proportional to the sample distance.



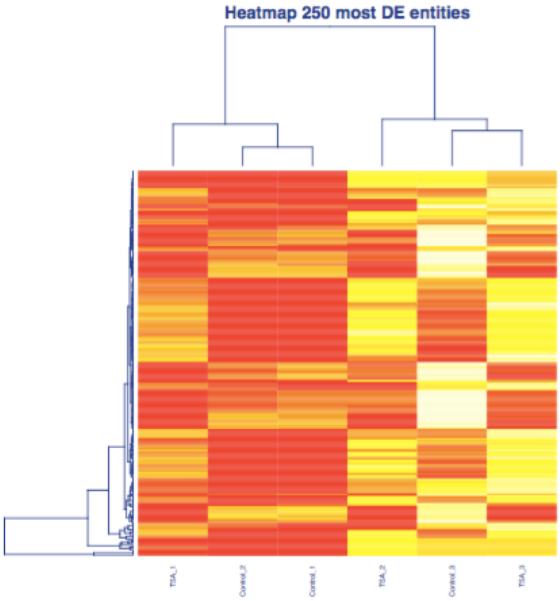
-Multidimensional plot (MDS): The MDS plot divides the samples in a two-dimensional plot according to their similarity. Each sample is represented with its name and coloured according to the experimental condition that belongs to.



-Principal Component Analysis (PCA) plot: The PCA plot divides the samples in a two-dimensional plot according to their similarity. Each sample is represented with its name and coloured according to the experimental condition that belongs to.



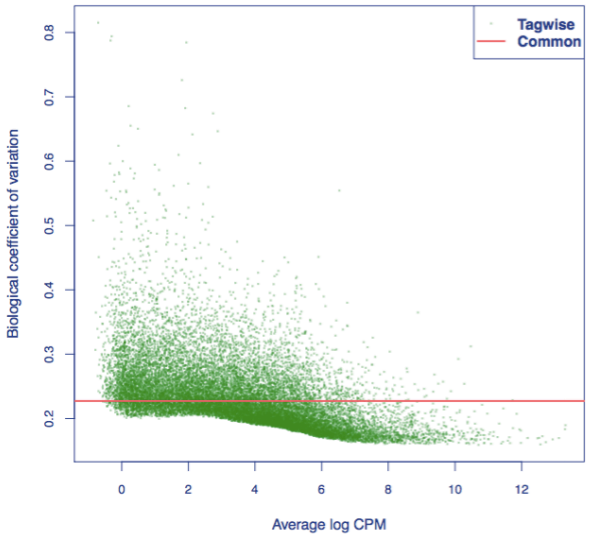
-Heatmap: The heatmap allows to the user evaluate the similarity between the samples according to the 250 most expressed mRNAs expression.



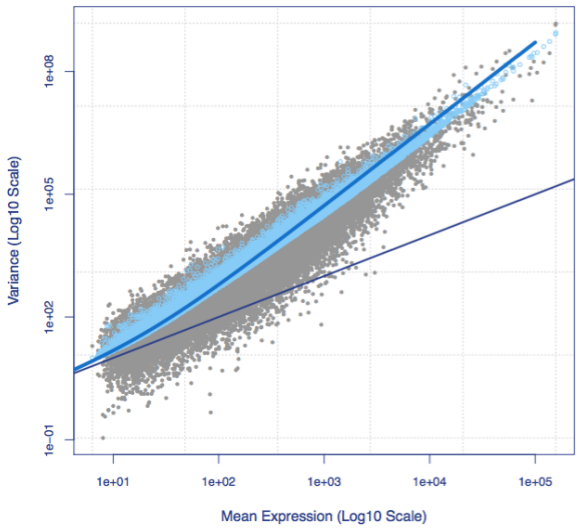
3. Results plots of the analysis.- miARma-Seq generates a PDF report with plots to explore the results with both tools, EdgeR and Noiseq.

3.1. Results plots with EdgeR:

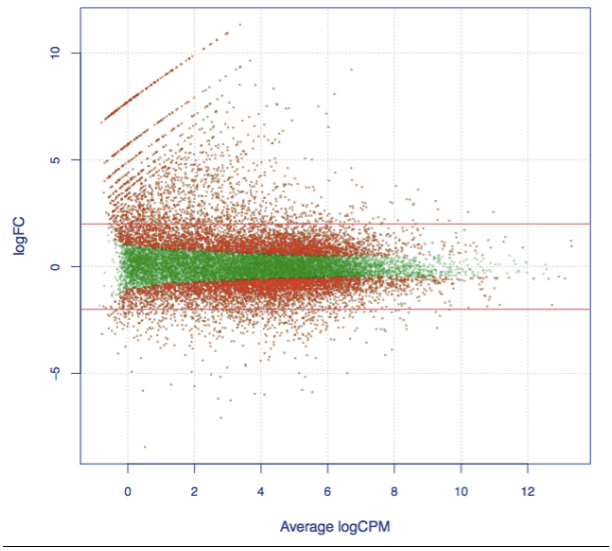
-Biological Variation Plot: The square root of dispersion is the coefficient of biological variation (BCV). This plot illustrates the relationship of biological coefficient of variation versus mean log CPM.



-Mean Variance Plot: This plot can be used to explore the mean-variance relationship; each dot represents the estimated mean and variance for each gene, with binned variances as well as the trended common dispersion overlaid.

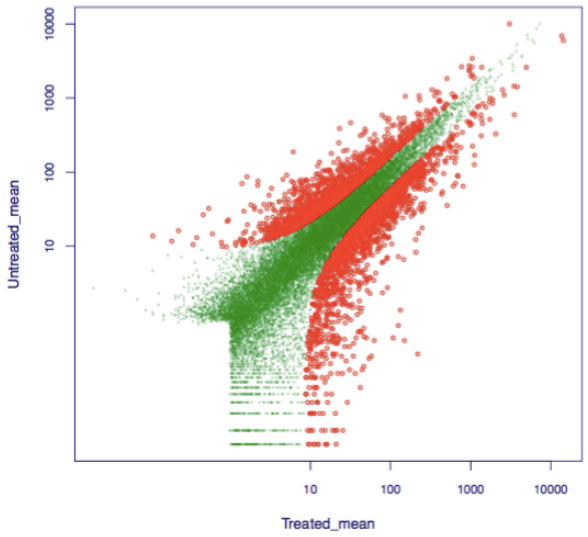


-Expression Plot: miARma-Seq generates one expression plot for each comparison. This plot shows all the logFCs against average count size, highlighting the DE genes.

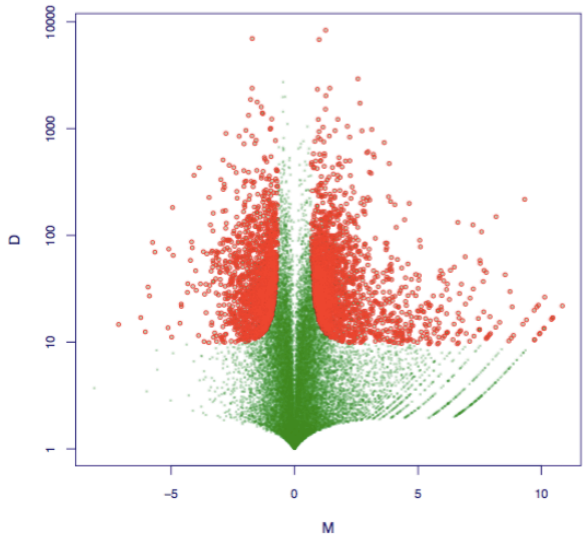


3.2. Results plots with Noiseq: For each comparison a PDF report with the results plot is generated

- Expression Plot: Summary plot of the expression values for both conditions (green), where differentially expressed genes are highlighted (red)



- MD Plot: Summary plot for (M,D) values (green) and the differentially expressed genes (red).



4. Summary results report (xls format) with the main statistics of the analysis. This report will be generated in the output directory provided by the user. In this report, "Differential Expression Analysis" section with the path of the Differential Expression Analysis results can be founded for each tool EdgeR and Noiseq. Each tool shows different information.

For EdgeR analysis the summary table shows the columns:

-[Comparison]- Name of the comparison defined in the contrastfile.
-[File]- Name of the file with the DE elements.
-[Number of DE elements (Pval<=0.05)]- Number of DE elements with a p-value <=0.05.
-[Number of DE elements (FDR <=0.05)]- Number of DE elements with a FDR <=0.05.

For Noiseq analysis the summary table shows the columns:

-[Comparison]- Name of the comparison defined in the contrastfile.
-[File]- Name of the file with the DE elements.
-[Number of DE elements (Prob>=0.8)]-Number of DE elements with a probability >=0.8.

An example of the summary report can be consulted in the following link.

4.5.2. Configuration file

To execute this analysis the heading **[DEAnalysis]** must be included in the configuration file. The parameters included in this analysis are:

Mandatory parameters	
desoft	Specific software to perform the differential expression analysis. As state above the tools EdgeR and Noiseq are implemented in miARma-Seq. These tools can be selected alone or in combination. Thus allowed values for this parameter are: edger, noiseq or edger-noiseq. Note that, each specific tool requires specific parameters. See examples below to deep in the analysis possibilities. Example: desoft=EdgeR-Noiseq

targetfile	Complete path of the target file. This is a tabulated file that contains the experimental condition of each sample. The first column of this file must coincide with the column names of the input files. Note that, only those samples present in the target file will be used for the analysis. The second column must contain the names of the samples to be used to the plots, and the next columns the condition of each factor. For example, for the input previously showed the correspondent target file will contain the next information: Filename Name Type SRR488566 TSA_1 Treated SRR488567 TSA_2 Treated SRR488568 TSA_3 Treated SRR488569 Control_1 Untreated SRR488570 Control_2 Untreated SRR488571 Control_3 Untreated In this example, the first column "Filename" contain the name of the samples obtained from SRA, the second column "Name" contain the names to use in the exploratory plots and the third column "Type" corresponds to the experimental condition, which in this case is the treatment or not with TSA. This target file can be downloaded as stated in section 3.2. Example: targetfile=Examples/basic_examples/mRNAs/data/targets.txt
contrastfile	Path of the contrast file o perform the DE analysis with EdgeR. This file has one column with the contrasts user want to evaluate. The syntax of the contrast should be: name_of_contrast=contrast to evaluate. Any type of contrast can be done but condition name must be one of the conditions present in targets file. In addition, contrast must differ of 0 (ie: cond=WT-WT is not allowed). There is no limit in the number of contrasts. For example, for the input previously showed the correspondent contrast file will contain the next information: Name Comp=Untreated-Treated In this example, there is only one contrast condition: Untreated-Treated. This contrast file can be downloaded as stated in section 3.2.1. Example: contrastfile=Examples/basic_examples/mRNAs/data/contrast.txt
filter	This value refers to filter processing in the reads. Filter process is usually recommended to remove the noise and less informative reads, such as low expressed elements with very low read counts. This low read counts might not reveal a real biological information, being due to sequencing errors or inaccuracy during the procedure of read alignment to the reference genome, such as cross mapping artefacts. For this reason, a minimum read count value could be used to filter out reads detected below the cutoff. EdgeR and Noiseq offers different options to filter the reads. While EdgeR is implemented with a filter processing using a value ofcounts per million as a cutoff, Noiseq offers 3 different methods of filtering. See in the specific parameters below for more information. Thus, allowed values for this parameter are: yes or no. Example: filter=yes
Optional parameters	
cpmvalue	Cutoff for the counts per million value to be used in filter processing

	with methods 1 and 3 with Noiseq software (see below for more information about these methods) and in filter processing with EdgeR (1 cpm by default). Example: cpmvalue=2
Specific parameters for EdgeR:	
edger_normethod	Normalization method to perform the DE analysis with EdgeR. Normalization allows the comparison between the samples by means of the elimination of the effects of artefacts, noise or outlier values. If data comes from different experiments or datasets normalization process is highly recommended to minimize the effect of the different experimental conditions. EdgeR allows the normalization with 3 methods: "TMM" (default), "RLE", "upperquartile" or "none" (no normalization). Example: edger_normethod=TMM
repthreshold	Number of replicates that have to contains at least a defined number of reads per million to perform the filtering process with EdgeR software (2 replicates by default) Example: repthreshold=3
replicates	Value to indicate if replicates samples are present in the analysis to perform the DE analysis with EdgeR. It is highly recommended to perform the analysis with replicates, but if there are not available a biological coefficient variation (bcv) value (see below for more information about this parameter) can be used to perform the differential expression analysis. The allowed values for this parameter are: "yes" (by default) or "no". Example: replicates=no
bcvvalue	Value for the common biological coefficient variation (bcv) (square-root-dispersion) in experiments without replicates to perform the DE analysis with EdgeR. Standard values from well-controlled experiments are 0.4 for human data (by default), 0.1 for data on genetically identical model organisms or 0.01 for technical replicates. Example: bcvvalue=0.3
Specific parameters for Noiseq	
qvalue	Probability of differential expression to perform the DE analysis with Noiseq. The elements with a probability greater than the defined q-value will be highlighted in the results plots. Please remember that, when using NOISeq, the probability of differential expression is not equivalent to 1 − pvalue. Noiseq team recommends for q to use values around 0.8. If no replicates are available, then it is preferable to use a higher threshold such as q = 0.9. See Noiseq user's manual for more information. By default qvalue is 0.8 Example: qvalue=0.9
filtermethod	Method that will be used to filtering process with Noiseq software. See filter parameter above for general recommendations. Noiseq allows filtering with 3 methods: CPM method (1) (by default), Wilcoxon method (2) and Proportion test (3). See Noiseq user's manual for more information. Thus allowed values are: 1, 2 or 3, to refer the previously stated filtering methods. Example: filtermethod=2
cutoffvalue	Cutoff for the coefficient of variation per condition to be used in filter processing with CPM method (1) in Noiseq analysis. This cutoff is expressed in percentage (100 by default). See Noiseq user's manual for more information.

	Example: cutoffvalue=80
noiseq_normethod	Normalization method to perform the DE analysis with Noiseq. Normalization allows the comparison between the samples by means of the elimination of the effects of artefacts, noise or outlier values. If data comes from different experiments or datasets normalization process is highly recommended to minimize the effect of the different experimental conditions Noiseq allows the following normalization methods: "rpkm" (default), "uqua" (upper quartile), "tmm" (trimmed mean of M) or "n" (no normalization). See Noiseq user's manual for more information. Example: noiseq_normethod=tmm
replicatevalue	Type of replicates to be used to perform the DE analysis with Noiseq. Allowed values are: Technical, biological or no. Inclusion of technical or biological replicates is highly recommended. Technical replicates involve taking one sample from the same source tube, and analysing it across multiple conditions, while biological replicates are different samples measured across multiple conditions. See Noiseq user's manual for more information. By default, technical replicates option is chosen. Example: replicatevalue=biological
kvalue	Counts equal to 0 are replaced by k value to perform the DE analysis with Noiseq. See Noiseq user's manual for more information. By default, kvalue = 0.5. Example: kvalue = 1
lcvalue	Additional length correction in the normalization process. This correction is done by dividing expression by length^lc to perform the DE analysis with Noiseq. See Noiseq user's manual for more information. By default, lcvalue = 0 for no length correction is applied. Example: lcvalue = 0.5.
pnrvalue	Percentage of the total reads used to simulate each sample when no replicates are available to perform the DE analysis with Noiseq. See Noiseq user's manual for more information. By default, pnrvalue = 0.2. Example: pnrvalue = 0.5.
nssvalue	Number of samples to simulate for each condition (nss>= 2) to perform the DE analysis with Noiseq. See Noiseq user's manual for more information. By default, nssvalue = 5. Example: nssvalue = 3.
vvalue	Variability in the simulated sample total reads to perform the DE analysis with Noiseq. Sample total reads is computed as a random value from a uniform distribution in the interval [(pnr-v)*sum(counts), (pnr+v)*sum(counts)]. See Noiseq user's manual for more information. By default, vvalue = 0.02. Example: vvalue = 0.05.

4.5.3. Examples of configuration file to run DEAnalysis module

1) Differential expression analysis of mRNAs by EdgeR: In this example, user will perform the differential expression analysis of the number of reads associated to a gene. User will execute miARma from its own directory, the input files are tabulated files with the counts from example 3.1. located in the input directory (Examples/basic_examples/mRNAs/results/ in the example) and the results will be saved in output directory (Examples/basic_examples/mRNAs/results/ in this case). The differential expression

analysis will be performed by EdgeR, using the experimental conditions defined in the target file and the comparisons defined in the contrast file. Filtering process will be carry out discarding those counts less than 2 counts per million in at least 1 replicate. Normalization process will be performed using TMM method.

```
[General]
type=mRNA
verbose=0
read_dir= Examples/basic_examples/mRNAs/reads/
threads=4
label= TSA
miARmaPath=.
output_dir= Examples/basic_examples/mRNAs/results/
organism=human
seqtype=Single
strand=no


[DEAnalysis]
desoft=EdgeR
targetfile=Examples/basic_examples/mRNAs/data/targets.txt
contrastfile=Examples/basic_examples/mRNAs/data/contrast.txt
filter=yes
cpmvalue=2
repthreshold=1
edger_normethod=TMM
replicates=yes
```

This configuration file can be founded in the examples folder of the miARma downloaded directory and can be executed using:

```
perl miARma Examples/basic_examples/mRNAs/4.DEAnalysis/4.1.DEAnalysis_EdgeR.ini
```

2) Differential expression analysis of mRNAs by Noiseq: In this example, user will perform the differential expression analysis of the number of reads associated to a gene variant. User will execute miARma from its own directory, the input files are tabulated files with the counts from example 3.1. located in the input directory (Examples/basic_examples/mRNAs/results/ in the example) and the results will be saved in the output directory (Examples/basic_examples/mRNAs/results/ in this case). The differential expression analysis will be performed by Noiseq tool, using the experimental conditions defined in the target file and the comparisons defined in the contrast file. Filtering process will be carry out with CPM method using as cutoff 2 counts per million and a coefficient of variation per condition of 80%. Normalization process will be performed using tmm method. In addition, counts equal to 0 are replaced by 1 in the normalization process and a length correction will also performed be performed using 0.5 value. In this analysis the q value cutoff has been established in 0.9 to select the differentially expressed elements with Noiseq.

```
[General]
type=mRNA
verbose=0
read_dir= Examples/basic_examples/mRNAs/reads/
threads=4
label= TSA
miARmaPath=.
output_dir= Examples/basic_examples/mRNAs/results/
organism=human
```

```
seqtype=Single
strand=no

[DEAnalysis]
desoft=Noiseq
targetfile=Examples/basic_examples/mRNAs/data/targets.txt
contrastfile=Examples/basic_examples/mRNAs/data/contrast.txt
qvalue=0.9
filter=yes
filtermethod=1
cpmvalue=2
cutoffvalue=80
noiseq_normethod=tmm
replicates=yes
replicatevalue=biological
kvalue = 1
lcvalue = 0.5
```

This configuration file can be founded in the examples folder of the miARma downloaded directory and can be executed using:

```
perl miARma Examples/basic_examples/mRNAs/4.DEAnalysis/4.2.DEAnalysis_Noiseq.ini
```

3) Differential expression analysis of mRNAs by EdgeR and Noiseq: In this example, user will perform the differential expression analysis of the number of reads associated to a gene. User will execute miARma from its own directory, the input files are tabulated files with the counts from example 4.1. located in the input directory (Examples/basic_examples/mRNAs/results/ in the example) and the results will be saved in results directory (Examples/basic_examples/mRNAs/results/ in this case). The differential expression analysis will be performed with both, EdgeR and Noiseq tools, using the experimental conditions defined in the target file and the comparisons defined in the contrast file. Filtering process will be carry out with default filter methods (those counts less than 1 counts per million will be discarded), and normalization process will be performed as default option using tmm method for analysis with EdgeR and rpkm for Noiseq. In this analysis the default cutoff of q value (0.8) will be used to select the differentially expressed elements with Noiseq.

```
[General]
type=mRNA
verbose=0
read_dir= Examples/basic_examples/mRNAs/reads/
threads=4
label= TSA
miARmaPath=.
output_dir= Examples/basic_examples/mRNAs/results/
organism=human
seqtype=Single
strand=no

[DEAnalysis]
desoft=EdgeR-Noiseq
targetfile=Examples/basic_examples/mRNAs/data/targets.txt
contrastfile=Examples/basic_examples/mRNAs/data/contrast.txt
filter=yes
replicates=yes
```

This configuration file can be founded in the examples folder of the miARma downloaded directory and can be executed using:

```
perl miARma Examples/basic_examples/mRNAs/4.DEAnalysis/4.3.DEAnalysis_EdgeR_Noiseq.ini
```

4.6. Functional Analysis module

The aim of the Functional Analysis module is to perform an enrichment analysis in order to identify gene ontologies and metabolic pathways enriched by the DE genes identified. For this analysis, miARma-Seq has implemented GoSeq using GO ontologies and KEGG pathways.

4.6.1. Input/Output files

Input: Output files from differential expression analysis with Noiseq and/or EdgeR. Please see section 4.5.1 to obtain more information about the format of these files.

Output:

1. Two tabulated text file (xls format) with the enriched functional categories for both, up-regulated and down-regulated DE genes in the Functional_Analysis_results directory. Each file contains 7 columns:

- [category]- Code for a GO category or KEGG pathways.
- [over_represented_pvalue]- P-value of over-represented terms.
- [under_represented_pvalue]- P-value of under-represented terms.
- [numDEinCat]- Number of DE genes identified in the category.
- [numInCat]- Total number of genes identified in the category.
- [term]- Term of GO category or KEGG pathway.
- [ontology]- Type of ontology. It could be: CC (cellular component), MF (molecular function), BP (biological process) and KEGG (pathway).

Example:

category	over_represented_pvalue	under_represented_pvalue	numDEInCat	numInCat	term	ontology
GO:0032991	0.000241498	0.999922311	139	151	macromolecular complex	CC
GO:0043234	0.000774474	0.999748454	117	127	protein complex	CC
GO:0044822	0.010084294	0.998137681	43	45	poly(A) RNA binding	MF
GO:0044422	0.012150324	0.993359232	244	281	organelle part	CC
GO:0044446	0.019395511	0.989022467	234	270	intracellular organelle part	CC

2. Summary results report (xls format) with the main statistics of the analysis. This report will be generated in the output directory provided by the user. In this report, "Functional Analysis by GoSeq" section with the path of the target prediction results can be founded, with a summary table below with the columns:

-[File]- Name of the Functional Analysis results file.

-[Number of Over Represented Terms (Pval<0.05)]- Number of over-represented terms (pval<0.05).

-[Number of Under Represented Terms (Pval<0.05)]- Number of under-represented terms (pval<0.05).

- [Ontology]- Type of ontology.

An example of the summary report can be consulted in the following link.

4.6.2. Configuration file

To execute this analysis the heading **[FAnalysis]** must be included in the configuration file. The parameters included in this analysis are:

Mandatory parameters	
seqid	GTF attribute to be used as feature ID. Allowed values are presented in the GFT file, for instance: gene_id (for EnsEMBL gene identifiers), gene_name (for gene symbols) and transcript_id (for EnsEMBL transcripts ids). Example: seqid=transcript_id
Optional parameters	
noiseq_cutoff	Cutoff value to select statistically significant genes identified with Noiseq tool. The selected genes will be included in the functional analysis. By default, this value has been established in 0.8. Example: noiseq_cutoff=0.9.
edger_cutoff	Cutoff value to select statistically significant genes identified with EdgeR tool. The selected genes will be included in the functional analysis. By default, this value has been established in 0.05. Example: edger_cutoff=0.01

4.6.3. Examples of configuration file to run ReadCount module

1) Functional analysis of DE genes identified: In this example, user will perform the functional analysis of the DE genes identified from example 4.3 located in the input directory (Examples/basic_examples/mRNAs/results/ in the example). User will execute miARma from its own directory and the results will be saved in results directory (Examples/basic_examples/mRNAs/results/ in this case). In the functional analysis only those DE genes identified with EdgeR with a p-value<0.05 and those identified with Noiseq with a probability>0.8 will be included.

```
[General]
type=mRNA
verbose=0
read_dir= Examples/basic_examples/mRNAs/reads/
threads=4
label= TSA
miARmaPath=.
output_dir= Examples/basic_examples/mRNAs/results/
organism=human
seqtype=Single
strand=no

[FAnalysis]
seqid=transcript_id
noiseq_cutoff=0.8
edger_cutoff=0.05
```

This configuration file can be founded in the examples folder of the miARma downloaded directory and can be executed using:

```
perl miARma Examples/basic_examples/mRNAs/5.FAnalysis/5.1.Functional_Analysis.ini
```

4.7. Target prediction module

The aim of this module is to perform the target prediction analysis. To achieve this goal, miARma-Seq has implemented miRGate tool. miRGate is a new database containing novel computational predicted miRNA-mRNA pairs that are calculated using well-established algorithms such as miRanda, Pita, TargetScan, RNAhybrid or MicroTar. In addition, miRGate includes experimental validated miRNA-mRNAs pairs providing to miARma-Seq a high reliability tool to miRNA-mRNA target prediction. Thus, miRGate provides the target genes of the input DE miRNA data, the target miRNAs of the input DE mRNA data or even the DE mRNAs targeted by DE miRNAs from the negative correlations between DE miRNAs and DE mRNAs.

4.7.1. Input/Output files

Input: Tabulated file (xls format) with the DE expressed mRNAs from the Noiseq or EdgeR analysis with the DEAnalysis module. To obtain more information about the format of these files please consult the output format of the Section 4.5.2. Differential Expression module.

Output:

1. Tabulated file (excel compatible) with the predicted targets and the statistical values of the prediction. The standard output file will contain 12 columns:

- [miRNA]- Name of the miRNA.
- [miRNA FC]- Fold Change of the miRNA obtained in the DE analysis.
- [miRNA FDR]- FDR value of the miRNA obtained in the DE analysis.
- [Ensembl Gene]- Ensembl code of the targeted gene.
- [Gene Symbol]- Symbol of the targeted gene.
- [Ensembl Transcript]- Ensembl code of the targeted transcript.
- [Gene FC]- Fold Change of the mRNA obtained in the DE analysis.
- [Gene FDR]- FDR value of the mRNA obtained in the DE analysis.
- [Method]- Method of prediction.
- [Target Site]- Type of target site.
- [Score]- Z-score of the prediction.
- [Energy]- Energy of the prediction.

2. Summary results report (xls format) with the main statistics of the analysis. This report will be generated in the output directory provided by the user. In this report, "miRNA-mRNA Target Predictions by miRGate" section with the path of the target prediction results can be founded, together with 2 summary tables:

miRNAs with more associations:

-[File]- Name of the DE Analysis results file.
-[miRNA]- Name of the miRNA with more associations (5 for each file).
-[Number of associations]- Number of associations.

Genes more regulated:
-[File]- Name of the DE Analysis results file.
-[GeneName]- Name of the gene regulated for more miRNAs (5 for each file).
-[Number of associations]- Number of associations.

An example of the summary report can be consulted in the following link.

4.7.2. Configuration file

To execute this analysis the heading **[TargetPrediction]** must be included in the configuration file. The parameters included in this analysis are:

Optional parameters	
noiseq_cutoff	Cutoff value to select statistically significant results performed with Noiseq tool. The selected entities will be included in the target prediction analysis. By default, this value has been established in 0.8. Example: noiseq_cutoff=0.9.
edger_cutoff	Cutoff value to select statistically significant results performed with EdgeR tool. The selected entities will be included in the target prediction analysis. By default, this value has been established in 0.05. Example: edger_cutoff=0.01
fc_threshold	Value to filter low DE expressed mRNAs. As logFC is expressed in log2 a fc_threshold=1 means a real change in expression of 2 folds. Example: fc_threshold=3
miRNAs_folder	Path of the folder with DE miRNAs to obtain DE mRNAs-miRNAs pairs with miRGate. Example: Examples/basic_examples/miRNAs/results/

4.7.3. Examples of configuration file to run Target Prediction module

1) Target prediction analysis of DE mRNAs: In this example, user will perform the target prediction analysis of the DE mRNAs. User will execute miARma from its own directory, the input files are tabulated files with the results of the DE analysis from example 4.3. located in the input directory (Examples/basic_examples/mRNAs/results/ in the example) and the results will be saved in results directory (Examples/basic_examples/mRNAs/results/ in this case). The target prediction will be performed only in those mRNAs with a probability greater than 0.8 in Noiseq analysis and a p-value less than 0.05 in EdgeR analysis.

```
[General]
type=mRNA
verbose=0
read_dir= Examples/basic_examples/mRNAs/reads/
threads=4
label= TSA
miARmaPath=.
output_dir= Examples/basic_examples/mRNAs/results/
organism=human
seqtype=Single
strand=no

[TargetPrediction]
noiseq_cutoff=0.8
```

```
edger_cutoff=0.05
```

This configuration file can be founded in the examples folder of the miARma downloaded directory and can be executed using:

```
perl miARma Examples/basic_examples/mRNAs/6.TargetPrediction/6.1.miRGate.ini
```

2) Target prediction analysis of DE pairs miRNAs-mRNAs: In this example, user will perform the target prediction analysis of the DE mRNAs using as targets the DE miRNAs contained in the miRNAs_folder. User will execute miARma from its own directory, the input files are tabulated files with the results of the DE analysis from example 4.3. located in the input directory (Examples/basic_examples/mRNAs/results/ in the example) and the results will be saved in results directory (Examples/basic_examples/mRNAs/results/ in this case). The target prediction will be performed only in those mRNAs with a probability greater than 0.8 in Noiseq analysis and a p-value less than 0.05 in EdgeR analysis.

```
[General]
type=mRNA
verbose=0
read_dir= Examples/basic_examples/mRNAs/reads/
threads=4
label= TSA
miARmaPath=.
output_dir= Examples/basic_examples/mRNAs/results/
organism=human
seqtype=Single
strand=no

[TargetPrediction]
noiseq_cutoff=0.8
edger_cutoff=0.05
miRNAs_folder=Examples/basic_examples/miRNAs/results/
```

This configuration file can be founded in the examples folder of the miARma downloaded directory and can be executed using:

```
perl miARma Examples/basic_examples/mRNAs/6.TargetPrediction/6.2.miRGate_genes_and_miRNAs.ini
```