
CircularMapper Documentation

Release 1.94.0

Alexander Peltzer

Jan 25, 2018

Contents

1	General Usage of CircularMapper	3
1.1	Prerequisites	3
1.2	Execution	3

Contents:

CHAPTER 1

General Usage of CircularMapper

1.1 Prerequisites

The CircularMapper requires a running Java Runtime Environment 8 or later to be run on your local machine. In case you want to use it on Windows (which should be entirely possible, but no guarantees on this), you may also run it there from a commandline. You can download the latest release from the GitHub project page, following the links to Bintray as listed in the Readme.

1.2 Execution

The tool consists of two individual tools, the “CircularGenerator” and the “RealignSAMFile” component. The former is used to create a modified reference genome with elongation, whereas the latter is used to realign reads mapped with BWA to the modified reference genome. A typical workflow is shown here:

```
#create a modified reference genome, with an elongation of 500 bases and specifically
↳elongating the mitochondrial chromosome
java -jar CircularGenerator.jar -e 500 -i reference.fasta -s "chrMT"
```

Note: You need to ensure that the specified FastA entry is correctly specified, or this won't work at all.

```
#index the new genome, map against it using bwa
bwa index reference_500.fasta
#using 4 cpu cores here, -n 0.04 (default), -l 32 (default)
bwa align -t 4 reference_500.fasta your.merged.fastqfile -n 0.04 -l 32 -f output.sai
bwa samse -r "specifyyourreadgroupshere" reference_500.fasta output.sai your.merged.
↳fastqfile -f output_circmapper.sam
```

Now we do have a SAM file aligned against our modified reference, thus spanning the circular part of the mitochondrial genome. We can now safely use RealignSAMFile to get the reads matching our unmodified reference genome again. To do this, run this:

Note: There are two ways to do this: Either you want to keep all the reads mapped, or you just want to keep reads mapping to your specified mitochondrial genome (e.g. “chrMT” in this case). We will post both commands here, to ease this step for you!

```
# default, no filtering. Input SAM does not have to be sorted or anything else, ↵
↪output will be BAM already, saving you to convert ;-)
java -jar RealignSAMFile.jar -e 500 -i output_circmapper.sam -r reference.fasta
# with filtering, same command
java -jar RealignSAMFile.jar -e 500 -i output_circmapper.sam -r reference.fasta -f ↵
↪true -x false
```

This will create a file called “output_circmapper_realigned.bam” in the same folder than your input automatically.

```
#again using 4 cpu cores;-)
samtools sort -@ 4 output_circmapper_realigned.BAM -o output_circmapper_realigned.
↪sorted.bam
samtools index output_circmapper_realigned.sorted.bam
```

Now you can use your “reference.fasta” and load this in IGV, or perform any downstream analysis you’d like to.