Understanding and Pre-processing Raw Illumina Data

Matt Johnson

October 4, 2013

1 Understanding FASTQ files

After an Illumina sequencing run, the data is stored in very large text files in a standard format known as FASTQ. Although there is no required file extension, frequently these files are saved with a `.fq` or `.fastq` . FastQ files contain both the sequence and the quality of each base call for every read in the run.

Each read is typically listed on four consecutive lines:

- Sequence ID beginning with `@`
- Base calls (DNA Sequence)
- A plus sign
- Sequence quality codes

Therefore, the number of lines in your FASTQ file is four times the number of reads. This means it is relatively easy to figure out the number of reads in a FASTQ file using simple UNIX commands:

Terminal 1: The number of reads in a FASTQ file

```
CBG002827:~ mjohnson$ cat MIRS-read_1.fq | wc -l
 43010636
CBG002827:~ mjohnson$ READS=$(cat MIRS-read_1.fq | wc -l)
CBG002827:~ mjohnson$ expr $READS / 4
10752659
CBG002827:~ mjohnson$
```

Here, the result of `wc` is stored in the variable `$READS`. Since every read is spread across four lines, the number of reads is `$READS` divided by four.

1.1 Sequence ID

The first line of a FASTQ file stores information about the sequencing run:

```
@FCD15YFACXX:3:1101:2082:1990#ACCAGACT/1
```

@FCD15YFACXX	Instrument Name
3	Flowcell lane
1101	Tile number in flowcell lane
2082	X coordinate of cluster
1990	Y coordinate of cluster
ACCAGACT	Sequence barcode for multiplexed sample
/1	Member of pair for paired-end runs

For paired-end runs, one file will have sequence IDs ending in `/1` while the other will have sequence IDs ending in `/2`. When there is no barcode, older versions of Illumina may have a `0` in that position, but newer versions typically use `NNNNNN`.

1.2 Quality Score

The Illumina sequencer records a quality score for the fluorescence of each base. The scoring system is borrowed from Sanger sequencing, known as a Phred score. This score ranges from 0-64, but for Illumina the best score is 41.

In the FASTQ files, the score is converted to one of the ASCII symbols on a standard English keyboard. Each key on the keyboard has a three-digit code associated with it. Since Illumina version 1.3, the scores are converted by adding 64 to the Phred Score.

The following chart shows some typical (average) quality codes for Illumina.

Phred Score	Quality Score	ASCII Code
25	089	Y
26	090	Z
27	091	[
28	092	\
29	093	]
30	094	ˆ
31	095	_
32	096	‘
33	097	a
34	098	b
35	099	c

Therefore, lowercase letters have higher quality scores than uppercase letters.

Scanning through an entire read file, containing perhaps 60 million reads, would be impossible. Instead, a tool such as FastQC can help summarize the quality of the Illumina run.

2 FastQC

As the name implies, FastQC is way to quickly see some summary statistics to check the quality of your Illumina run. It runs both as a GUI (requires Java) and as a command line program.

2.1 Obtaining FastQC

Download FastQC from:

`http://www.bioinformatics.babraham.ac.uk/projects/download.html#fastqc`

Download the version appropriate for your operating system.

2.2 FastQC GUI

When you start FastQC, choose Open from the File menu to select your FASTQ file to open and analyze. You may need to select "FASTQ" from the Filetype dialog. FastQC can read compressed FASTQ files (eg: read1.fq.tar.bz2) but it will take a little longer.

The figures below are for the dataset `miniClimacium_1.fq`, which is one pair of a paried-end.

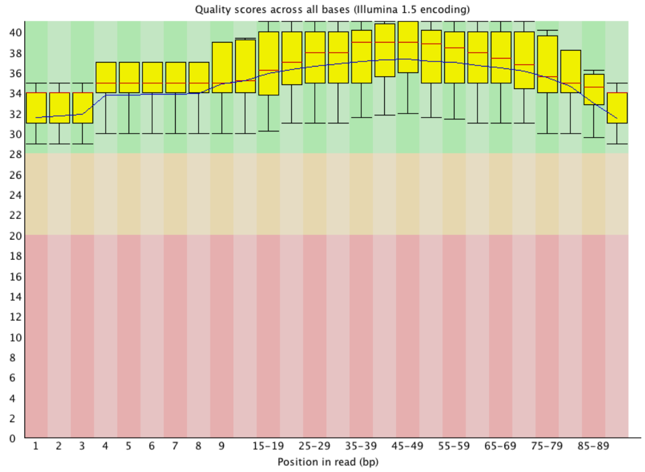
As soon as FastQC is done reading the sequences, it will provide several reports:

2.2.1 Basic Statistics

Here, basic info is listed, such as the total sequence length, the sequencing platform, and GC content.

2.2.2 Per Sequence Quality

This is a very useful graph showing the quality scores for every read in a file, summarized by position. A good run will have quality scores all above 28. If the quality drops below this level after a certain point, you may consider trimming all reads to be shorter than that length.



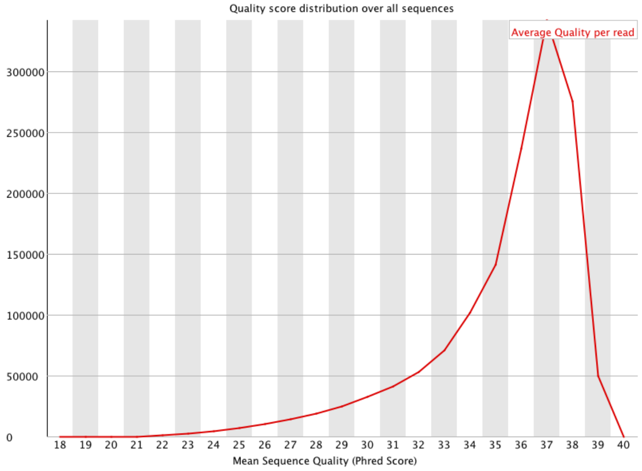
2.2.3 Per sequence quality scores

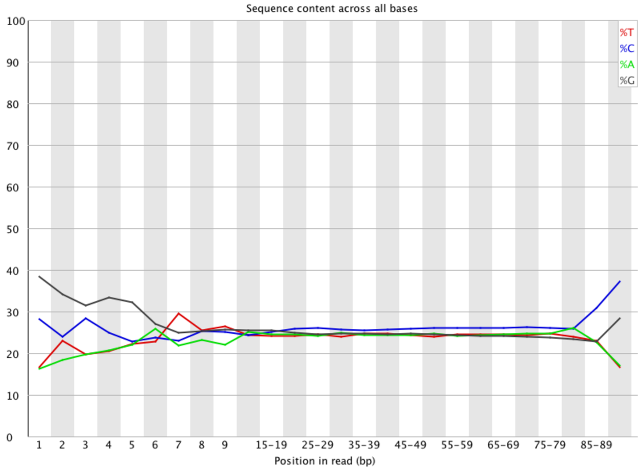
This graph should show a fairly smooth curve with one peak at your overall average quality. In some runs, there may be a cluster of reads with relatively poor scores, which can create a secondary peak. If this occurs, you may consider filtering your reads by average quality, to retain only the best reads.

2.2.4 Per base sequence and GC content

The next two graphs show, by position in the reads, the base call (ACGT) or GC content. Ideal runs should have no variation among base calls or GC content along the length of the read.

In practice, there may be variation at the beginning of reads, especially for RNA-seq. This occurs because during RNA-seq, “random” primers are annealed to the beginning of sequences. It turns out these primers are not truly random, leading to uneven proportions at the beginning of the sequence. See below.



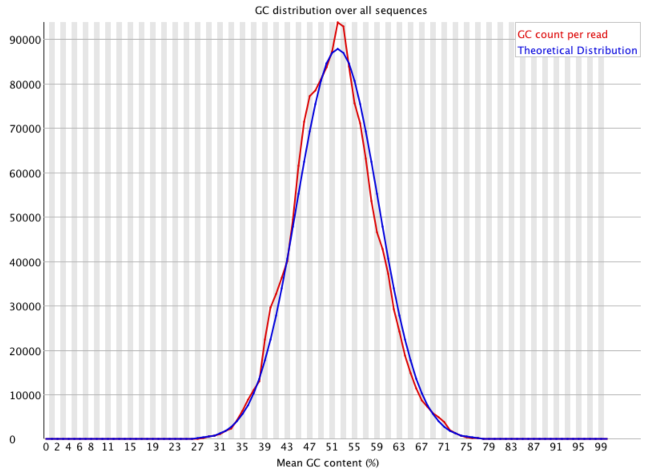


2.2.5 Per Sequence GC Content

This graph shows a theoretical distribution of GC content (blue) and the GC count observed per read (red). An ideal run would have very close matches between the distributions. Secondary peaks and large deviations from the theoretical distribution can indicate contamination.

2.2.6 Per Base N Content

This graph is pretty self-explanatory. The number of unclear base calls by base should be a flat line at 0.



2.2.7 Sequence Length Distribution

This graph is intended more for other platforms, like PacBio and 454, which may have variable lengths. For Illumina, you should see a peak at exactly 90 base pairs.

2.3 Sequence Duplication Levels

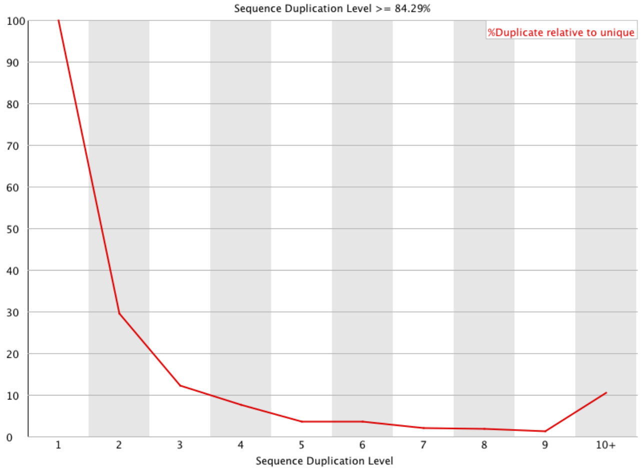
The way FastQC considers sequence duplication is by reading in the first 200,000 reads and creating a database. Then, when it reads the rest of the reads, it checks how many of these reads are duplicates. For this reason, the y-axis in this plot is not an absolute number, but relative to the initial database. Reads that occur only once but after the first 200,000 are not represented.

This graph should drop quickly to zero, but once again there is a caveat for RNA-seq. FastQC was originally written for genome sequencing, where approximately even coverage is expected. By contrast, transcriptome data contains many duplicate reads as a result of uneven expression. Therefore, a tick up in the '10+' category is expected.

2.3.1 Overrepresented Sequences

This section is primarily intended to catch whether primer, barcode, and adapter sequences have been properly trimmed from the sequence. FastQC can detect whether these are recognizable sequences and report this.

For transcriptome data, as with sequence duplication levels, there may be highly expressed genes with very common motifs. Some of these may appear as overrepresented. If there is



nothing listed in the “Possible Source” column, they can be safely ignored.

2.3.2 Kmer Content

This section lists whether a particular pattern of sequences occurs with a certain location in the reads. Ideally, this should be very stable and random, but once again may have odd results for RNA-seq. Non-random primer sequences and over-represented transcripts may cause repeats to occur.

2.4 FastQC Command Line

If the FastQC executable is in your path, you can simply supply it with a list of one or more files:

```
fastqc somefile.fq someotherfile.fq
```

It will generate, in the current directory, a set of reports for each input file. Alternatively you may specify a directory name, into which the reports will go, with: `-o` You may also specify that you want only a compressed (zipped) directory for each report with: `--noextract`

Within each report is an .html file that you can open in your browser to view the full report.

Cygwin Users: There appears to be a bug that prevents Cygwin from correctly executing FastQC. If you want to use FastQC from the command line in Windows, try using it from the DOS command prompt.

3 Trimmomatic

Regardless of the reports from FastQC, it is a good idea to trim your raw reads before assembly. The primary reason for this is to remove poor quality reads that might reduce assembly speed and accuracy.

One popular tool for trimming raw reads is Trimmomatic, available here:
`http://www.usadellab.org/cms/?page=trimmomatic`

Trimmomatic requires Java, but otherwise is fairly straightforward to run. It has several features, including:

- Support for either single or paired-end reads.
- Can trim from either beginning of end of read if quality is below a given threshold.
- Can trim based on a "Sliding Window" if the local quality drops below a given threshold.
- Can trim Illumina adapters.
- Deletes reads below a certain length.

For paired-end reads, it will save only pairs of reads where both members of a pair pass the quality tests. It will also save reads where only one direction passes into a separate file. The components of a typical Trimmomatic command for paired-end reads are:

java -jar	Normal commands to call a java program
Trimmomatic.jar	Or a path to the program if it is not in the current directory
PE	For paired-end
-phred64	Quality score codes, use this for everything since Illumina version 1.3
input1 input2	The two paired read files.
paired_output1 unpaired_output1	Two filenames for the reads in input1. The first file contains reads where the pair also passed the quality checks. The second file is for reads where the pair did not pass.
paired_output2 unpaired_outptu2	Output file names as above, but for input2.
LEADING:3	Trims bases at the beginning of a read if they are below quality score of 3.
TRAILING:3	Trims bases at the end of a read if below quality score 3.

SLIDINGWINDOW:4:15 Scan with a window of size 4 for reads with local quality below a score of 15, and trim if found.

MINLEN:36 Delete a sequence with a length less than 36.

Note that the quality commands are executed in order specified. In the example above, the `MINLEN` command is executed after the other checks, removing reads below length 36 after trimming. If `MINLEN` had been specified before the other commands, it would be useless (on Illumina data)!

Exercise: Using the miniClimacium data available on the workshop remote computer, run Trimmomatic to trim the two read files.