Quality assessment of NGS data

Ines de Santiago

July 18, 2015

Contents

1 Introduction

When you get your sequences back from a sequencing facility, its important to check that they are high quality. In this tutorial, we will use a software called FastQC which checks whether a set of sequence reads in a .fastq file exhibit any unusual qualities.

2 Checking read quality with FASTQC

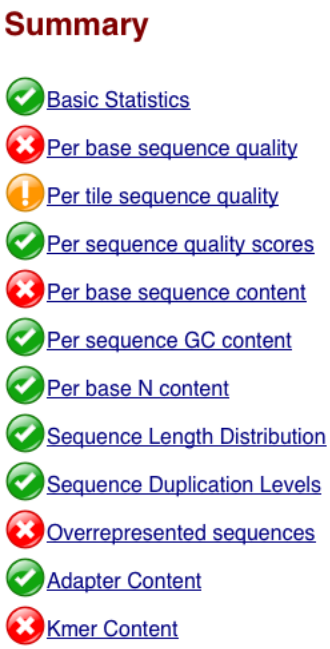
The data we will use in this tutorial consists of a fastq file with example sequences. The file is called "sample.fastq" and is located in the" located in the "/Data_For_QC_Practical/" directory. FastQC generates a html report with a nice graphical summary output about the quality of our sequencing reads.

Use Case: Run FASTQC to check read quality

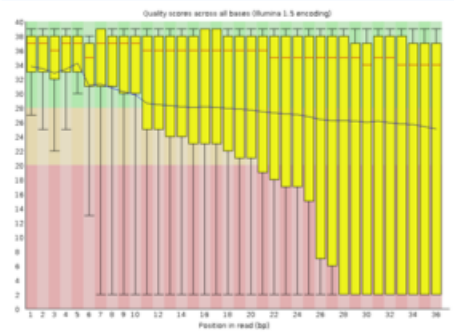
There are two ways in which FastQC can be run: in "command line" mode, or as a GUI (graphical user interface). This exercise addresses the command line version of FastQC.

```
fastqc sample.fastqc
```

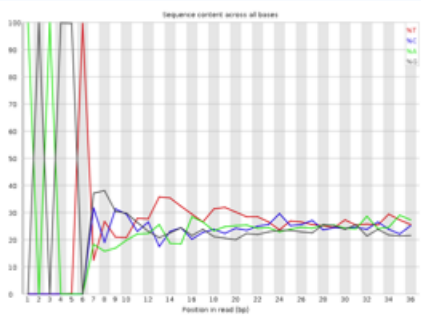
FastQC generates two files , one compressed, and one not. To view the report, open the file "sample_fastqc.html" in a browser. Your output should have a menu on the left-hand side that looks like this:



Make sure you understand the results. Do they look ok? Are there any warnings or errors? What do they mean? Do you have adapter sequences in your reads? What preprocessing do we need to do?



Bad quality -> Use "FASTQ Quality Filter" and/or "FASTQ Quality Trimmer"



Flagged Kmer Content: About 100% of the first six bases are the same sequence -> Use "FASTQTrimmer"

3 Preprocessing with Fastx Toolkit

In this exercise we will be using two preprocessing tools:

- Fastx Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/)
  - fastx_trimmer: Shortening reads in a FASTQ or FASTQ files (removing barcodes or noise).
  - fastq_quality_filter: Filters sequences based on quality

For most programs and scripts, you can see their instructions by typing their name in the terminal followed by the flag -h. There are many options available, and we will use only a few of those.

Use Case: Remove reads with lower quality

```
-i:  input file
-o:  output file
-v:  report number of sequences
-q 20: the quality value required
-p 75: the percentage of bases that have to have that quality value
```

```
fastq_quality_filter -v -q 20 -p 75 -i sample.fastq -o sample_filtered.fastq
```

fastq_quality_filter should give you the following output:

Quality cut-off: 20 Minimum percentage: 75 Input: 9053 reads. Output: 6629 reads. discarded 2424 (26%) low-quality reads.

Use Case: Trim the reads

```
-f: First base to keep
-l: Last base to keep
-i: input file
-o: output file
-v: report number of sequences
```

```
fastx_trimmer -v -f 7 -l 36 -i sample_filtered.fastq -o sample_filtered_and_trimmed.fastq
```

fastx_trimmer should give you the following output:

Trimming: base 7 to 36 Input: 6629 reads. Output: 6629 reads.

4 Conclusions

Dont forget FastQC can tell us that the sequences in the .fastq file have these unusual features, but it cant necessarily explain why; the causes are open to interpretation. It could be due to poor sequencing quality or bad base calling, or maybe we accidentally contaminated our sequencing library with PCR-amplified microsatellite sequences, or maybe our study species just has a lot of repetitive sequences in its genome. If you want more information on what each piece of FastQCs output means, the documentation is available online at http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/Help/