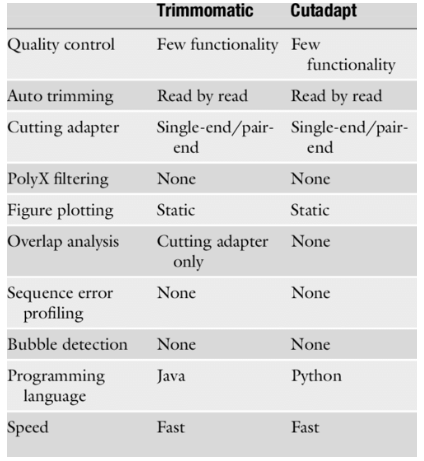
**Step n.1 –** Quality control with FastQC; adapters cut-off with CutAdapt.

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

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

The quality control phases are performed with *FastQC*.

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.

**Step n.2 –** 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 or transcriptome. 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 tool *ShortStack*.

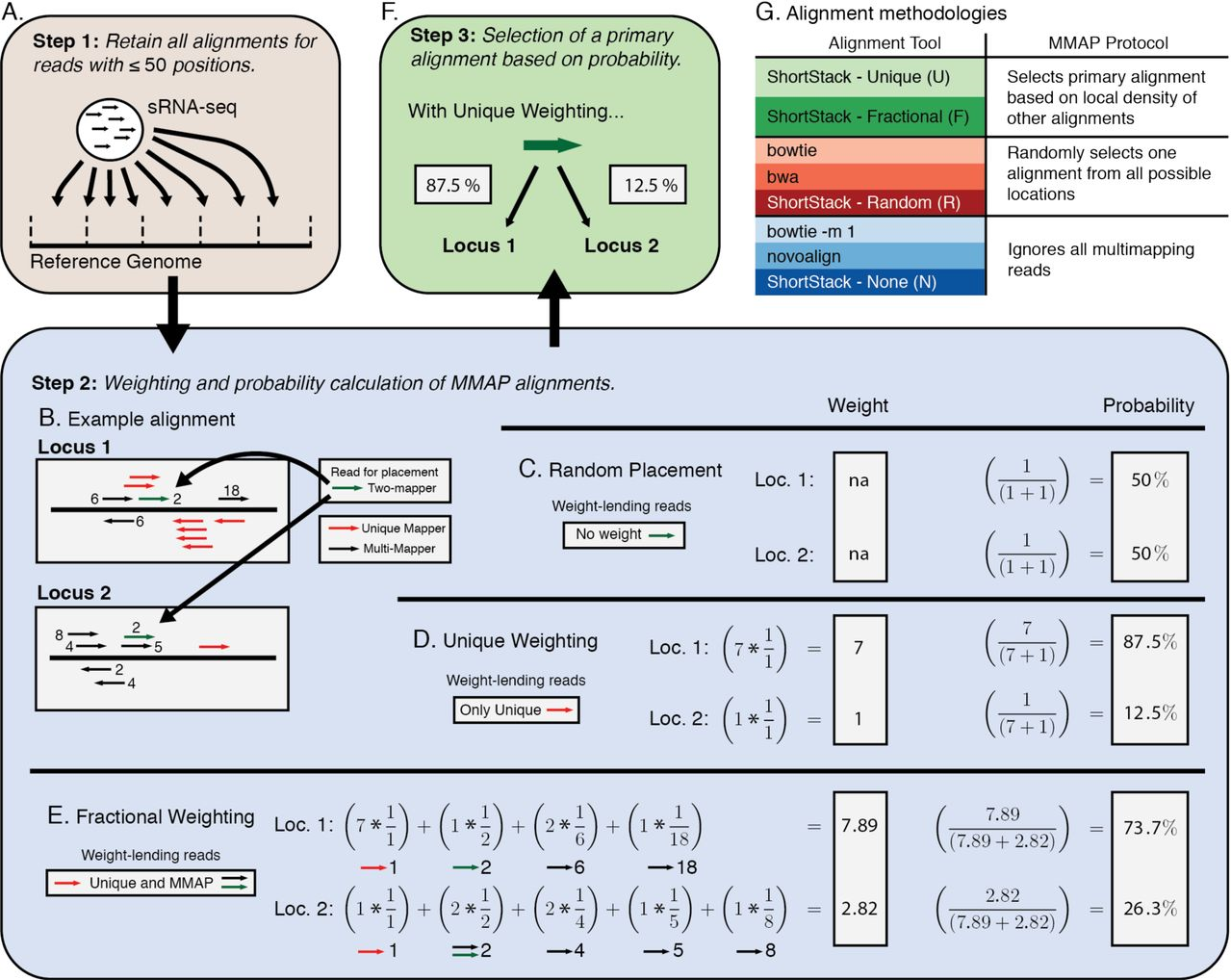
**Briefly description about ShortStack**

<https://www.biorxiv.org/content/10.1101/044099v2.full>

It makes two assumptions. First, each read in a small RNA-Seq experiment represents a single small RNA molecule, which therefore must have had a single genomic origin. Note that this assumption does not mean we assume that all reads with the same sequence necessarily have the same genomic origin. For instance, if we find 100 reads of identical sequence with a MMAP value of two (e.g., two possible alignment positions), it's certainly possible that some of those reads came from one location, and the rest from the other. Second, the goal of small RNA-Seq alignment to the reference genome is to identify the site of transcriptional origin of the small RNAs, not to list their possible targets.

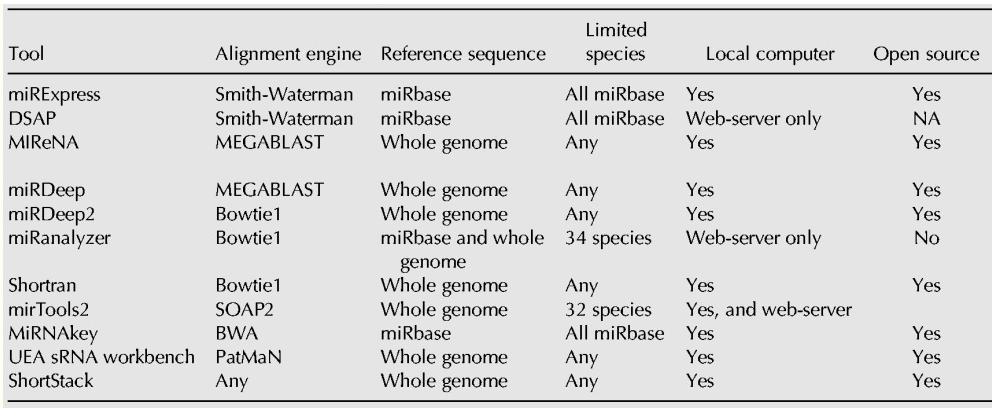
In the first step of alignment ShortStack uses bowtie to identify all possible best-matched alignments for each read, subject to a limit of 50 alignments per read. ShortStack will then calculate a probability for each alignment according to one of three alternative methods.

1. In **unique-weighting mode** (U), the frequencies of uniquely aligned reads mapping within the vicinity of the alignment under consideration are tallied. For the example read in Figure 2, locus 1 has seven uniquely aligned reads in the vicinity, while locus 2 has only one. The probability of read originating from locus 1 is 87.5%, while the probability of originating from locus 2 is 12.5%.
2. The **fractional method** (F) uses all reads mapping within the vicinity of an alignment. Unique reads in the vicinity have a full weight, while MMAP reads provide weights inversely proportional to their MMAP-value. The calculated probabilities are then added to a random number to designate the primary alignment for the read and all the other possible secondary alignments.
3. Alternatively, ShortStack can be instructed to simply **ignore** (N) all MMAP reads.



**Step n.3 –** mapping miRNAs against the reference database (mirBase)

The reads obtained in the previous step are aligned against the reference miRNA database: mirBase. The aligner that we chose is *TopHat*. TopHat is a short-read aligner specifically designed for alignment of RNA sequencing data. It uses the Bowtie aligner, aligning to both the genome and splice junctions without a reference splice site annotation. TopHat was also chosen to be consistent with the choices made before: the indexing process is performed with Bowtie, and Butter uses Bowtie too.



<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5792058/> confronto aligners

**Step n.4 –** miRNAs target prediction

<http://www.targetscan.org/vert_72/>

The prediction of miRNAs is performed thanks to *TargetScan*. In our project the prediction is made upon the mammals, but is possible to make a prediction on mouses, 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. As an option, predictions with only poorly conserved sites are also provided. TargetScanHuman considers matches to human 3' UTRs and their orthologs, as defined by UCSC whole-genome alignments. Conserved targeting has also been detected within open reading frames (ORFs).

**Step n.5 –** miRNAs annotations

<https://bio.tools/featurecounts>

<http://gensoft.pasteur.fr/docs/subread/1.4.6-p3/SubreadUsersGuide.pdf> The manual

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*, a read summarization program suitable for counting reads generated from either RNA or genomic DNA sequencing experiments. featureCounts implements highly efficient chromosome hashing and feature blocking techniques. It is considerably faster than existing methods + and requires far less computer memory.

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 either Gene Transfer Format (GTF) or General Feature Format (GFF).

**Step n. –** differential expression analysis

To compare two conditions and make the differential expression analysis we used *DESeq*.

While *Cuffdiff* tests, for each transcript, whether there is evidence that the concentration of this transcript is not the same in the two samples, DESeq tests, whether, for a given gene, the change in expression strength between the two conditions is large as compared to the variation within each replicate group.