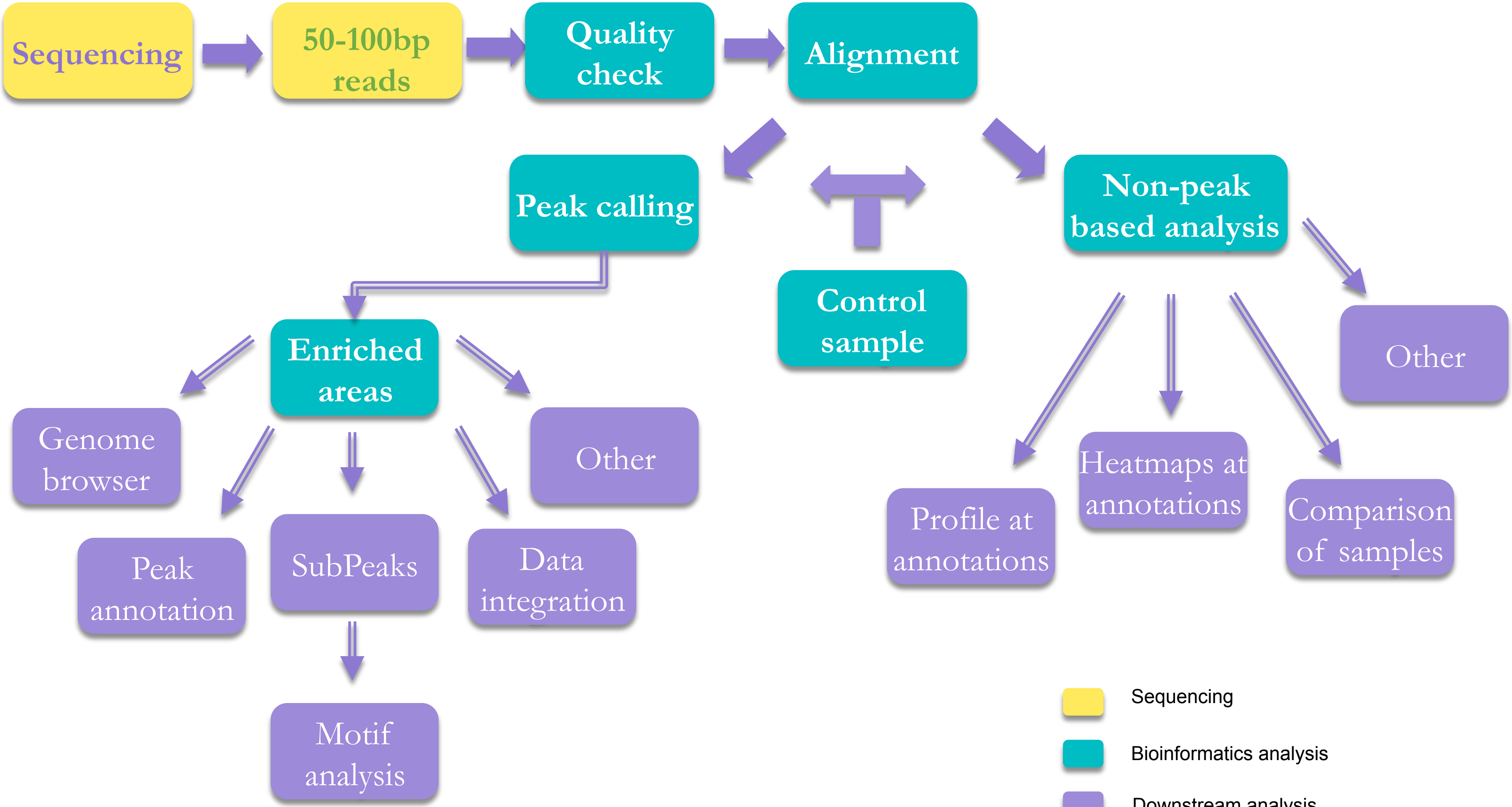
DATA FORMATS AND QUALITY CONTROL

Sandra Cortijo, SLCU (sandra.cortijo@slcu.cam.ac.uk) Sergio Martinez Cuesta (Sergio.MartinezCuesta@cruk.cam.ac.uk)
Sankari Nagarajan (Sankari.Nagarajan@cruk.cam.ac.uk) Ashley Sawle (Ashley.Sawle@cruk.cam.ac.uk)

ChIP-seq ANALYSIS OVERVIEW



Sequencing

50-100bp reads

Quality check

Alignment

Peak calling

Control sample

Non-peak based analysis

Enriched areas

Genome browser

Peak annotation

SubPeaks

Motif analysis

Data integration

Other

Profile at annotations

Heatmaps at annotations

Comparison of samples

Other

🟨 Sequencing

🟦 Bioinformatics analysis

🟪 Downstream analysis

DATA FORMAT

From the sequencer to you

- Sequencing is usually done by core facilities
- Each sequencing run will generate millions of short (~100 bp) reads
  - + read quality score for each base
- They often perform initial processing
  - Adaptor trimming
  - Basic quality control
  - Demultiplexing
- You will (usually) receive a FASTQ file

FASTQ Files

- FASTQ = FASTA + Quality

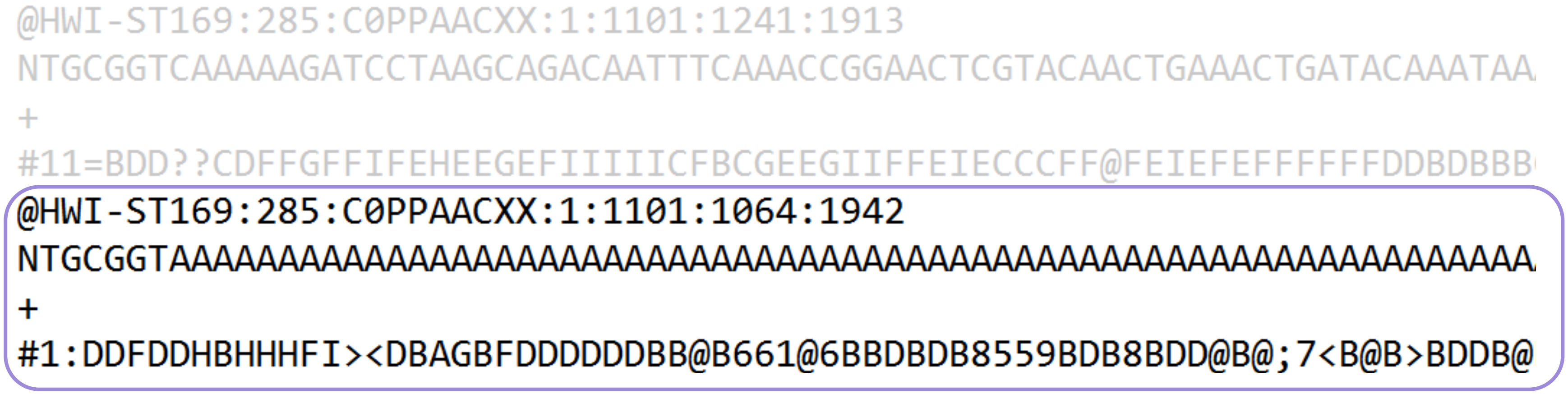
- So what is FASTA?

FASTA Format

```
>CHROMOSOME_1
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTC
AACTCACAGTTTGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATT
…
>CHROMOSOME_2
TCACAGTTTGGTTCAAAGCAGTATCGATCATATCGATCAAATAGTAAA
…
```

- Format for raw DNA sequences
- For each DNA sequence:
  1. > NAME
  2. Nucleotides, with line breaks every ~60 bp

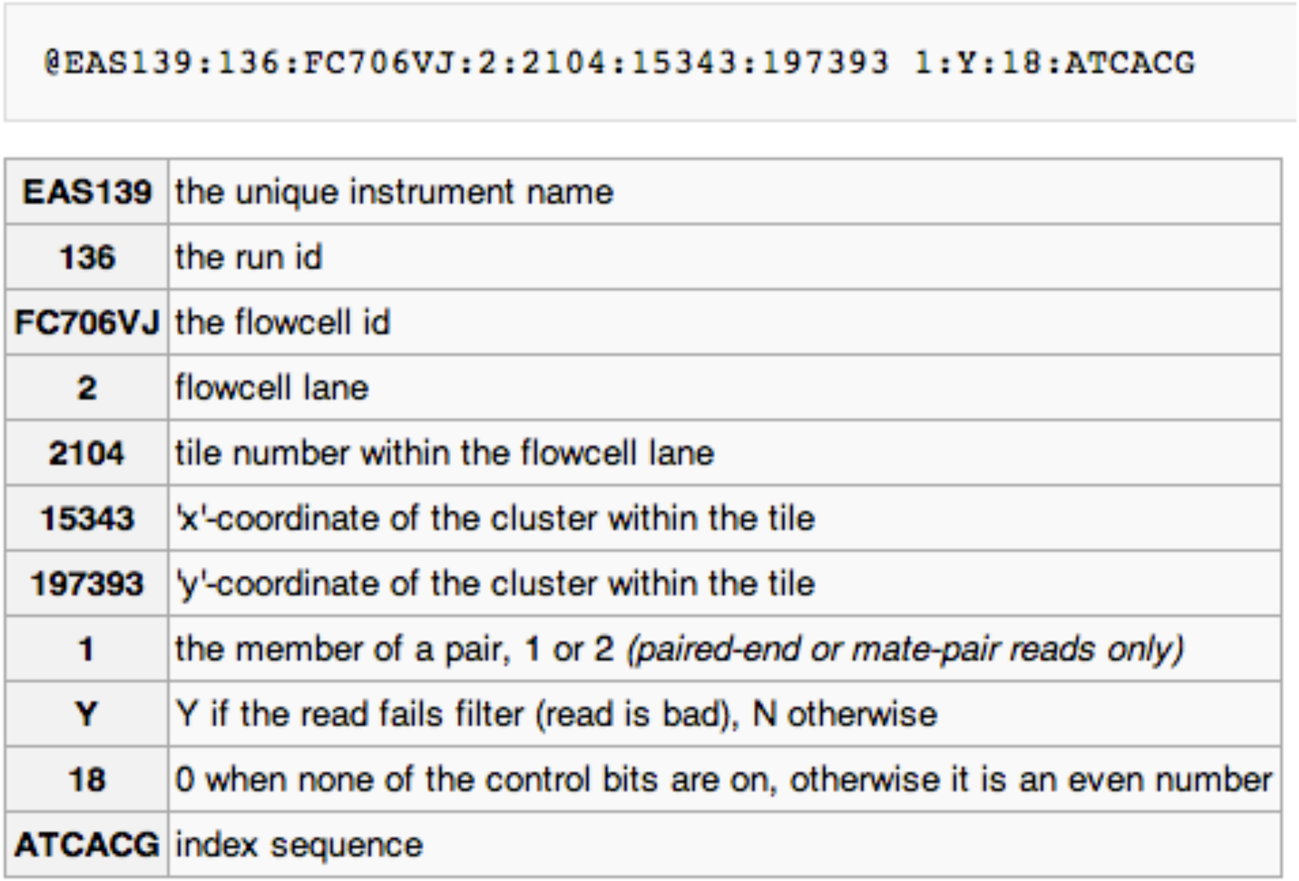
FASTQ format



```
@HWI-ST169:285:C0PPAACXX:1:1101:1241:1913
NTGCGGTCAAAAAGATCCTAAGCAGACAATTTCAAACCGGAACTCGTACAACTGAAACTGATACAAATAA
+
#11=BDD??CDFFGFFIFEHEEGEFIIIIICFBCGEEGIIFFEIECCCFF@FEIEFEFFFFFFDDBDBBB
@HWI-ST169:285:C0PPAACXX:1:1101:1064:1942
NTGCGGTAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
+
#1:DDFDDHBHHHFI><DBAGBFDDDDDDBB@B661@6BBDBDB8559BDB8BDD@B@;7<B@B>BDDB@
```

- Format for DNA sequencing reads
- For each read:
  1. @ Read ID
  2. Nucleotide sequence of the read
  3. +
  4. Quality score for each nucleotide of the read

Illumina sequence identifiers



```
@EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG
```

EAS139	the unique instrument name
136	the run id
FC706VJ	the flowcell id
2	flowcell lane
2104	tile number within the flowcell lane
15343	'x'-coordinate of the cluster within the tile
197393	'y'-coordinate of the cluster within the tile
1	the member of a pair, 1 or 2 *(paired-end or mate-pair reads only)*
Y	Y if the read fails filter (read is bad), N otherwise
18	0 when none of the control bits are on, otherwise it is an even number
ATCACG	index sequence

Quality Scores

- P = probability that the base call is wrong

$$Q_{\text{Sanger}} = -10\log_{10} p$$

  - p = 0.1 → Q = 10
  - p = 0.01 → Q = 20
  - P = 0.001 → Q = 30
- Encoding:

Sanger/Phred format can encode a quality score from 0 to 93 using ASCII 33 to 126:

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
|                                                                                            |
33                                                                                         126
```

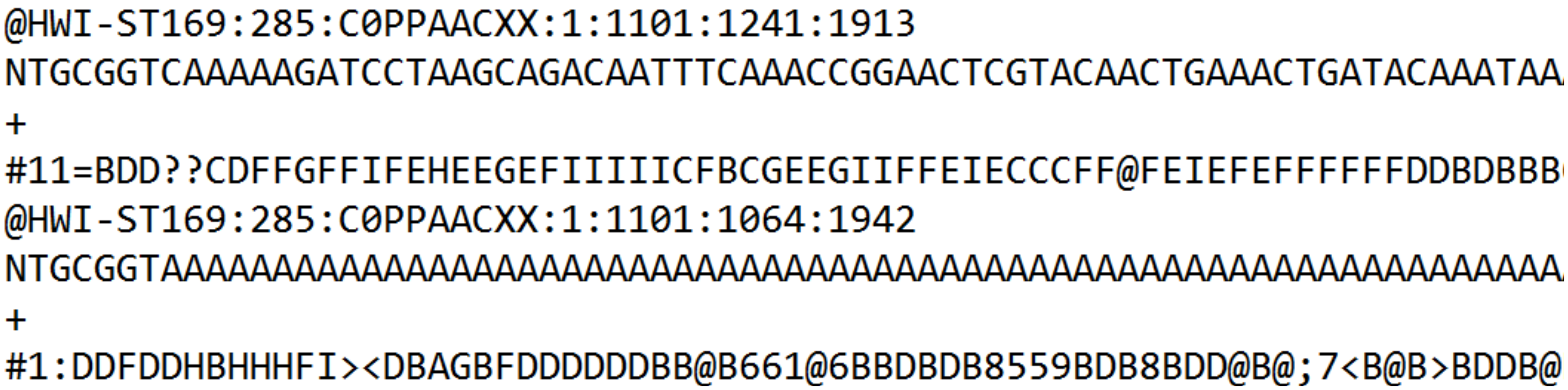
Quality Score Encoding

ASCII Code
Quality Score

```
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHI
 |                         |    |        |
33                        59   64       73
 0........................26...31.......40
```

- Each character has an associated ASCII Code
- ASCII Code – Offset = Quality Score
- Normal Sanger Encoding is Phred + 33
  - Lowest: "!" = ASCII 33 = Quality 0
  - Highest: "I" = ASCII 73 = Quality 40

Quality Encoding Example

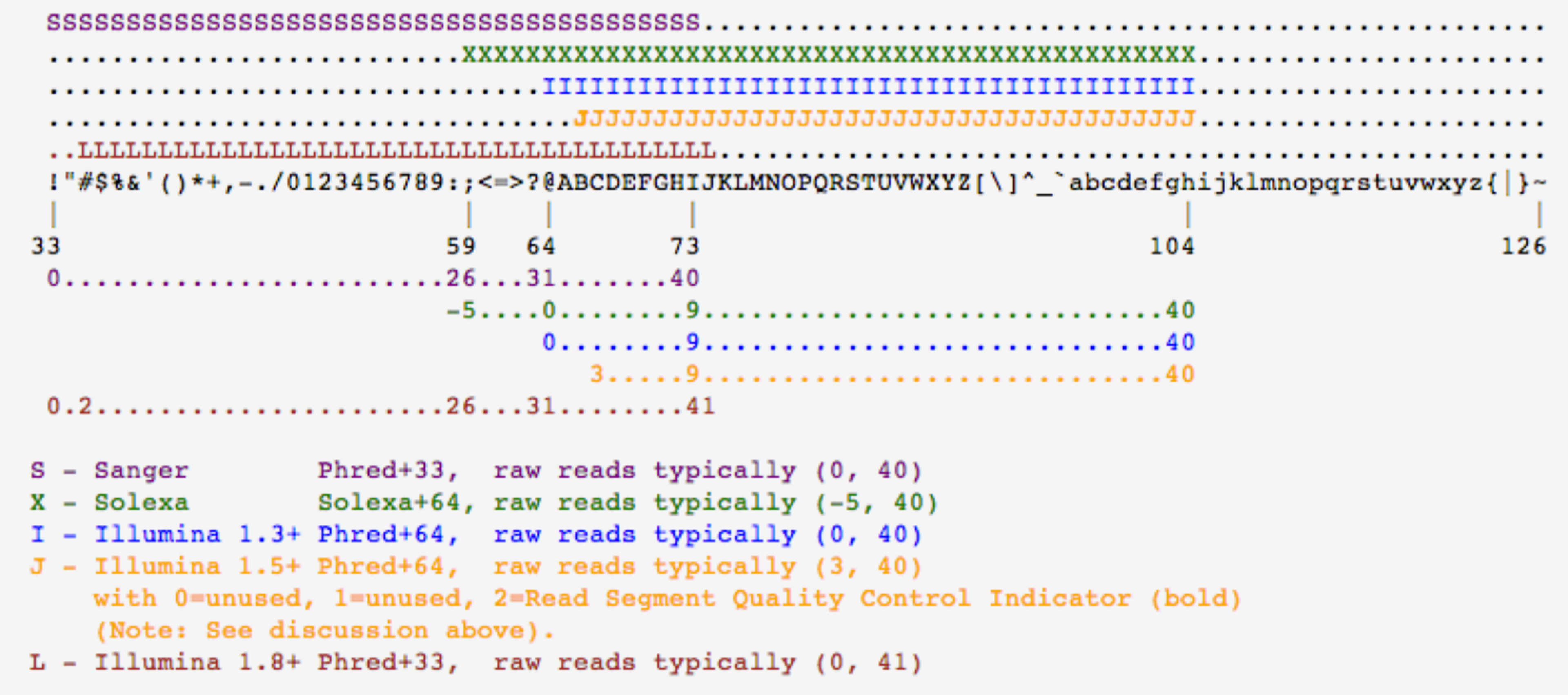


```
@HWI-ST169:285:C0PPAACXX:1:1101:1241:1913
NTGCGGTCAAAAAGATCCTAAGCAGACAATTTCAAACCGGAACTCGTACAACTGAAACTGATACAAATAA
+
#11=BDD??CDFFGFFIFEHEEGEFIIIIICFBCGEEGIIFFEIECCCFF@FEIEFEFFFFFFDDBDBBB
@HWI-ST169:285:C0PPAACXX:1:1101:1064:1942
NTGCGGTAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
+
#1:DDFDDHBHHHFI><DBAGBFDDDDDDBB@B661@6BBDBDB8559BDB8BDD@B@;7<B@B>BDDB@
```

```
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHI
 |      |    |    |    |   |    |     |  |
33     40   45   50   55  59   64    70 73
 0......7.........17........26...31........40
```

Different Quality Encodings

- Beware of different versions! (especially for old data)

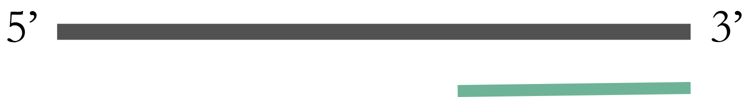


```
  SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
  ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......................
  ...............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII......................
  .................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ......................
  ..LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL....................................................
  !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
  |                         |    |        |                              |                     |
 33                        59   64       73                            104                   126
  0........................26...31.......40
                           -5....0........9.............................40
                                 0........9.............................40
                                    3.....9.............................40
  0.2......................26...31........41

S - Sanger        Phred+33,  raw reads typically (0, 40)
X - Solexa        Solexa+64, raw reads typically (-5, 40)
I - Illumina 1.3+ Phred+64,  raw reads typically (0, 40)
J - Illumina 1.5+ Phred+64,  raw reads typically (3, 40)
    with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
    (Note: See discussion above).
L - Illumina 1.8+ Phred+33,  raw reads typically (0, 41)
```

Single end vs paired-end

Single end sequencing

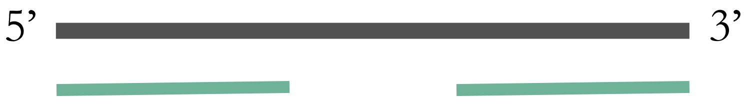


my_sequence.fastq

```
@HWI-BRUNOP16X_0001:1:1:1466:1018#0/1
AAGGAAGTGCTTGTCTGGCTAACACAGCNAGNCACGTGAC
+
aVfbe`^^^_TTTSSdffffdfffabbZbbfebafbbbbb
```

SE

Paired end sequencing



my_sequence_1.fastq

```
@HWI-BRUNOP16X_0001:1:1:1278:989#0/1
NAAATTTCGAATTTCTGTGAAGTAAGCATCTTCTTTGTCAT
+
BJJGGKIINN^^^^^QQNTUQOOTTTRTOTY^^Y^\\^^^\
```

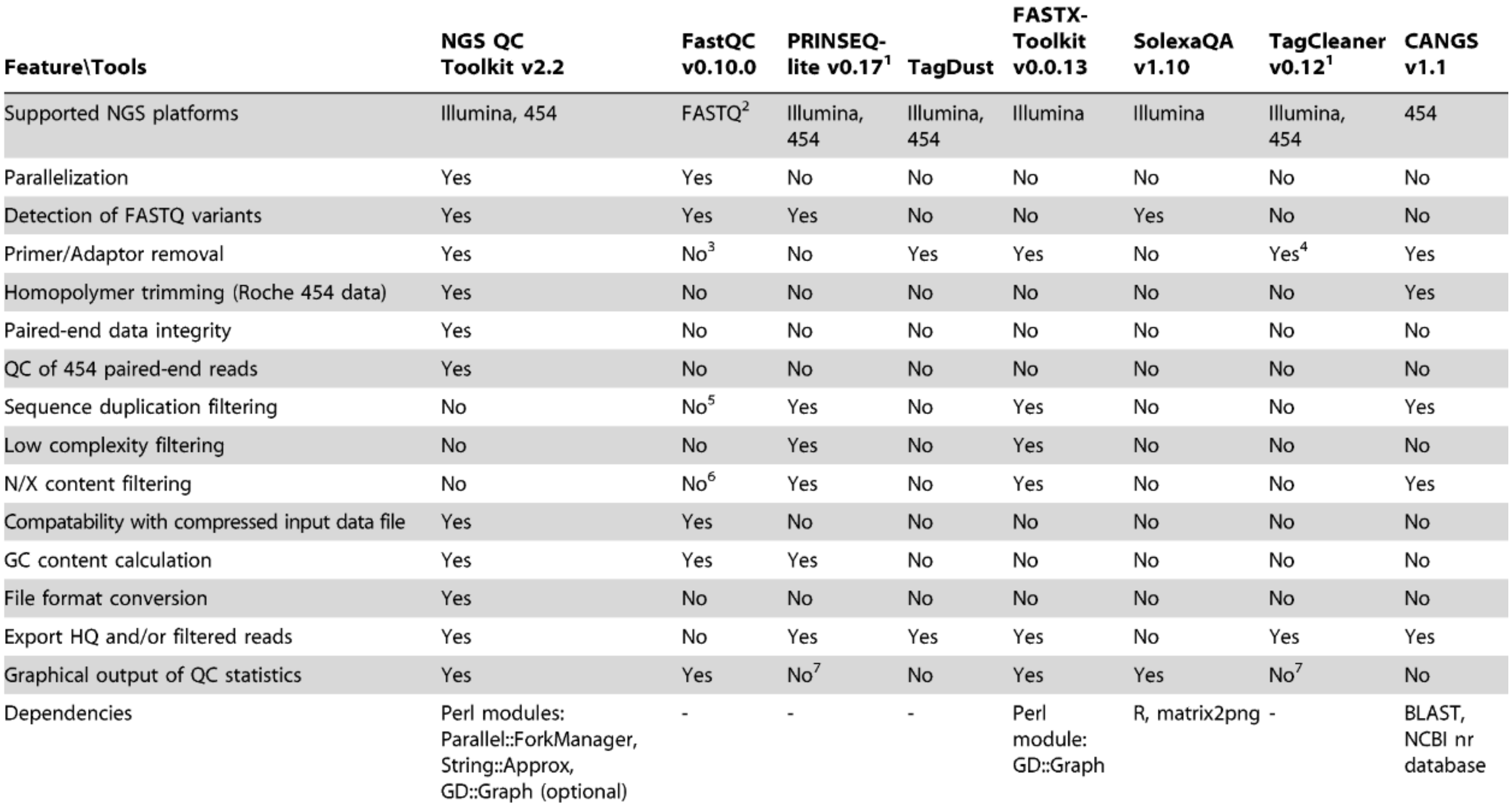
my_sequence_2.fastq

```
@HWI-BRUNOP16X_0001:1:1:1278:989#0/2
AACCCACACAGGAGAGCAGCCTTACAGATGCAAATACTGTG
+
]K___fffffggghegggggdgggggfgggggegggghh
```

PE

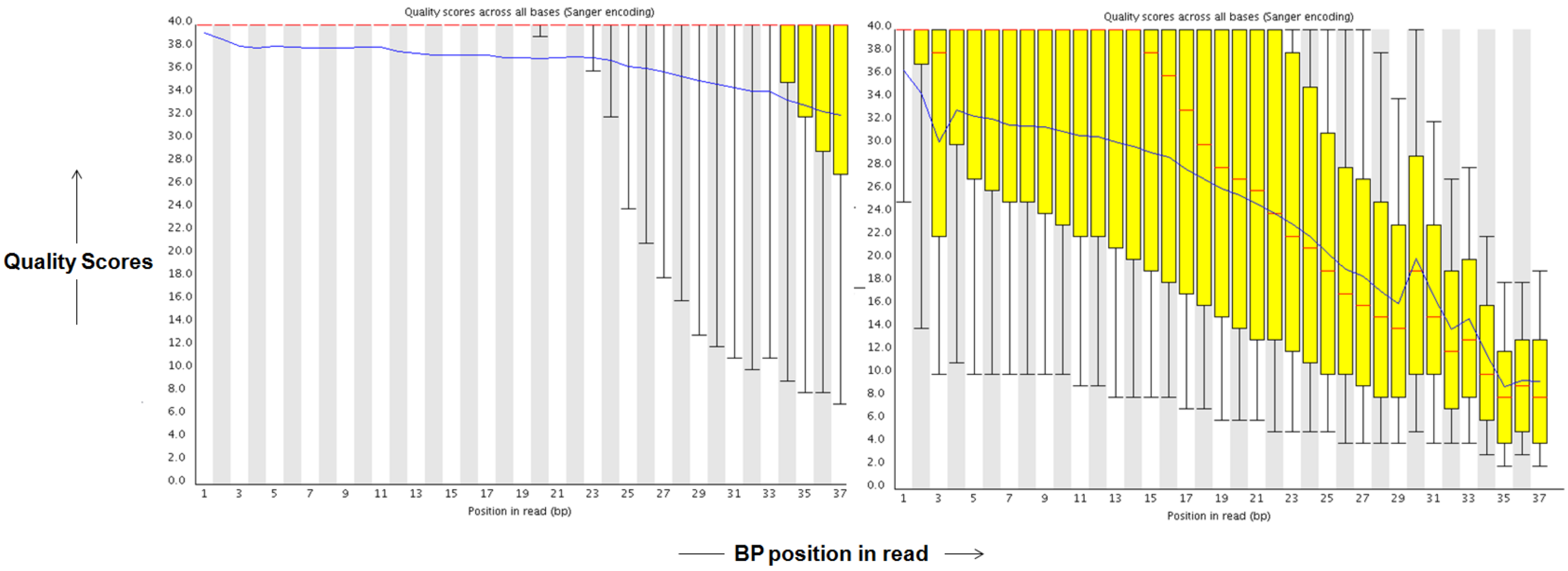
QUALITY CHECK

Comparison of various features across available QC tools



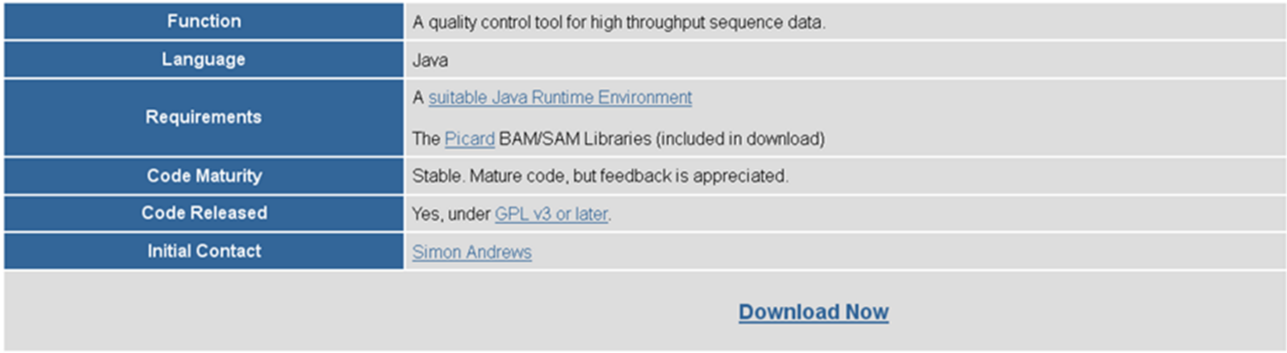
Feature\Tools	NGS QC Toolkit v2.2	FastQC v0.10.0	PRINSEQ-lite v0.17[1]	TagDust	FASTX-Toolkit v0.0.13	SolexaQA v1.10	TagCleaner v0.12[1]	CANGS v1.1
Supported NGS platforms	Illumina, 454	FASTQ[2]	Illumina, 454	Illumina, 454	Illumina	Illumina	Illumina, 454	454
Parallelization	Yes	Yes	No	No	No	No	No	No
Detection of FASTQ variants	Yes	Yes	Yes	No	No	Yes	No	No
Primer/Adaptor removal	Yes	No[3]	No	Yes	Yes	No	Yes[4]	Yes
Homopolymer trimming (Roche 454 data)	Yes	No	No	No	No	No	No	Yes
Paired-end data integrity	Yes	No	No	No	No	No	No	No
QC of 454 paired-end reads	Yes	No	No	No	No	No	No	No
Sequence duplication filtering	No	No[5]	Yes	No	Yes	No	No	Yes
Low complexity filtering	No	No	Yes	No	Yes	No	No	No
N/X content filtering	No	No[6]	Yes	No	Yes	No	No	Yes
Compatability with compressed input data file	Yes	Yes	No	No	No	No	No	No
GC content calculation	Yes	Yes	Yes	No	No	No	No	No
File format conversion	Yes	No	No	No	No	No	No	No
Export HQ and/or filtered reads	Yes	No	Yes	Yes	Yes	No	Yes	Yes
Graphical output of QC statistics	Yes	Yes	No[7]	No	Yes	Yes	No[7]	No
Dependencies	Perl modules: Parallel::ForkManager, String::Approx, GD::Graph (optional)	-	-	-	Perl module: GD::Graph	R, matrix2png	-	BLAST, NCBI nr database

FastQC: Per base sequence quality



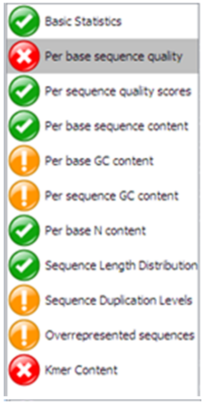
Quality Scores

BP position in read



Function	A quality control tool for high throughput sequence data.
Language	Java
Requirements	A suitable Java Runtime Environment The Picard BAM/SAM Libraries (included in download)
Code Maturity	Stable. Mature code, but feedback is appreciated.
Code Released	Yes, under GPL v3 or later.
Initial Contact	Simon Andrews

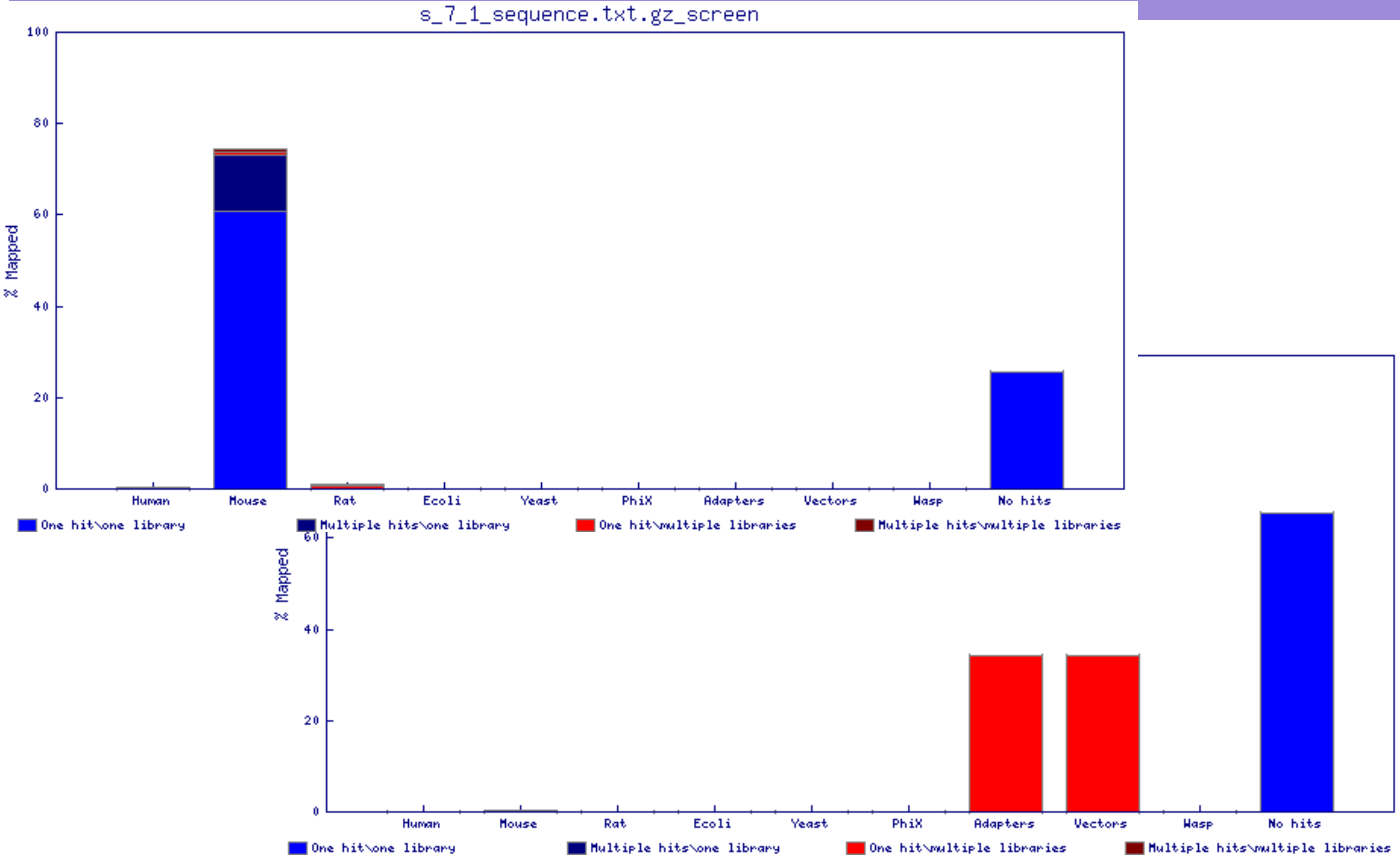
Download Now



- ✓ Basic Statistics
- ✗ Per base sequence quality
- ✓ Per sequence quality scores
- ✓ Per base sequence content
- ! Per base GC content
- ! Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ! Sequence Duplication Levels
- ! Overrepresented sequences
- ✗ Kmer Content

FastQC: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

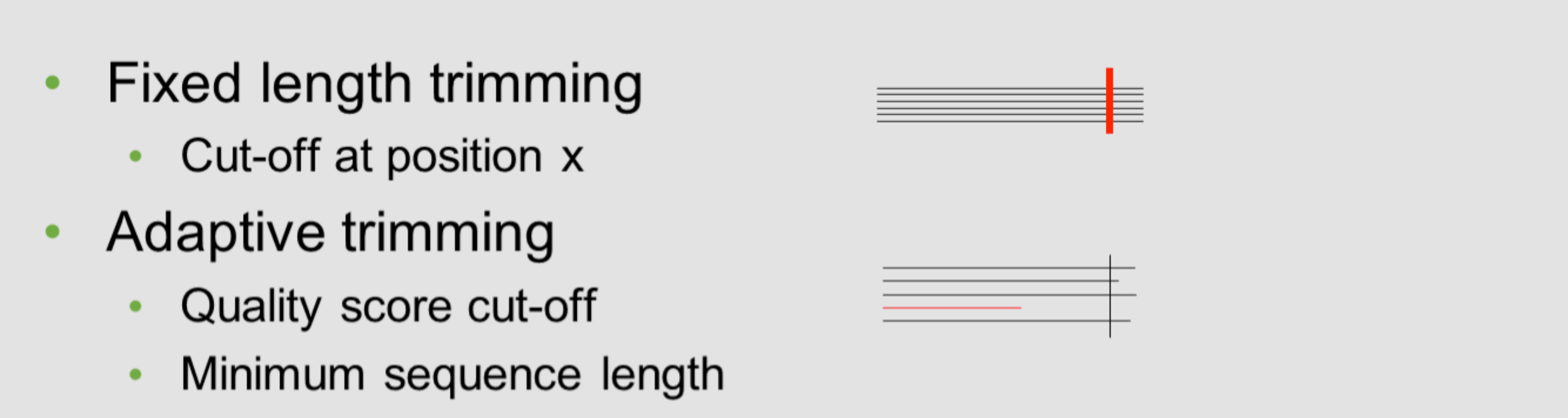
FastqScreen: contamination



Common sequence artefacts in NGS data

- Read errors
  - Base calling errors
  - Small insertions and deletions
- Poor quality reads
- Primer / adapter contamination

Quality trimming



- Fixed length trimming
  - Cut-off at position x
- Adaptive trimming
  - Quality score cut-off
  - Minimum sequence length

Filtering

- How?
  - Fastx (*http://hannonlab.cshl.edu/fastx_toolkit/*)
  - PRINSEQ (*http://prinseq.sourceforge.net/*)
  - Tally and Reaper:

    http://www.ebi.ac.uk/~stijn/reaper/tally.html

    http://www.ebi.ac.uk/~stijn/reaper/reaper.html#recipe

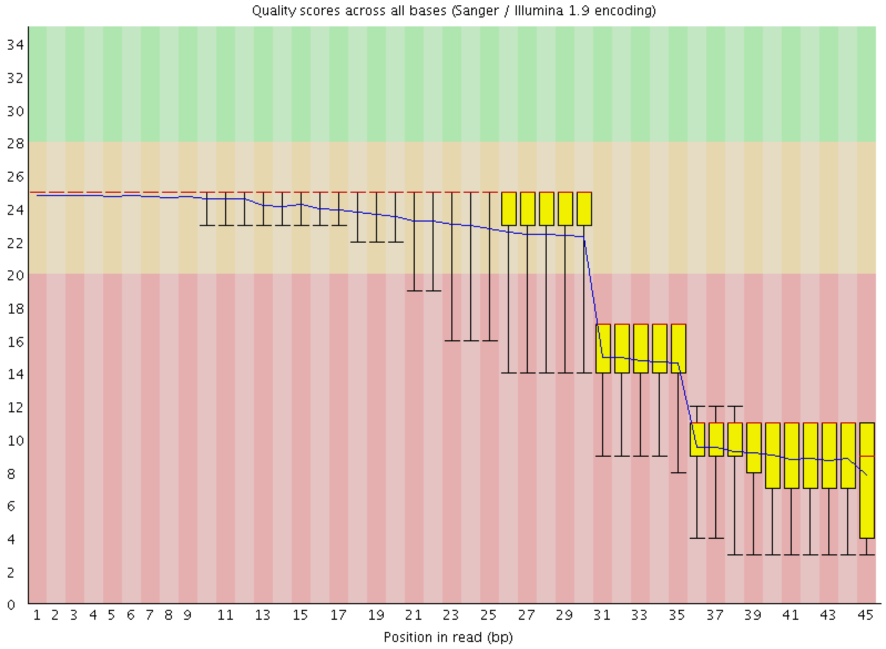
    http://www.ebi.ac.uk/~stijn/reaper/src/reaper-12-048/
  - ShortRead (R) (*http://www.bioconductor.org/packages/release/bioc/html/ShortRead.html*)

FASTQ Processing – FASTX Toolkit

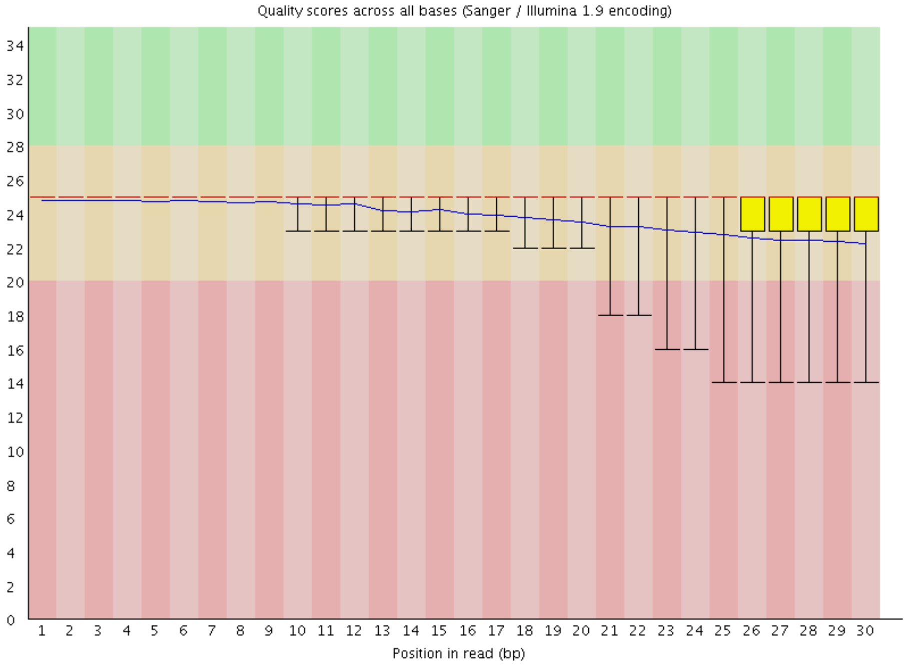
- http://hannonlab.cshl.edu/fastx_toolkit/
- Many tools for common operations on FASTQ files:
  - Conversion
  - Trimming (remove barcodes)
  - Clipping (remove adapters)
  - Quality trimmer (trim off low-quality bases)
  - Quality filter (remove low-quality reads)

Filtering example

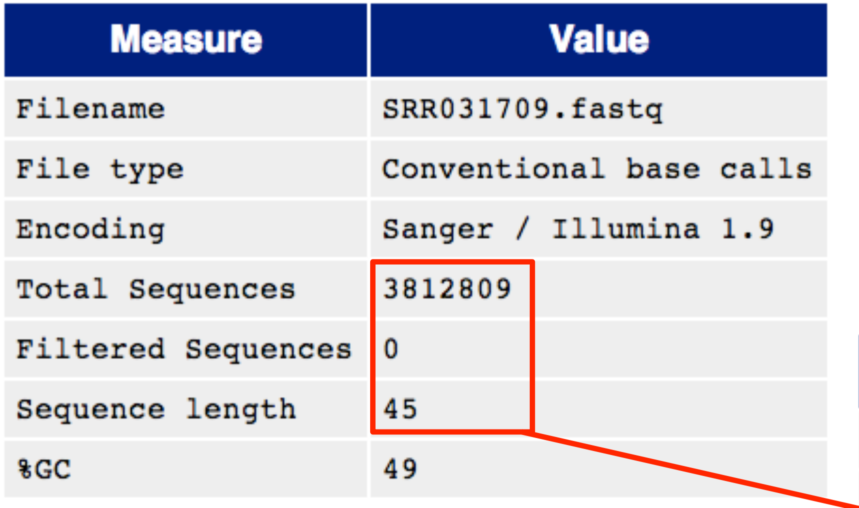
Before



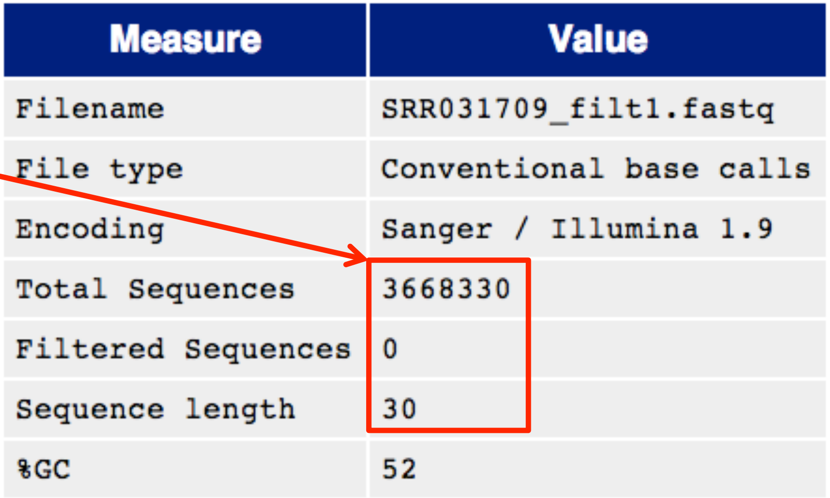
After



Filtering comes at a price



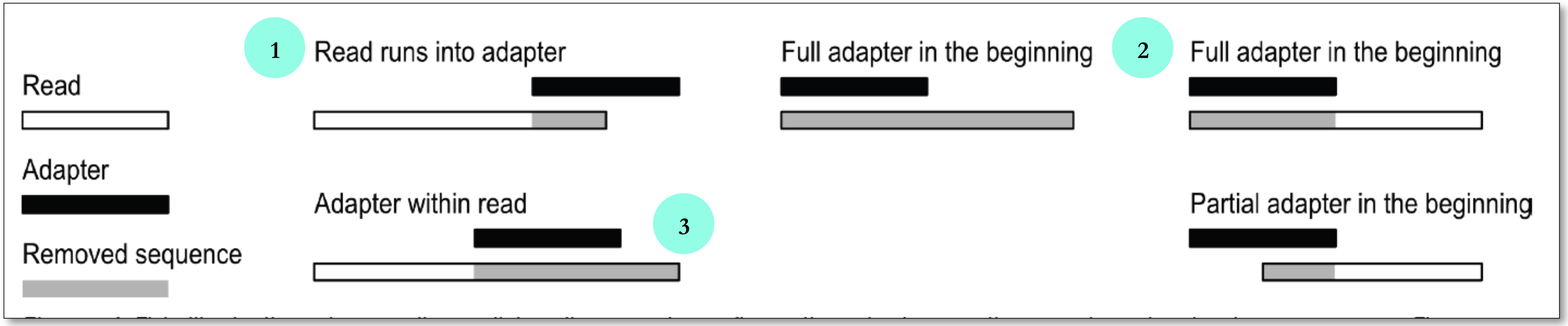
Measure	Value
Filename	SRR031709.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	3812809
Filtered Sequences	0
Sequence length	45
%GC	49



Measure	Value
Filename	SRR031709_filt1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	3668330
Filtered Sequences	0
Sequence length	30
%GC	52

Removal of adapter sequences

- Necessary when the read length > molecule sequenced e.g. small RNAs.


Read
Adapter
Removed sequence
1
Read runs into adapter
Adapter within read
3
Full adapter in the beginning
2
Full adapter in the beginning
Partial adapter in the beginning

- Different scenarios requiring adapter removal
  1. Trim the 3' end
  2. Trim/discard the reads based on the residual minimum read length.
  3. Trim the adapter region but retain reads only with a minimum read-length.
- Tools for adapter trimming
  - fastx_clipper (FastX-Toolkit), PRINSEQ

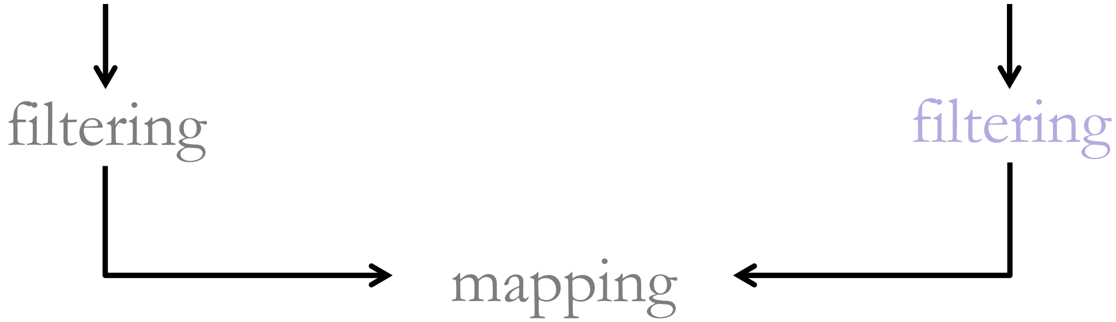
Important: PE files

my_sequence_1.fastq

```
@HWI-BRUNOP16X_0001:1:1:1278:989#0/1
NAAATTTCGAATTTCTGTGAAGTAAGCATCTTCTTTGTCAT
+
BJJGGKIINN^^^^^QQNTUQOOTTTRTOTY^^Y^\\^^^\
```

my_sequence_2.fastq

```
@HWI-BRUNOP16X_0001:1:1:1278:989#0/2
AACCCACACAGGAGAGCAGCCTTACAGATGCAAATACTGTG
+
]K___fffffggghegggggggdgggggfgggggegggghh
```



filtering

filtering

mapping

How?

- Trim Galore! (http://www.bioinformatics.babraham.ac.uk/projects/trim_galore/)

Conclusions

- Quality control of sequencing data is essential for downstream analysis.

- A range of QC tools are available to remove noise

- Decide on which data can be corrected and discard the rest.