ZYMO RESEARCH
The Beauty of Science is to Make Things Simple

FASTQ File Format

Understanding the FASTQ & QSEQ Raw Sequencing Formats

What is FASTQ Format?

FASTQ format is a human-readable file format that stores the nucleotide base sequences, the calculated confidence for each base in a sequence, and information describing the origin of the read down to its position on the sequencing platform flow-cell. Most, if not all, modern sequencers produce FASTQ files or files that are readily converted to FASTQ files, and nearly all bioinformatics tools dedicated to processing raw sequence and expect FASTQ files as an input. An untrimmed, unfiltered FASTQ file is considered the standard for "raw" sequence in a study and should always be maintained as a permanent part of the study's data.

FASTQ vs. FASTA

FASTA is a file format that was developed in the mid 1980's to store sequence data with annotations. Both protein (single-letter amino acid codes) as well as nucleotide sequences could be stored in FASTA files along with sequence identifier lines and additional annotations covering the locations of genes and other interesting features. FASTQ is an extension of the FASTA file format, with the extension allowing for the storage of sequencing quality data along with the sequence itself and the sequence ID. In general, FASTA files are the most common standard for storing reference or consensus sequence data, while FASTQ is the most common format for storing raw sequence data.

FASTQ vs. QSEQ

FASTQ has not always been the only standard for raw sequencing data, with the QSEQ format providing early competition. Over the last several years, FASTQ has emerged as the clear winner with QSEQ format often only being discussed in historical contexts as a format one may encounter when handling old sequencing data. The typical workflow for handling QSEQ data often starts with simply converting it to FASTQ for downstream analysis. One of the easiest ways to identify QSEQ data is that it will have much of the same data as a FASTQ, but instead of 4 lines per read, there will only be 1 line per read.

Below is an example of the sequencing data found in a FASTQ file representing a given read:

All FASTQ files will consist of a set of reads, with each read having 4 lines of data. The first line of each set will always begin with an "@" symbol and is often called the sequence identifier or the metadata line. The exact values on this line will vary between sequencing platforms or sequence databases, with the example above being generated by CASAVA version 1.8 on Illumina short-read data. The second line will always contain the raw nucleotide sequence and should only contain raw nucleotide sequence. The third line is a spacer that will start with a "+". There may be additional comments put on this line, but important data will generally not be found there. The fourth line will contain the quality string and should only contain the quality string. In the case above, we can break down the dataset as follows:

"@" represents the start of the FASTQ read dataset and all FASTQ read datasets will always begin with the "@" character.

"MM123" refers to the **machine ID**. As the name implies, this identifier is unique to the machine regardless of the brand or model of the sequencing instrument.

"002" lists the **run session** of the Illumina sequencing instrument.

"FC123AB" specifies the **flow cell ID** which is unique to every flow cell. This allows for the identification of which specific flow cell the sample was sequenced on. The flow cell is the major solid consumable used with each sequencing run (along with liquid chemicals and buffers). Each flow cell is divided into lanes (often 8 for Illumina), with each lane being divided into a grid pattern of tiles and positions of each read being measured within tiles.

"3" refers to the **lane** that the sample was sequenced on. In this case, the above read was lane 3 of flow cell FC123AB.

"2208" represents the specific **tile** of the lane the sample was sequenced on.

```
@MM123:002:FC123AB:3:2208:3330:9840 2:Y:18:ATCACG
AGGATACTAGCATAGATACCCTAGATAGTCATAGATCATGATAGGGAGATCTA
+
IJJJJJIIIIIJIIIIIFFFEEEEEDDDDDDCABBBBB@@00))))*(*&%!
```

“3330” refers to the **x-coordinate** of the location on the tile that the sample was sequenced on. This value can be negative or zero.

“9840” refers to the **y-coordinate** of the location on the tile that the sample was sequenced on. This value can be negative or zero as well.

“2” represents the **read direction** of this file. A value of “1” would denote a **Read 1** or **forward read** sequence and a value of “2” as shown here specifies that this sequenced read is a **Read 2** or **reverse read** generated from your starting fragment.

“Y” denotes whether this read was successful in **pass filter**. A “Y” represents that the sample has passed filtering while an “N” specifies that the sample did not pass filtering. Again, pass filter is a measure of quality and serves as an internal QC procedure conducted by the Illumina instrument on how statistically accurate the machine was in determining the true sequence of a read.

“18” refers to a parameter known as **control bits**.

“ATCACG” specifies the **index barcode sequence** used to ID the sample during library preparation. This will be the same, exact sequence found in the index primer that was used.

The sequence ID line, which will always start with the “@” symbol, combined with some basic record keeping, can allow us to identify the exact provenance of any read. We are capable of tracing a given read back to the individual run on an individual machine where it was generated, and we can identify from which specific flow cell, which lane within the flow cell, and which position within the lane our read originated. Combined with records from the sequencing facility, we can also determine exactly when the sequencing data was generated.

“ATCACGAGGATACTAGCATAGATACCCTAGATAGTCATAGATCATGATAGGGAGATCTA” represents the entire base pair **sequence** of the read determined by the Illumina instrument.

“+” is a character that acts as a **separator** between the determined base pair sequence (shown above) and the subsequent data that is known as a **quality string**. The separator character will always take up an entire line on its own with the subsequent quality string being displayed in the next line below it.

“IJJJJJJIIIIIJIIIIIFFFFEEEEDDDDDDCABBBBB@@00))))*)()&%!” is known as the quality string.

Strings in programming are defined as a sequential sequence of characters. For instance, a string can be represented as AT6GZ+59%!. In the case of FASTQ files, these characters will be comprised of only alphanumerical characters and common punctuation symbols.

Quality strings are a sequence of alphanumerical characters (a string) with each individual character encoding for a probability value; this probability refers to the statistical likelihood that a determined base pair call in a sequenced read was accurately performed. This probability is represented by a **Phred quality score**. It is important to realize that each single character in a quality string is associated with its own individual base that was determined in a sequence of interest. As such there is always an individual quality score for each single corresponding base pair.

ZYMO RESEARCH

The Beauty of Science is to Make Things Simple

The **BEAUTY** *of* **SCIENCE** *is to Make Things* **SIMPLE**®