Quality Control of Sequence data

Expected learning outcomes

Modern sequencing technologies can generate a massive number of sequence reads in a single experiment. However, no sequencing technology is perfect, and each instrument will generate different types and amounts of error. Therefore, it is necessary to understand, identify and exclude error-types that may impact the interpretation of downstream analysis.

The objective of this activity is to understand some relevant properties of raw sequence data. We will focus on properties such as length, quality scores and base and k-mer distribution in order to assess the quality of the data and discard low quality or uninformative reads.

We will use basic UNIX commands, FastQC, and Trimmomatic applied to several datasets. We will use datasets from several source experiments (genome shotgun, amplicons, RAD-tags, metagenome shotgun and microRNA) so that you can see the particularities of the different data types.

Getting Started

Modern sequencing technologies can generate a huge number of sequence reads in a single experiment. However, no sequencing technology is perfect, and each instrument will generate different types and amounts of error. Therefore, it is necessary to understand, identify and exclude error types that may impact the interpretation of downstream analysis. The objective of this activity is to understand some relevant properties of raw sequence data. We will focus on properties such as length, quality scores and base and k-mer distribution in order to assess the quality of the data and discard low quality or uninformative reads.

Filtering of low quality data and trimming of erroneous sequences such as adaptors are important for all applications. For example, low quality data is more likely to contain errors. If this is used for a genome assembly it may lead to errors in the genome.
We will be using bioinformatics tools to review, analyse and filter our sequencing data based on quality. This part of the workshop will involve using basic UNIX commands, **FastQC**, **cut-adapt** and **fastq-mcf** (more programs are available - see end of document). We will be reviewing datasets from a number of different experiments including genome shotgun, RAD-Seq and amplicons. By the end of this activity, you should feel more comfortable with assessing the quality of raw sequencing data and with filtering the data to improve the quality. All data for this workshop can be found in
`~/workshop_data/quality_control/`

Details on the FASTQ format

Although it looks complicated (and it is), it's easy to understand the fastq format with a little decoding. Some rules about the format include…

Line	Description
1	Always begins with '@' and then information about the read
2	The actual DNA sequence
3	Always begins with a '+' and sometimes the same info in line 1
4	Has a string of characters which represent the quality scores; must have same number of characters as line 2

We can view the first complete read in one of the files our dataset by using `head` to look at the first four lines.

```
$ head -n4 SRR098026.fastq
@SRR098026.1 HWUSI-EAS1599_1:2:1:0:968 length=35
NNNNNNNNNNNNNNNNCNNNNNNNNNNNNNNNNNN
+SRR098026.1 HWUSI-EAS1599_1:2:1:0:968 length=35
!!!!!!!!!!!!!!!!#!!!!!!!!!!!!!!!!!!
```

All but one of the nucleotides in this read are unknown (`N`). This is a pretty bad read!
Line 4 shows the quality for each nucleotide in the read. Quality is interpreted as the probability of an incorrect base call (e.g. 1 in 10) or, equivalently, the base call accuracy (eg 90%). To make it possible to line up each individual nucleotide with its quality score, the numerical score is converted into a code where each individual character represents the numerical quality score for an individual nucleotide. For example, in the line above, the quality score line is:

```
!!!!!!!!!!!!!!!!#!!!!!!!!!!!!!!!!!!
```

The `#` character and each of the `!` characters represent the encoded quality for an individual nucleotide. The numerical value assigned to each of these characters depends on the sequencing platform that generated the reads. The sequencing machine used to generate our data uses the standard Sanger quality PHRED score encoding, using by Illumina version 1.8 onwards. Each character is assigned a quality score between 0 and 40 as shown in the chart below.

```
Quality encoding: !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHI
                  |         |         |         |         |
Quality score:    0........10........20........30........40
```

Each quality score represents the probability that the corresponding nucleotide call is incorrect. This quality score is logarithmically based, so a quality score of 10 reflects a base call accuracy of 90%, but a quality score of 20 reflects a base call accuracy of 99%. These probability values are the results from the base calling algorithm and dependent on how much signal was captured for the base incorporation.

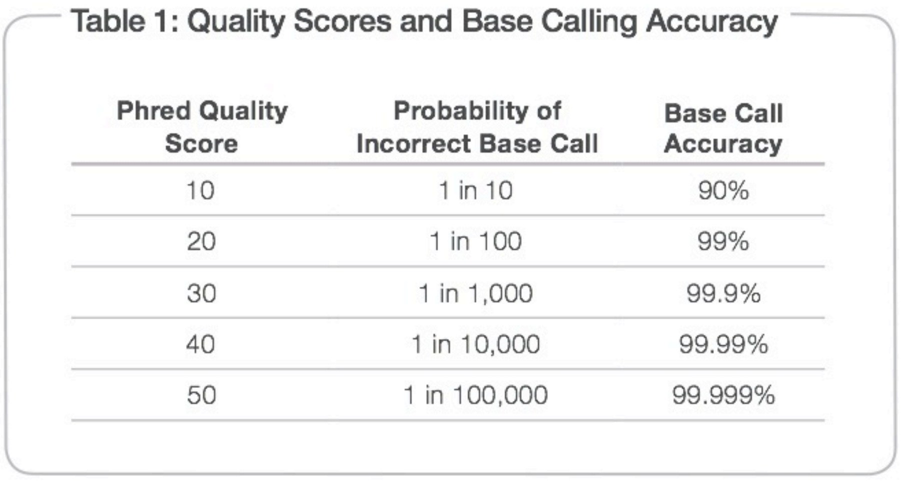

Table 1: Quality Scores and Base Calling Accuracy

Phred Quality Score	Probability of Incorrect Base Call	Base Call Accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1,000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%

```
@SRR098026.1 HWUSI-EAS1599_1:2:1:0:968 length=35
NNNNNNNNNNNNNNNNCNNNNNNNNNNNNNNNNNN
+SRR098026.1 HWUSI-EAS1599_1:2:1:0:968 length=35
!!!!!!!!!!!!!!!!#!!!!!!!!!!!!!!!!!!
```

Looking back at our read we can now see that the quality of each of the `N`s is 0 and the quality of the only nucleotide call (`C`) is also very poor (`#` = a quality score of 2). This is indeed a very bad read.

What is the last read in the `SRR098026.fastq` file? How confident are you in this read?

Answer:

```
$ tail -n4 SRR098026.fastq
@SRR098026.249 HWUSI-EAS1599_1:2:1:2:1057 length=35
CNCTNTATGCGTACGGCAGTGANNNNNNNGGAGAT
+SRR098026.249 HWUSI-EAS1599_1:2:1:2:1057 length=35
A!@B!BBB@ABAB#########!!!!!!!######
```

The second half of this read is poor quality. Many of the positions are unknown (Ns) and the bases that we do have guesses for are of very poor quality (#). However, the beginning of the read is fairly high quality. We will look at variations in position-based quality in just a moment.

Assessing Quality using FastQC

In real life, you won't be assessing the quality of your reads by visually inspecting your FASTQ files. Rather, you'll be using a software program to assess read quality and filter out poor quality reads. We'll first use a program called FastQC to visualize the quality of our reads.

FastQC is a software programme that assesses the quality of sequencing data and produces a report summarising the outcomes in graphical format. For more information take a look at the FastQC help pages. FastQC can be run using a graphical user interface or using the command line. Today we will explore both options.

Exercise 1:

Checking Illumina data with FastQC The goal of this exercise is to learn how to inspect sequence data using FASTQC, both using the user interface and command line. Keep at hand the documentation for FastQC.

DATASET 1: SRR098026.fastq

Have a look at the `SRR098026.fastq` file.

Hint: Use less or more.

1. How many reads are there?

   Hint: Use grep or wc.

2. Launch **FastQC** by typing `fastqc` in the terminal window. Note that if you want to be able to use the terminal after you launch `fastqc,` you need to add an "&" at the end of the command. Also, note that you can have several terminal windows open at the same time.
3. Load the `SRR098026.fastq` file into FastQC (File->Open). You can view the results either within the FastQC application or the exported report.
4. Inspect the data contained in the sequence file, `SRR098026.fastq`. Have a look at the numbers output on the "Basic Statistics" page. How many sequences do we have? What is the sequence length? And the GC content?

   Answer: There are 249 sequences of 35 nucleotides length. The total GC content is 45%.

5. Examine the "Per base sequence quality" and "Per sequence quality scores" pages. Roughly, how many incorrect base calls are expected at most positions? Do you think this run gave good quality sequences?

   Answer: The base calls all have quality score < 4. The sequences are of very bad quality

6. Examine the "Per base sequence content", "Per base GC content" and "Per sequence GC content" pages. FastQC points out a "potential problem" with an orange exclamation mark. Do you think we should worry about it in this particular case?

   Hint: Look at the Evaluating Results page of FastQC.

7. Examine the "Overrepresented sequences" page. Why does FastQC give a warning message?

   Hint: It identified a sequence that is repeated 17 times and that could be an adaptor contamination.

DATASET 2: SRR097977.fastq.

1. Inspect the data contained in the sequence file, `SRR097977.fastq`. Have a look at the numbers output on the "Basic Statistics" page. How many sequences do we have? What is the sequence length? And the GC content?

   Answer: There are 249 sequences of 36 nucleotides length. The total GC content is 44%.

2. Examine the "Per base sequence quality" and "Per sequence quality scores" pages. Roughly, how many incorrect base calls are expected at most positions? Do you think this run gave good quality sequences?

   Answer: The expected incorrect base calls range from 1 in ~1000 (quality score = 28) to 1 in ~5000 (quality score = 37). The base calls all have quality score > 28. The sequences are average good quality.

3. Examine the "Per base sequence content", "Per base GC content" and "Per sequence GC content" pages. FastQC points out a "potential problem" with an orange exclamation mark. Do you think we should worry about it in this particular case?

   Hint: Look at the Evaluating Results page of FastQC.

4. Examine the "Overrepresented sequences" page. Why does FastQC give a warning message?
5. Hint: It identified several overrepresented sequence that is found only once that could be an adaptor contamination.

Exercise 2:

Quality filtering and trimming adaptor sequences using Trimmomatic Depending on the downstream program that will be used, we may need to quality filter and/or trim the adaptor sequences. We will work with datasets 4 and 5.

DATASET 3: SRR026762-sample.fastq: a subset of an Illumina run belonging to a microRNA profiling experiment in human embryonic stem cells

1. Load the data into FASTQC and make a diagnosis of the data. What is the source of the overrepresented sequences?

   Hint: There are two types (excluding the only "N" sequences); Hint 2: Use Blast and look at the description of the dataset.

2. Use Trimmomatic in single end mode to remove low quality bases and trim adaptor sequence. The adaptor used to generate this library was SmallRNA3pAdapter_1.5 ATCTCGTATGCCGTCTTCTGCTTG.

   Hint: Note that Trimmomatic provides adaptor sequences, whcih we have placed in your working directory; however, the adaptor sequence you need here may not be included.

3. Look at the resulting file in FASTQC and make a new diagnosis of the data. What do you observe?

   Hint: There is still some undesired sequences resulting from the library preparation/sequencing process remaining, problably due to "primer-dimers" created during the library preparation and sequencing.

4. Try to identify those sequences by increasing the k-mer length in fastqc (command line).
5. If time allows, go through another clipping step to remove the remaining contaminant sequences

   Hint: The sequencing primer used to sequence this library and which is probably causingthe "primer-dimers" is SmallRNASequencingPrimer (CGACAGGTTCAGAGTTCTACAGTCCGACGATC)

6. Examine the resulting file in FastQC once more. Are you satisfied with the outcome?

See also

Other tools to verify quality of second-generation sequencing results are available:

- Fastx-toolkit:a collection of command line tools for Short-Reads FASTA/FASTQ files preprocessing.
- Galaxy, a web-based genomics pipeline, in which FASTX-Toolkit and FastQC are integrated.
- PRINSEQ, either as a standalone package or through a web-interface can generate summary statistics of sequence and quality data, which can subsequently be used to filter, reformat and trim next-generation sequence data.
- Perl and Bioperl, to write small scripts. There already exist a very large number of packages devoted to genomics in Bioperl.
- R and Bioconductor, are other solution to import and verify data. There are many contributed packages and modules available.