Preliminary Processing of NGS Data

Mark Dunning

23 January 2018

Practical setup

Go to one of following Galaxy servers

- Surnames: A-K
  - https://bioinf-galaxian.erasmusmc.nl/galaxy/
- Surnames: L-N
  - http://galaxy.dbcls.jp/
- Surnames: O-Z
  - http://services.cbib.u-bordeaux.fr/galaxy/

Data Upload

- Go *Get Data*
  - *Upload File*
  - *Choose local file*
  - Select `JoeBlogsBRCAPanel_R1.fastq`, `JoeBlogsBRCAPanel_R2.fastq`, `JoeBlogsBRCAPanel_R1.fastq.gz`, `JoeBlogsBRCAPanel_R2.fastq.gz`

Outline

- The *Fastq* format for sequencing reads
- Quality assessment of *fastq* files
- The *bam* format for representing *aligned* reads
- Stage 1 of an analysis *pipeline*

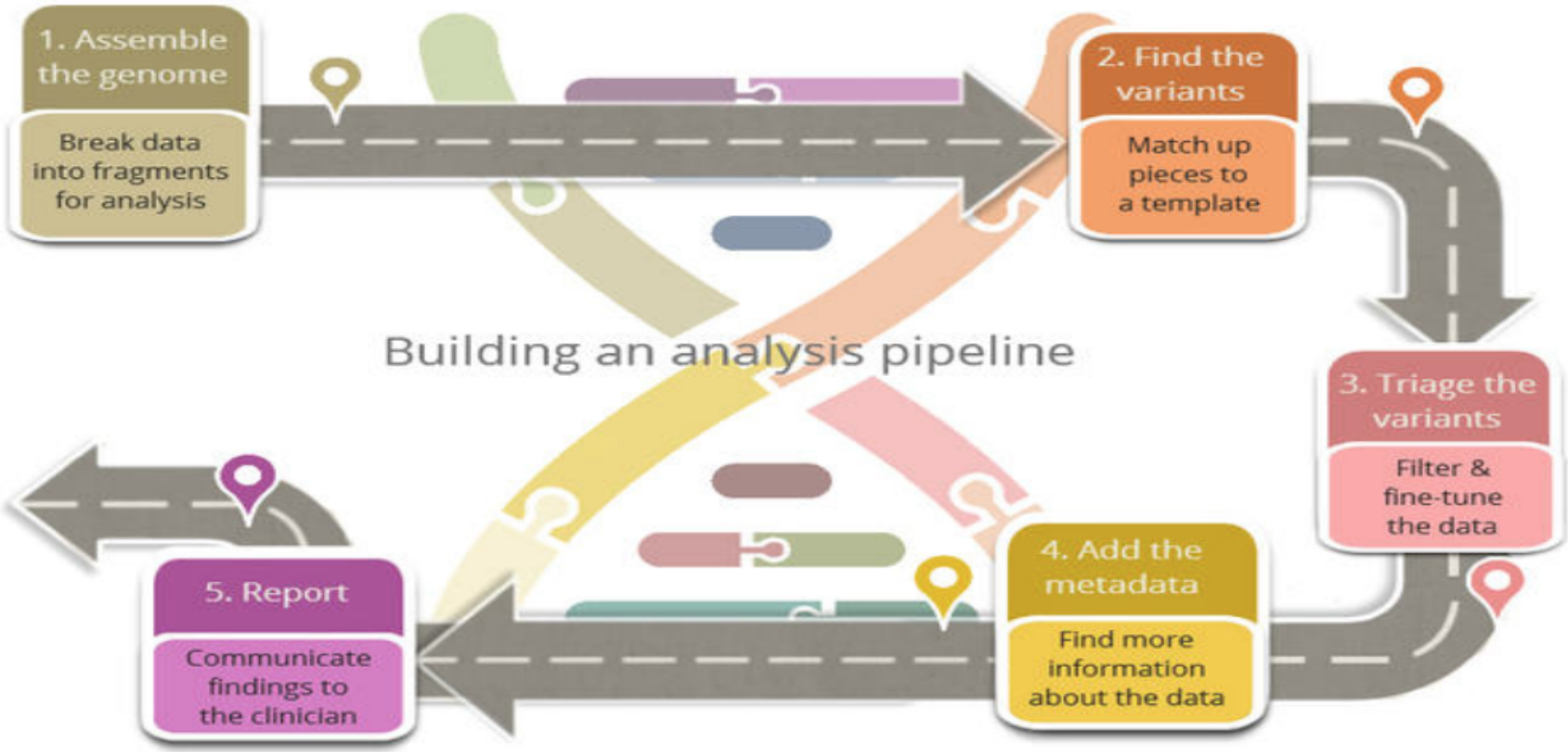
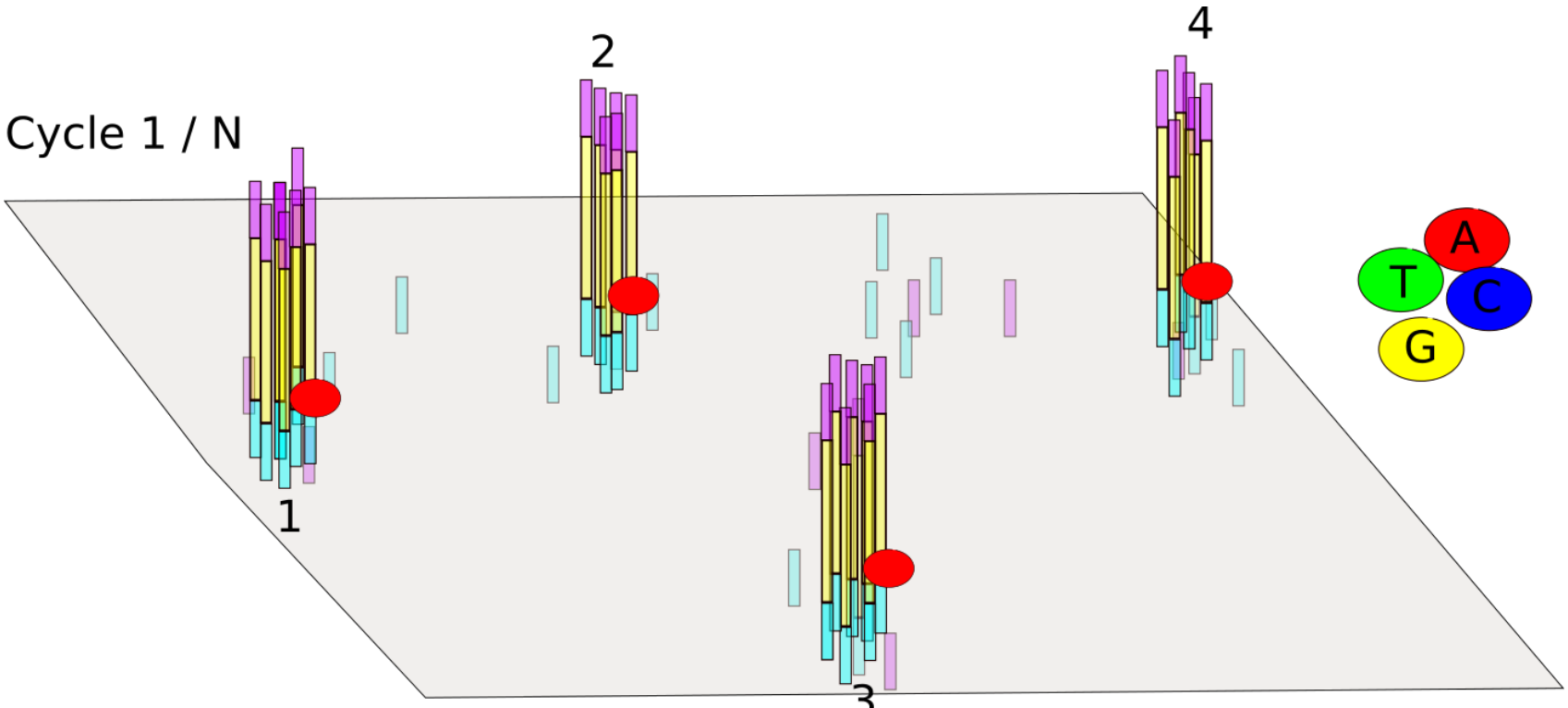


Figure 1

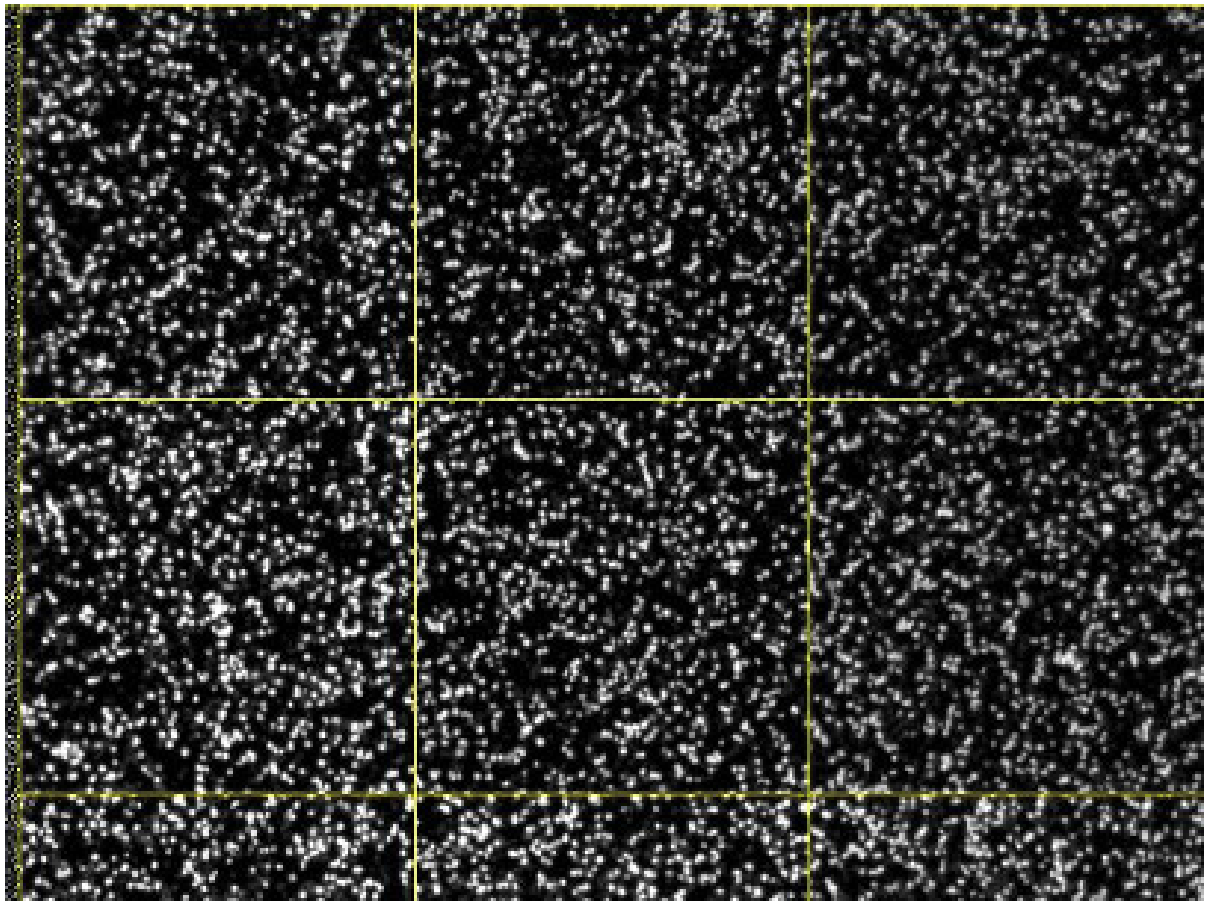
The sequencing process

- https://www.youtube.com/embed/HMyCqWhwB8E for a nice video
- sequencing consists of a series of *cycles* (e.g. 100), at each cycle we try and incorporate different bases (A, T, C, G)
- the base that is successfully added will illuminate brightly in a particular colour



Imaging

- much of the sequening time is spent taking images of the flowcell
- it is these images that are used to discover what fragments of DNA were sequenced
- this process is not perfect and can introduce *uncertainty*



Scale of data

Instrument	No. of Reads	Size
Ion Torrent PGM	5 million reads	1Gb
MiSeq	25 million reads	6GB
HiSeq rapid run	600 million	150GB (*)
HiSeq high-output	4 billion	1 TB

- Equivalent to **40** HD movies
- File sizes are for 100 bp reads, unzipped
- Number of reads from thermofisher.com and illumina.com

Fastq format

- a text file
- can be *compressed* as a gz file
- four lines per read
- the sequenced is most interesting, there are two other lines that we potentially investigate

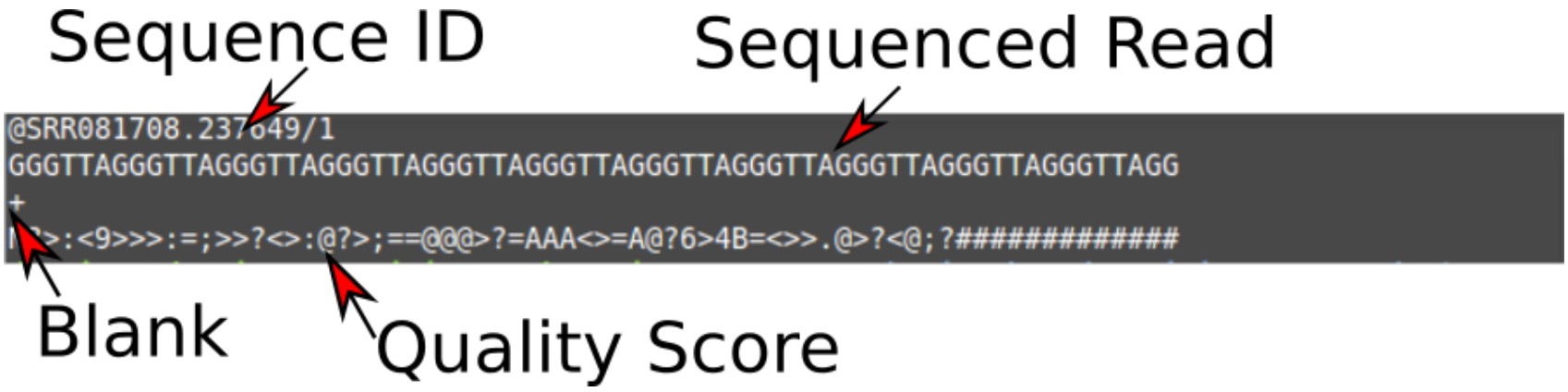


Figure 4

Sequence ID

Can contain the following

- name of sequencer
- flow cell *lane*
- coordinates of the read on the flow cell
- whether this is a *paired* read and whether it is read 1, or 2

Quality scores

- base-calling has some probability (p) that we make a mistake.
- the quality score expresses our *confidence* in a particular base-call; higher quality score, higher confidence
- one such score for each base of sequencing. i.e. 100 scores for 100 bases of sequencing
- these are of importance if we want to call SNVs etc.
  - need to be sure that differences detected from the reference genome are real, and not caused by sequencing error

```
N?>:<9>>>:=;>>?<>:@?>;==@@@>?=AAA<>=A@?6>4B=<>>.@>?<@;?####
```

Deriving the Quality Score

First of all, we convert the base-calling probability (p) into a `Q` score using the formula

- Quality scores

$$Q = -10 log_{10} p$$

  - Q = 30, p=0.001
  - Q = 20, p=0.01
  - Q = 10, p=0.1

- These numeric quanties are *encoded* as **ASCII** code
  - At least 33 to get to meaningful characters (https://en.wikipedia.org/wiki/FASTQ_format)

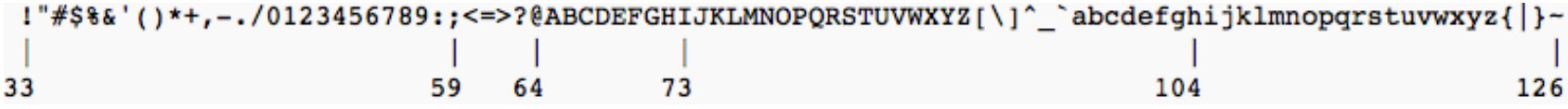


Figure 5

Quality Scores to probabilities

- look-up the ASCII code for each character
- subtract the offset to get the Q score
- convert to a probability using the formula:-

$$p = 10^{-Q/10}$$

Worked Example

for our particular example:

```
N?>:<9>>>:=;>>?<>:@?>;==@@@>?=AAA<>=A@?6>4B=<>>.@>?<@;?####
```

it works out as follows:-

	Character	Code	Minus.Offset..33..	Probability
1	N	78	45	0.00003
2	?	63	30	0.00100
3	>	62	29	0.00126
4	:	58	25	0.00316
5	<	60	27	0.00200
6	9	57	24	0.00398
7	>	62	29	0.00126
8	>	62	29	0.00126
9	>	62	29	0.00126
10	:	58	25	0.00316

...

Exercise

- Use the Galaxy tool *Text Manipulation -> Select last*
  - print the last *12* lines from the file `JoeBlogsBRCAPanel_R2.fastq`
- How many reads are shown in the result?
- Look at the last read and write down the first five and last five ASCII characters
  - is the quality greater at the start, or the end of the read?
- Use the Galaxy tool *Text Manipulation -> Line/Word/Character count*
  - count how many lines are in the file `JoeBlogsBRCAPanel_R2.fastq` in total
  - how many reads does this correspond to?

FastQC: Quality Assessment of fastqc

- FastQC from the Babraham Institute Bioinformatics Core has emerged as the standard tool for performing quality assessment on sequencing reads; https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
- The manual for `fastqc` is available online and is very comprehensive
- A *"traffic light"* system is used to draw your attention to sections of the report that require further investigation.
- `fastqc` will not actually *do* anything to your data. If you decide to trim or remove contamination for your samples, you will need to use another tool.
- it doesn't know what type of sequencing has been performed (WGS, exome, RNA-seq), which can affect interpretation of some of the plots

Example sections of a `fastqc` report

1. Basic Statistics

Basic Statistics

Measure	Value
Filename	sample.fastq
File type	Conventional base calls
Encoding	Illumina 1.5
Total Sequences	9053
Sequences flagged as poor quality	0
Sequence length	36
%GC	50

Figure 6

Example sections of a `fastqc` report

2. Per-base sequence quality

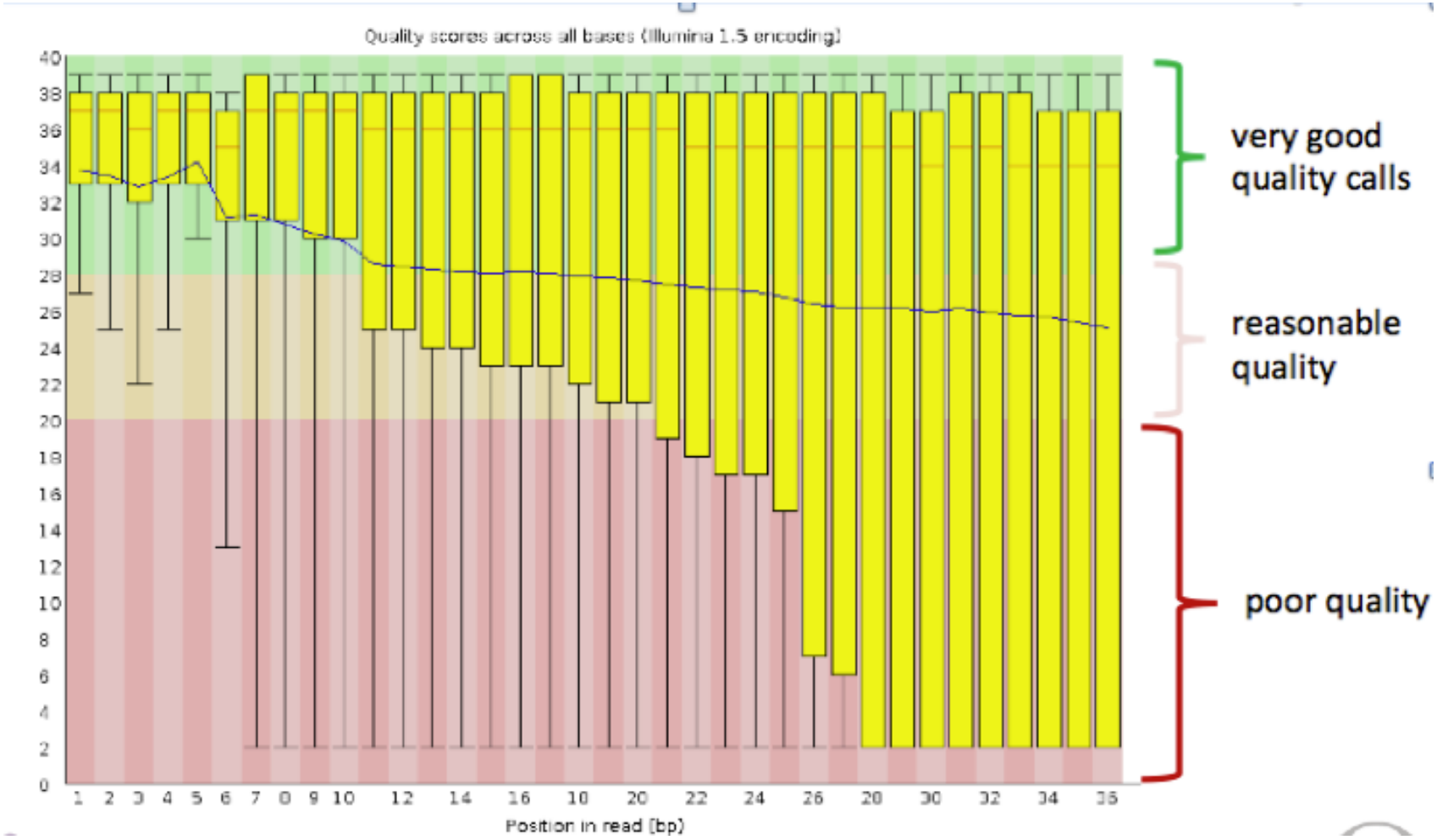
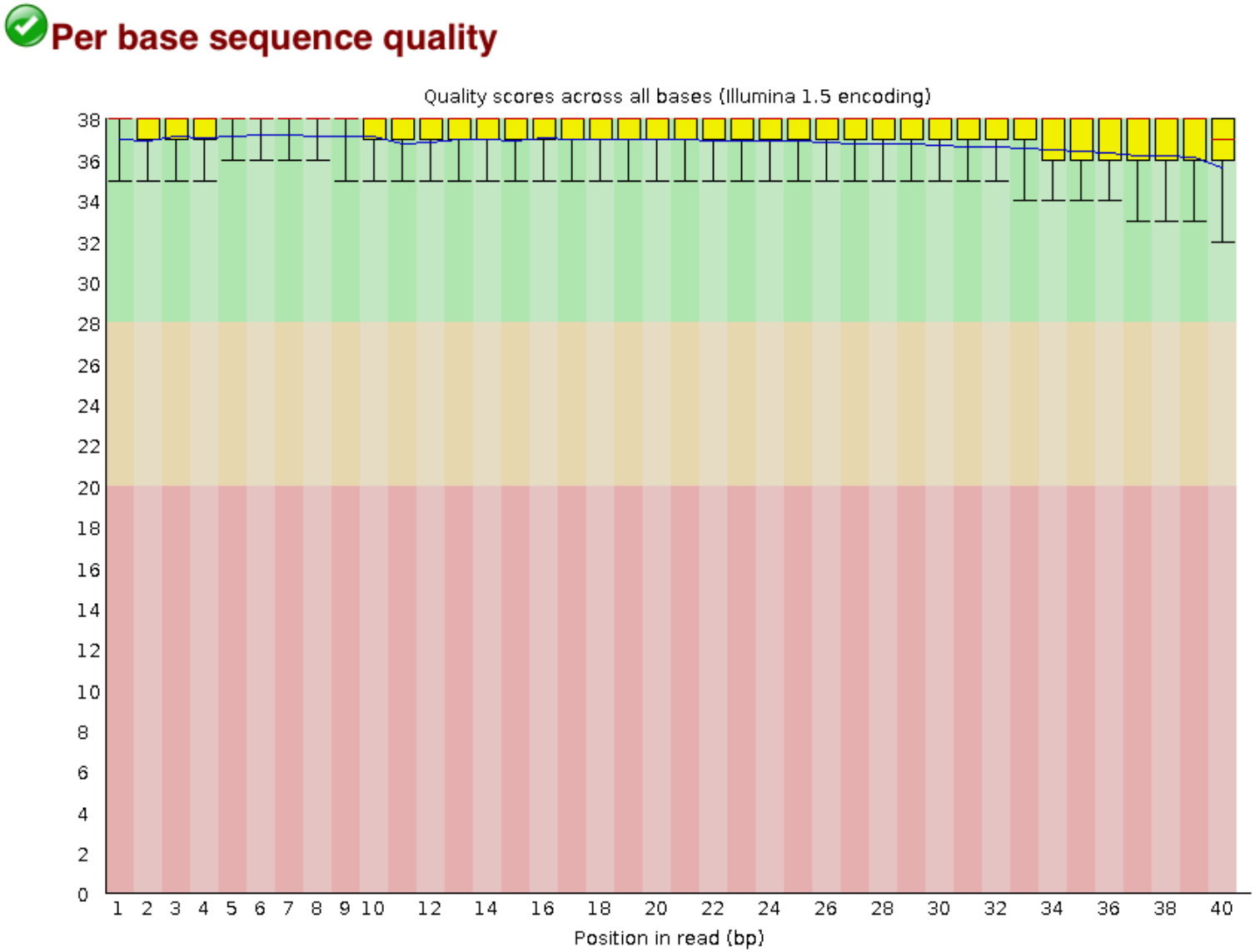


Figure 7

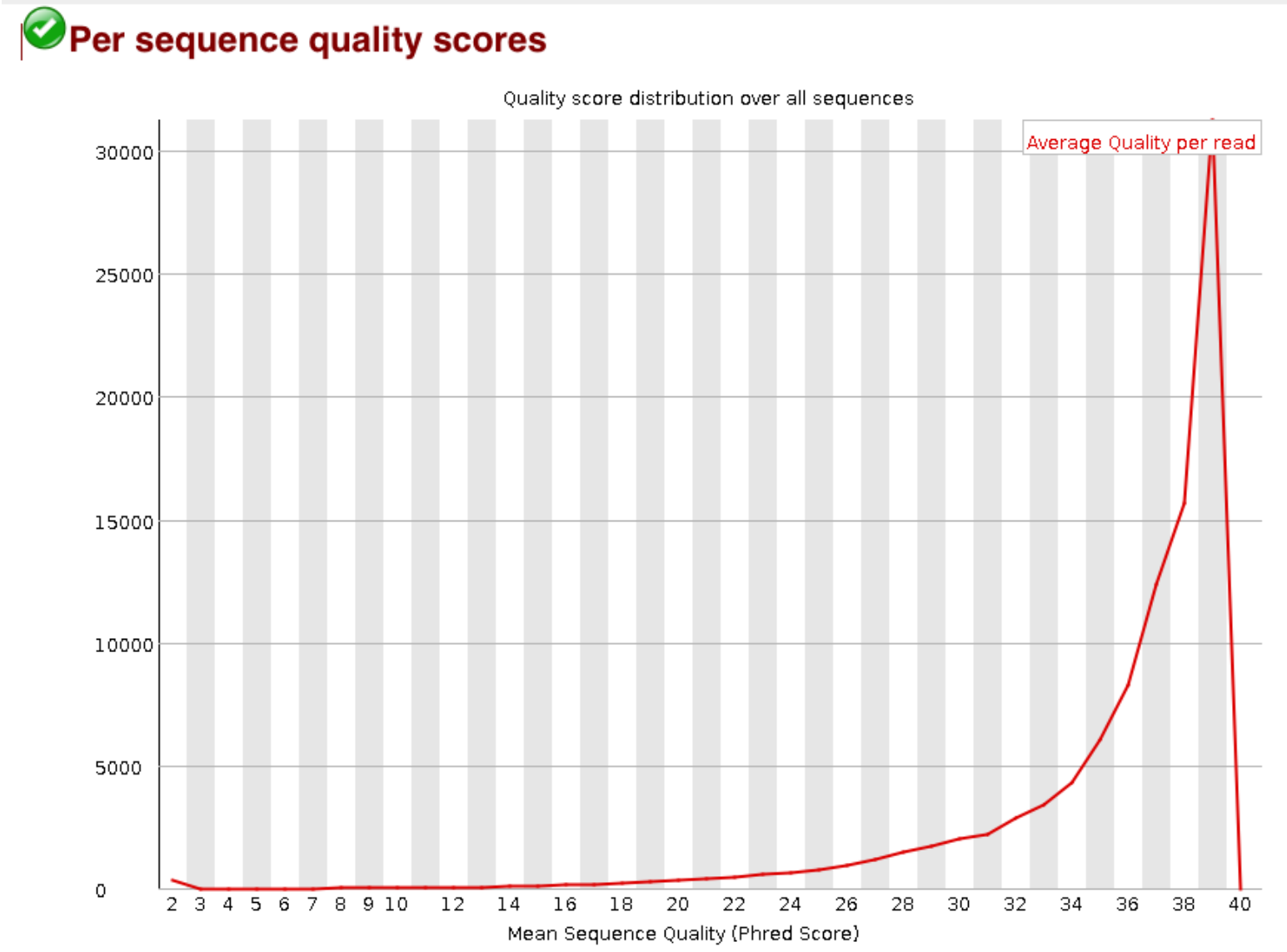
Example sections of a `fastqc` report

Ideally, the plot should look *something* like following:-



Example sections of a `fastqc` report

3. Per-sequence quality scores



Exercise

- using the Galaxy Tool *NGS: QC and manipulation -> FastQC Read Quality reports*
  - generate a QC report for the file `JoeBlogsBRCAPanel_R2.fastq`
- look at the basic statistics for the file
  - does the number of reads agree with your previous answer?
- is there any evidence for a decrease in quality as the read length increases?

The SAM / BAM format

- we don't really spend much time look at *fastq* files
- most of our time is spent with *aligned* reads
  - i.e. we have used some software to tell us whereabouts in the genome each read belongs to
  - we will have a go at this in the practical

The `.sam` file

- **S**equence **A**lignment/**M**ap (sam)
- The output from an aligner such as `bwa` or `bowtie`
- Same format regardless of sequencing protocol (i.e. RNA-seq, ChIP-seq, DNA-seq etc)
- May contain un-mapped reads
- Official specification can be found online http://samtools.github.io/hts-specs/SAMv1.pdf
- We normally work on a compressed version called a `.bam` file. See later.

The .sam file

Comprises a *tab-delimited* section that describes the alignment of each sequence in detail.

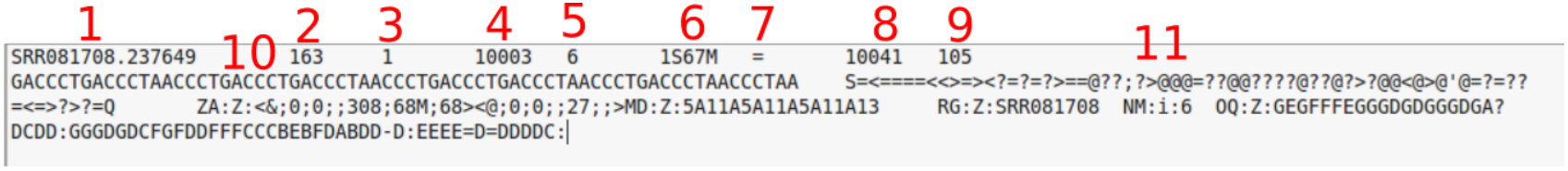


Figure 10

- 1:- Sequence ID
- 2:- Sequence quality expressed as a bitwise *flag*
- 3:- Chromosome that the read aligned to

Fun with flags!

The *"flags"* in the sam file can represent useful QC information

- Read is unmapped
- Read is paired / unpaired
- Read failed QC
- Read is a PCR duplicate (see later)

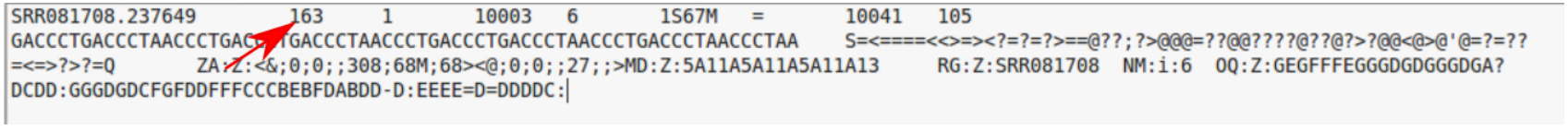


Figure 11

Derivation

	ReadHasProperty	Binary	MultiplyBy
Paired?	TRUE	1	1
Properly Paired?	TRUE	1	2
Unmapped?	FALSE	0	4
Unmapped Mate?	FALSE	0	8
On Minus Strand?	FALSE	0	16
Mate on Minus Strand?	TRUE	1	32
Is First Read?	FALSE	0	64
Is Second Read?	TRUE	1	128
Is Secondary Alignment?	FALSE	0	256
Is Not Passing QC?	FALSE	0	512
Is Duplicate Read?	FALSE	0	1024

$1x1 + 1x2 + 0x4 + 0x8 + 0x16 + 1x32 + 0x64 + 1x128 + 0x256 + 0x512 + 0x1024 = 163$

https://broadinstitute.github.io/picard/explain-flags.html

The .sam file

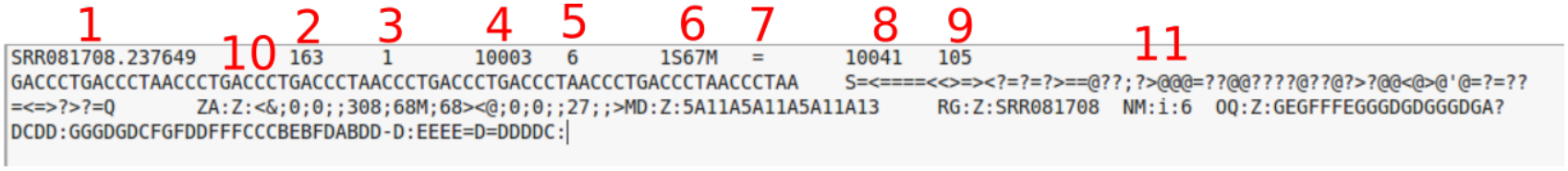


Figure 12

- 4:- Start Position
- 5:- Mapping Quality; Confidence that an alignment is correct
- 6:- CIGAR; Describes number of matches (M), insertions (I), deletions (D)
    - some bases

The .sam file

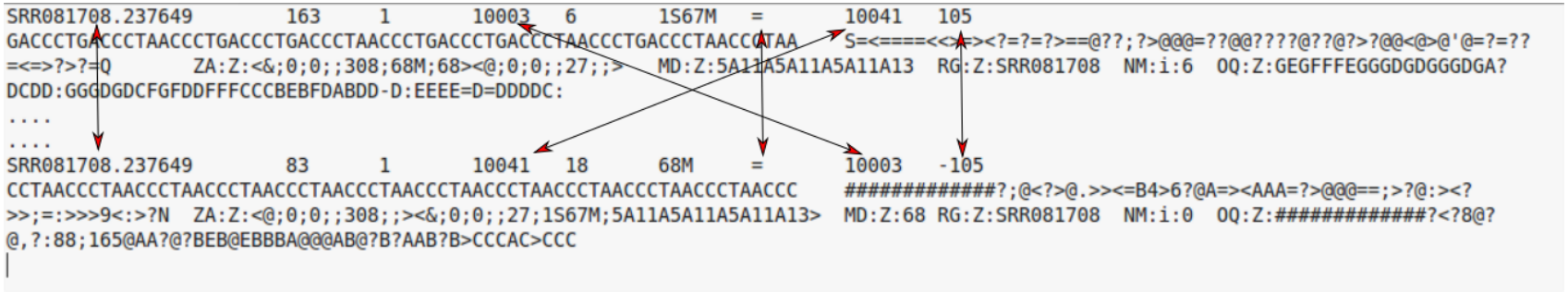


Figure 13

- 7, 8, 9: Alignment information for the paired read
  - whether they align to the same chromosome
  - where the position of the paired read is
  - how far apart did they map?

The .sam file

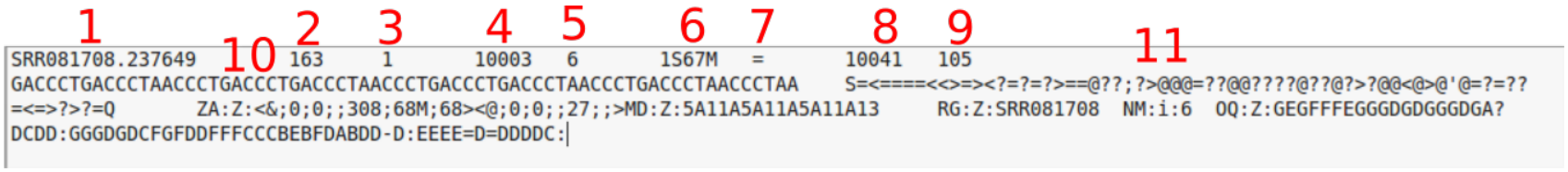


Figure 14

- 10:- Sequence
- 11:- Base Qualities

This is the same as the `fastq` file; so if you have aligned data you can always go back and re-align

Exercise: Alignment

- *NGS: QC and manipulation -> FASTQ Groomer*
  - Select file to groom as `JoeBlogsBRCAPanel_R1.fastq`
- Repeat with `JoeBlogsBRCAPanel_R2.fastq`

Exercise: Alignment

- *NGS: Mapping -> Bowtie2*
  - *Is this single-end or Paired-end?* Select Paired-end
  - Set *FastQ file #1* and *FastQ file #2* to the two files you created in the previous step
  - Press *Execute*