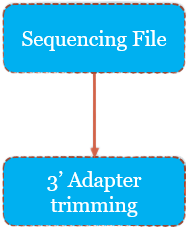
**Phase n.1: pre-processing**

At the beginning of the small RNA-Seq analysis there is a pre-processing step. It consists of:

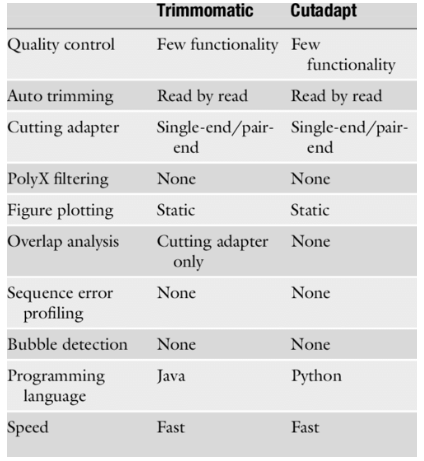
* Quality control;
* Adapters trimming;
* Quality control.

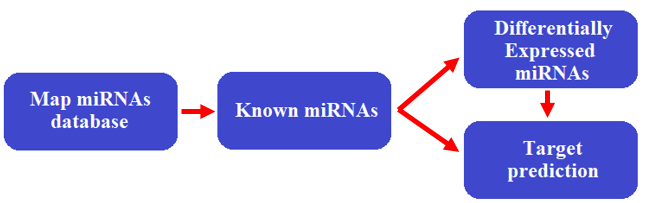
**Step n.1 –** Quality control.

The quality control phases are performed with *FastQC*. In order to visualize all the reports in a single file was used the *MultiQC* tool.

**Step n.2 –** 3’ Adapters trimming.

To cut-off the adapters we make a choice between the two widely used tools: *CutAdapt* and *Trimmomatic*. CutAdapt was chosen for these following reasons:

* Fast enough, easy to use, flexible in how/what you want to trim and what to get back.
* Great documentation, well maintained.
* Write to stdout so you can stream through BWA (or else) without writing massive files to disk.
* Quality trimming is very high, up to 150+ bases.
* Python-based.

**Phase n.2: miRNAs detection**

**Step n.1 –** mapping reads against the reference genome

Alignment is the first step in most RNA-seq analysis pipelines, and the accuracy of downstream analyses depends heavily on it. The majority of RNA-seq studies start with alignment to a reference genome. Analysis is also possible without a reference genome but generally underperforms alignment-guided analysis. After that, reads that are aligned to unique regions are then searched against small RNA databases to establish their identities, while those that are mapped to a large number (e.g.>5000) of genomic locations should be removed from further analysis. The easiest way to deal with multi-mapped small RNA reads is to simply ignore them, but this leads to the loss of great amounts of data. A more commonly used approach is to randomly assign them to one of the mapped positions. More sophisticated algorithms have also been developed to avoid lost precision or sensitivity of these approaches.

For these reasons we choose to use the *ShortStack* tool.

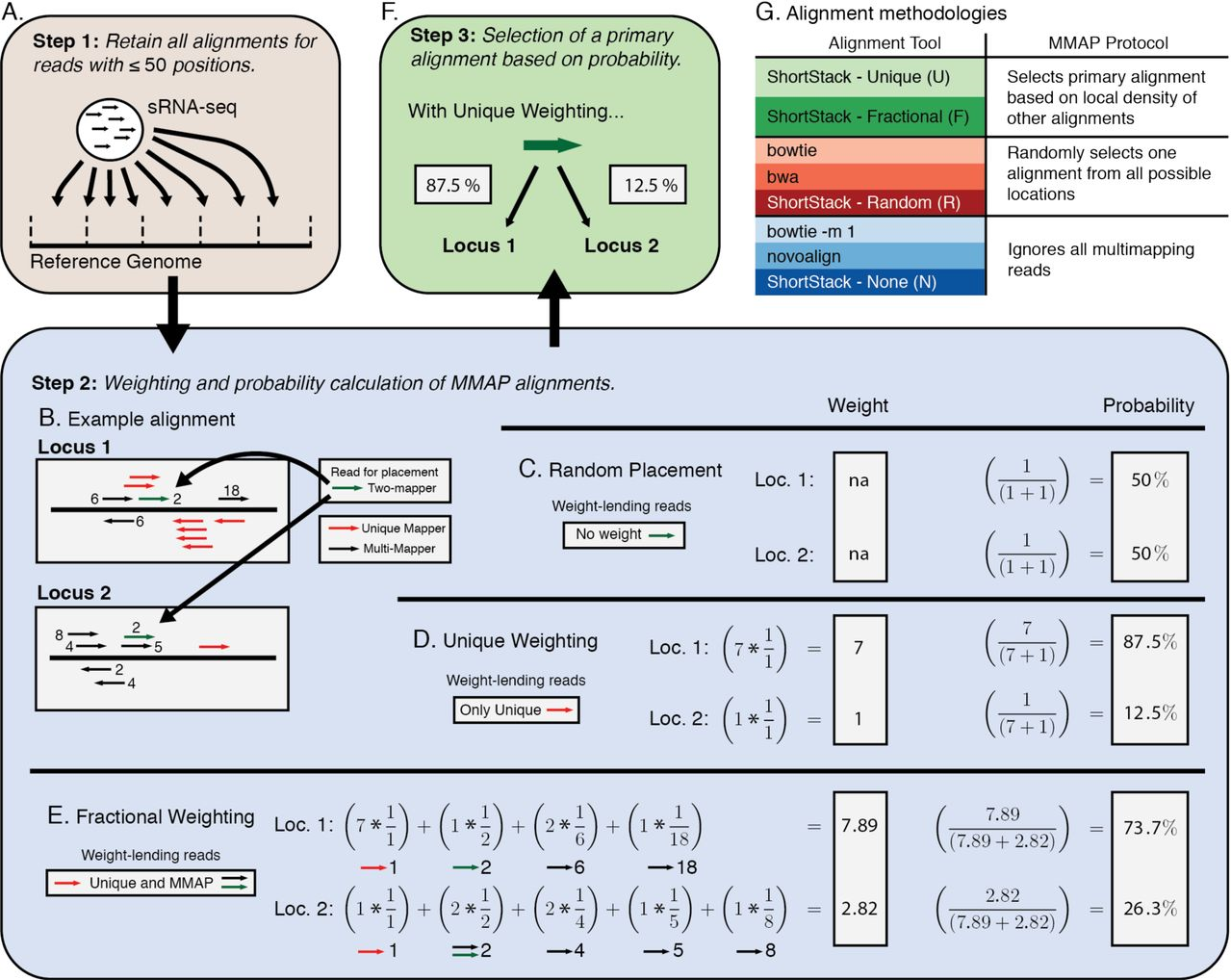
**Briefly description of ShortStack**

The goal of small RNA-Seq alignment to the reference genome is to identify the site of transcriptional origin of the reads.

In the first step of alignment ShortStack uses *bowtie* to identify all possible best-matched alignments for each read, subject to a limit of 500 alignments per read. ShortStack will then calculate a probability for each alignment according to one of three alternative methods (see documentation <https://www.biorxiv.org/content/10.1101/044099v2.full>). The method that was chosen is **unique-weighting mode** (u): it selects the primary alignment based on frequency of other alignment. As we can see in the figure below, locus 1 has 7 uniquely aligned reads in the vicinity, while locus 2 has only 1. The probability of read originating from locus 1 is 87.5%, while the probability of originating from locus 2 is 12.5%.

ShortStack is also able to identify the miRNAs. The main outputs are:

* A bam file, with the aligned reads, used to align the reads within the reference databases.
* miRNA directory, which contains all the identified clusters of miRNAs. This directory is used to perform both the target prediction analysis and the differential analysis.



**Step n.2 –** miRNAs target prediction

The target prediction of miRNAs is performed thanks to *TargetScan* via *mirBase*. In our project the prediction is made upon the mammals, but is possible to make a prediction on mouse, worms, flies and fishes. TargetScan predicts biological targets of miRNAs by searching for the presence of conserved 8mer, 7mer, and 6mer sites that match the seed region of each miRNA. Thanks to a python script, the scraping process allows to queries TargetScan and retrieve the genes that interact with the identified miRNAs.

**Step n.3 –** miRNAs annotations

NGS generates millions of short reads, which are usually aligned to a reference genome. The key information required for downstream analysis is the number of reads mapping to each genomic feature, for example to each exon or each gene. The process of counting reads is called read summarization.

The tool used is *featureCounts*: it implements highly efficient chromosome hashing and is faster than existing methods and requires less memory than other tools.

The data input to featureCounts consists of one or more files of aligned reads in either SAM or BAM format and a list of genomic features in SAF format according to the family of the sncRNA.

Here the manual of featureCounts - <http://gensoft.pasteur.fr/docs/subread/1.4.6-p3/SubreadUsersGuide.pdf>

**Step n.4 –** differential expression analysis

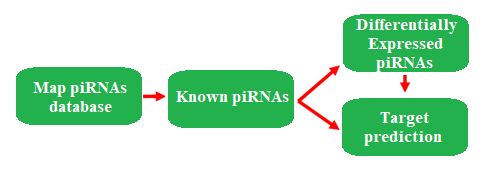
To compare two conditions and make the differential expression analysis we used *DESeq2*. This approach is the de facto standard in differential expression analysis. While *Cuffdiff* tests, for each transcript, if its concentration is not the same in the two different samples, DESeq2 tests, for a given gene, the difference of expression strength between the two conditions.

The user is able to set several filters according to:

* p\_value,
* p\_value\_adjusted,
* log\_2\_fold.

This task allows to plot different graphs:

* Heatmap: we’re miRNAs are showed according to the filters setted by the user.
* Barchart: it shows a normalized value that represents how many miRNAs are present in both conditions.

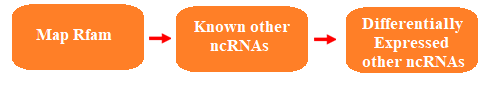
**Phase n.3: piRNAs detection**

**Step n.1 –** discover piRNAs

In this phase the goal is to find piRNAs. The used tool is featureCounts, that takes in input the BAM file of the reads and a SAF file with the annotation about the piRNAs, this file was taken from *pirnaBank database*. The output is a TSW file with the mapped piRNAs and other related information.

**Step n.2 –** differential expression analysis

This step is the same computed for miRNAs.

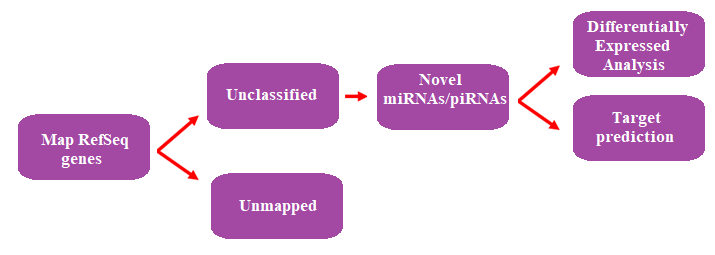
**Phase n.4: other snRNAs detection**

**Step n.1 –** discover other ncRNAs

In this phase the goal is to find other ncRNAs. The used tool is featureCounts, that takes in input the BAM file of the reads and a SAF file with the annotation about other ncRNAs, this file was taken from *daSHR database*. The output is a TSW file with the mapped ncRNAs and other related information.

**Step n.2 –** differential expression analysis

This step is the same computed for miRNAs.

**Phase n.5: novel miRNAs/piRNAs**

**Step n.1 –** discover novel miRNAs/piRNAs

To discover novel miRNAs/piRNAs the BAM file with the reads is aligned within the SAF file representing all the genes in the genome. These two files are the inputs of featureCounts; the output is then used to discover new miRNAs/piRNAs.

* *Novel miRNAs*: the used tool is *mirDeep2*. This is the most used tool for miRNAs prediction.
* *Novel piRNAs*: the used tool is *piRNN*: it was chosen because it is the most efficient tool for novel piRNAs prediction and also because other tools don’t allow to make the analysis on the human genome. piRNN uses a deep learning algorithm based on the free energy of piRNAs.

**Step n.2 –** differential expression analysis

The outputs of piRNN and mirDeep2 are used to make a differential expression analysis on novel miRNAs/piRNAs, as we have seen before.