Lesson 3 - Genomics

NGS data quality control and pre-processing

The `find` command

- Useful for searching for files within a directory

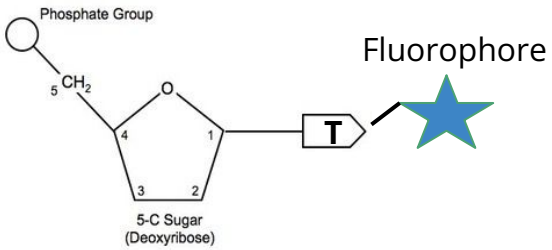
```
# show all files under a directory
$ find dir/
# search for a file with exact name
$ find dir -name ‘my_file.txt’
# search for files with name like
$ find dir -name ‘my_file.*’
$ find dir -name ‘*.txt’
# search for files modified in the last x days
$ find dir/ -mtime -5
```

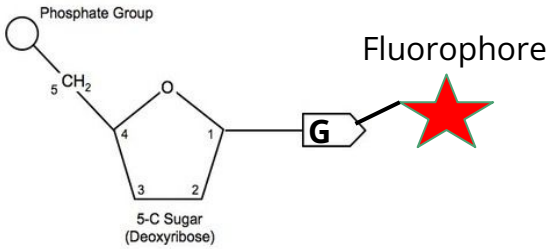
By the end of this lesson you will...

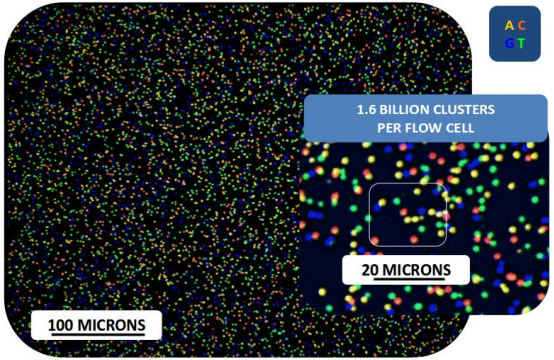
- Be familiar with some common problems with NGS data and how to handle them
- Understand the fastq and fasta file formats
- Be able to QC raw sequencing data
- Know when and how to use various tools for NGS data preprocessing

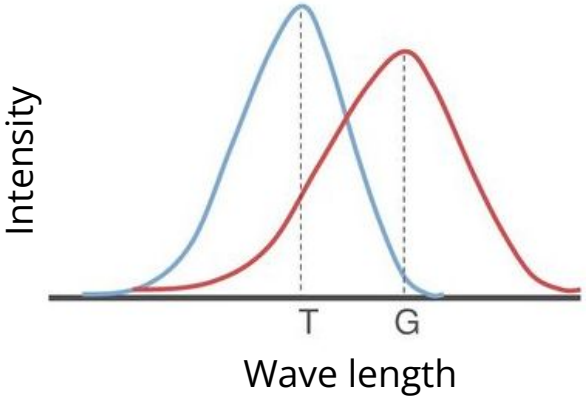
Low quality base calling

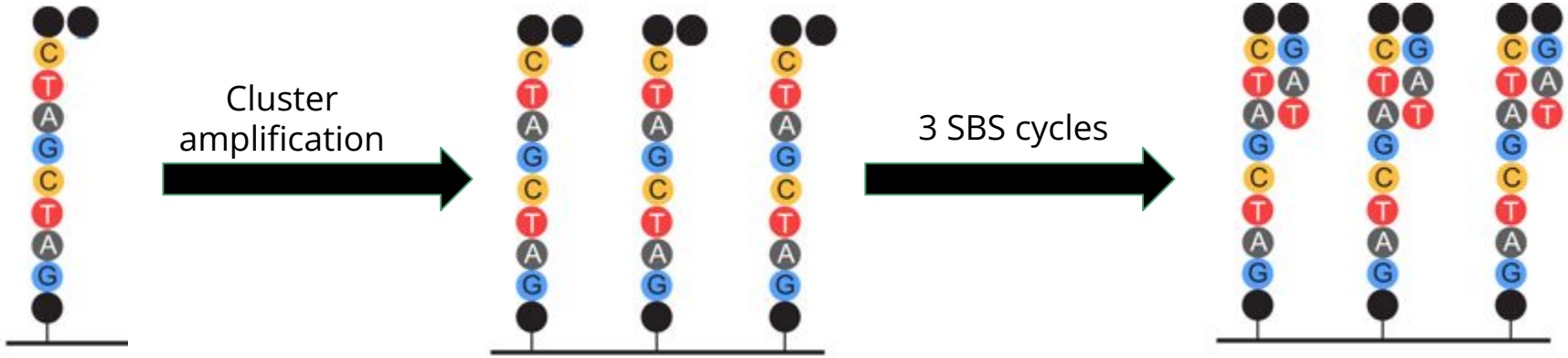
- Illumina machines make mistakes!
- Rate of errors: ~1/1000
- Reasons:
  - Color cross-talk
  - Clusters cross-talk
  - Phasing



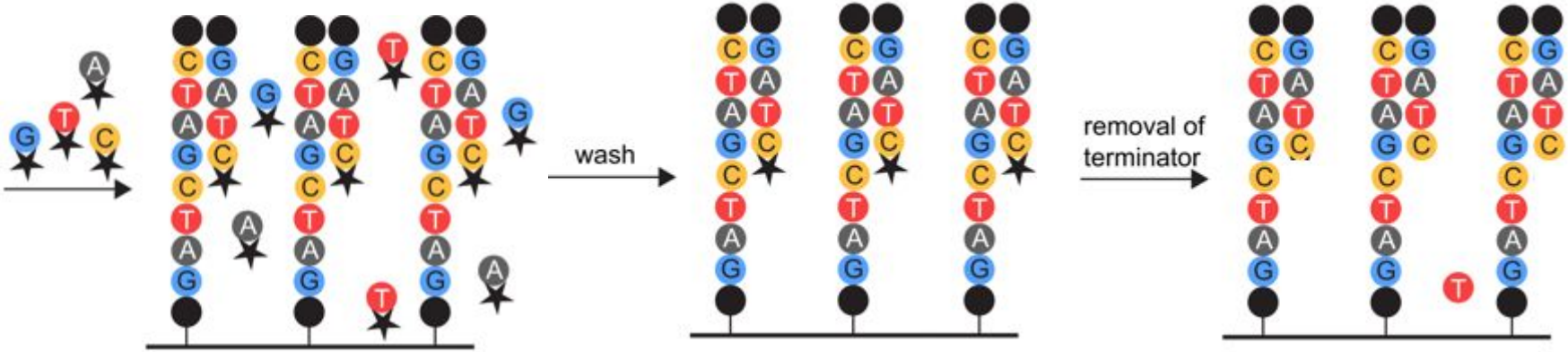


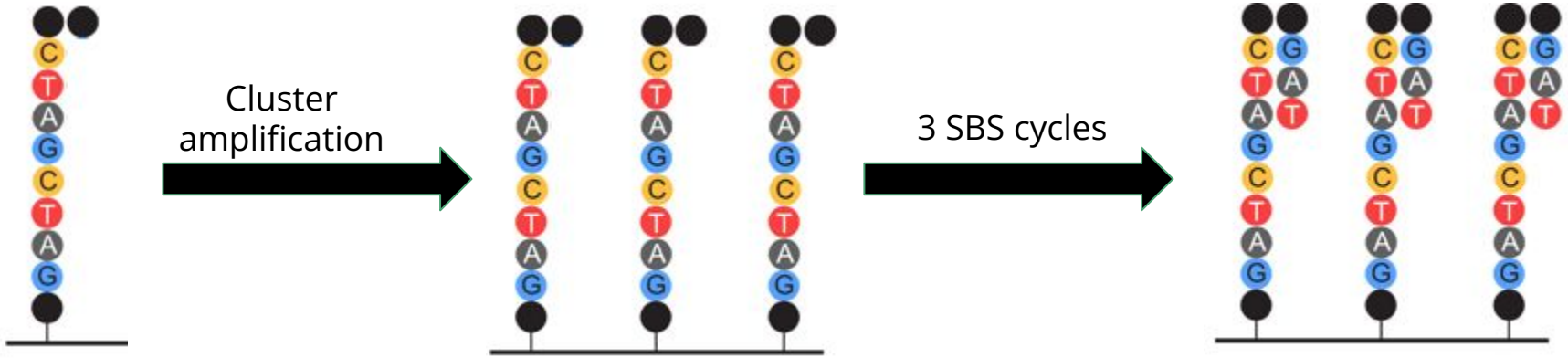




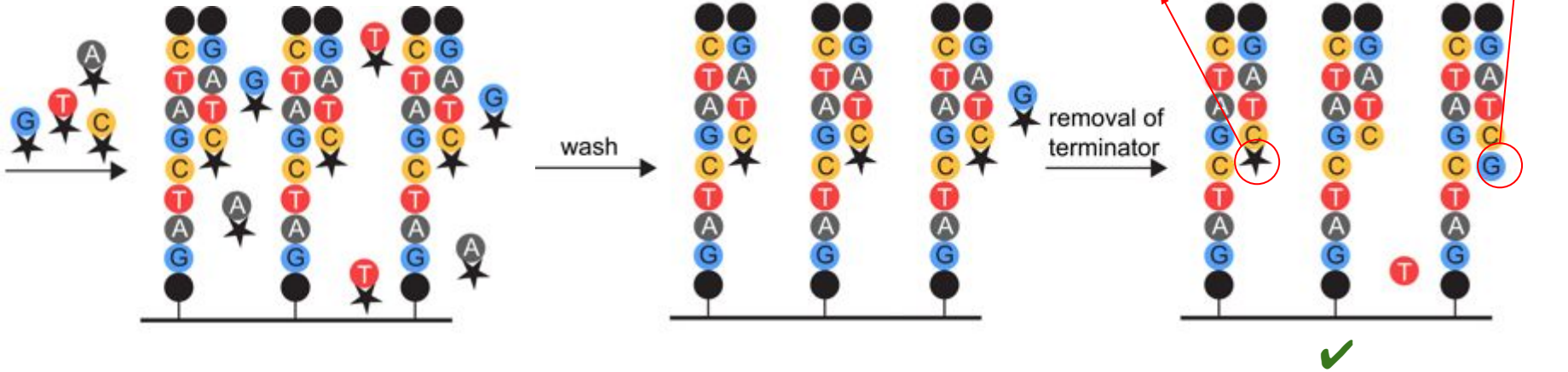


4th sequencing cycle

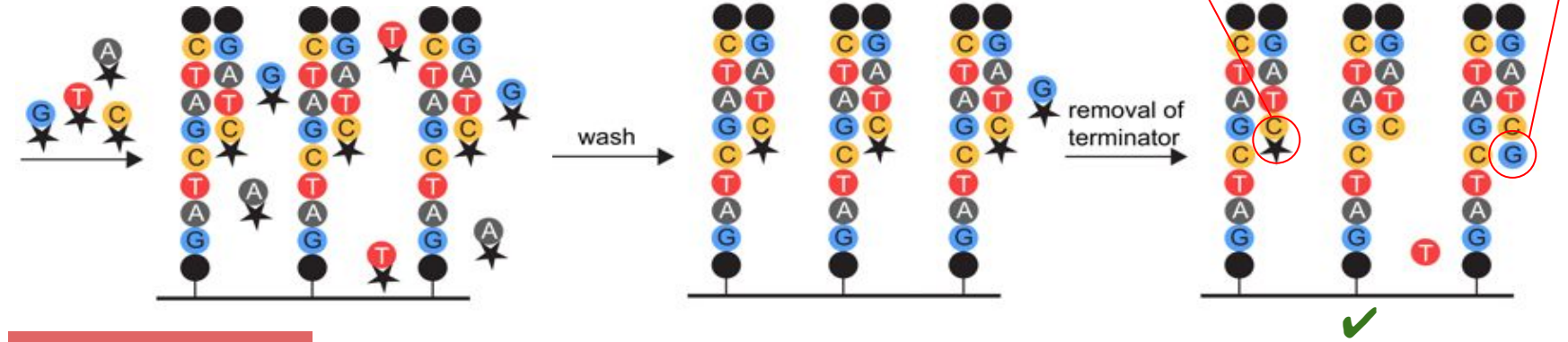




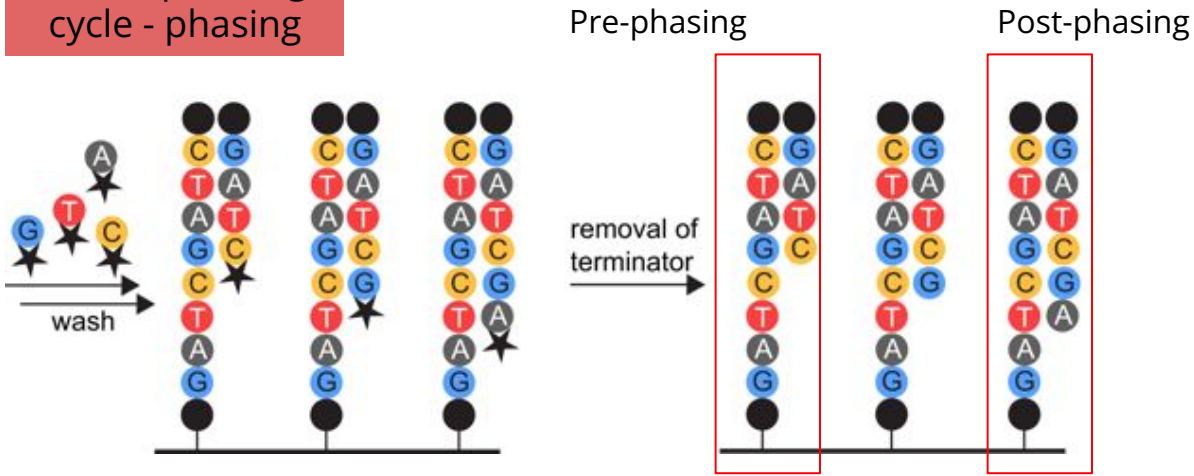
4th sequencing cycle - phasing



4th sequencing cycle - phasing



5th sequencing cycle - phasing



Low or biased yield

- Too few reads

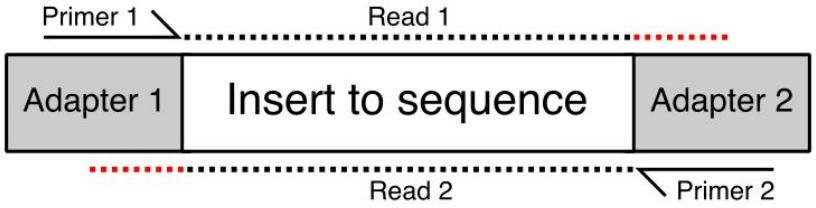
  Usually caused by non-optimal cluster density

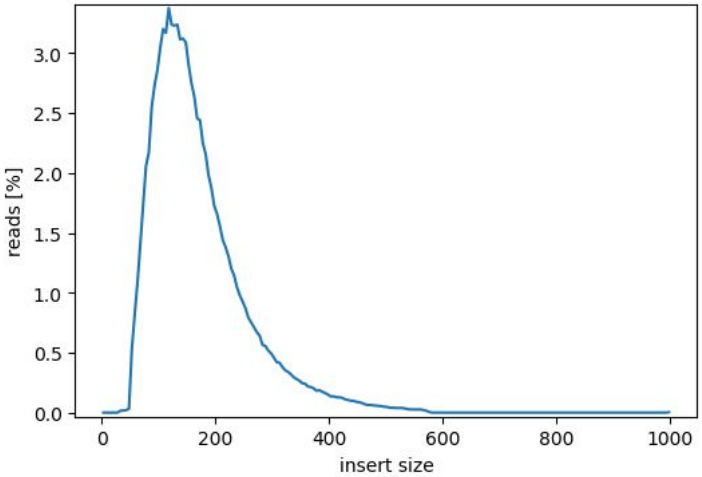
- Some genomic fragments sequenced more often than others

  Usually caused by library prep issues or PCR duplication

Technical contamination

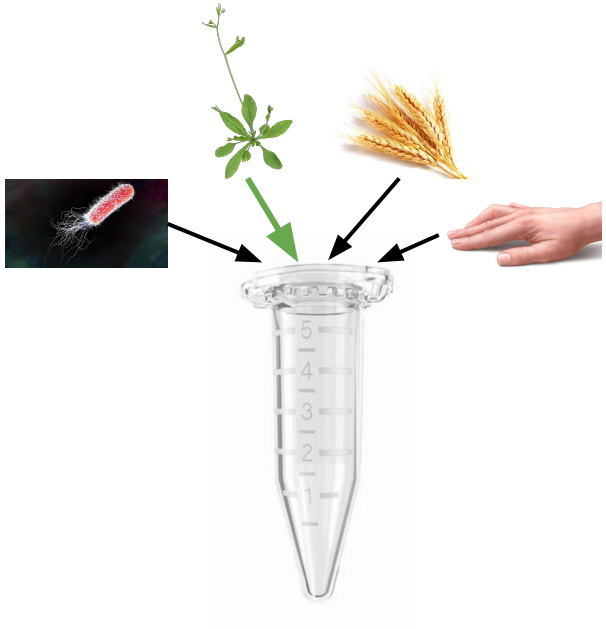
Technical contamination - adaptors “read-through”





Biological contamination

- Contamination with other species
  - Bacteria
  - Human
  - Other species
- Mixed samples



Quality control using PhiX spike-in

- PhiX is a bacteriophage
  - Snall - ~5,000 bp
  - Diverse nucleotides
  - Balanced GC content
- Add a small spike-in of PhiX to your run for quality control:
  - Estimate cross-talk and phasing rates
  - Positive control
  - Determine which step/s went wrong

The Fastq format

- Standard output of a sequencing run
- Large text file
- 4-line block per read
- Paired-end ➔ two files
- Reads order is random (but PE reads match)

What Linux command should we use to view a fastq file?

1. cat
2. less
3. vi

Read header

Sequence

Per-base quality

Read A

```
@ERR2348785.250578 250578/1
CCCAAAAGGCGGACCATTGGGCTGGCAAGGTACCAAATCTAGTTCTGTCTCTTATACA
+
BBBFFFFFFFFFFFIIFIFFFFFFI<FFIIFFFIIIFFFIFFIIFIB<FFFIFFFFFF
```

Read B

```
@ERR2348785.250579 250579/1
ATAACATACCATTTACATGTTGGAGGAAAAGATAAACGCTGTCTCTTATACACATCTC
+
BBBFFFFFFFFFFIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIFIIIIIIIIIFIIIIII
```

What can we learn from a fastq file?

- Number of reads
- Read length
- Average depth

$$D = \frac{L * N}{G}$$

Quality scores

- Also Q scores or Phred scores
- Usually between 0 and 40
- Each character encodes a number
- $$Q = -10 \log_{10} P.$$
- Higher Q → better quality
- Q_{20} → 99% correct call
- Q_{30} → 99.9% correct call

Table 1 ASCII Characters Encoding Q-scores 0–40

Symbol	ASCII Code	Q-Score	Symbol	ASCII Code	Q-Score	Symbol	ASCII Code	Q-Score
!	33	0	/	47	14	=	61	28
"	34	1	0	48	15	>	62	29
#	35	2	1	49	16	?	63	30
$	36	3	2	50	17	@	64	31
%	37	4	3	51	18	A	65	32
&	38	5	4	52	19	B	66	33
'	39	6	5	53	20	C	67	34
(	40	7	6	54	21	D	68	35
)	41	8	7	55	22	E	69	36
*	42	9	8	56	23	F	70	37
+	43	10	9	57	24	G	71	38
,	44	11	:	58	25	H	72	39
-	45	12	;	59	26	I	73	40
.	46	13	<	60	27			

The fasta format

- Used for storing general sequences
- Nucleotides or amino acids
- Anything goes!
  - Gene sequences
  - Full chromosomes
  - Proteins
  - miRNA...
- 2-line block per sequence
- No quality information

Header

```
>sp|Q56XQ6|PTR15_ARATH Protein NRT1/ PTR FAMILY 4.4 OS=Arabidopsis thaliana OX=3702 GN=NPF4.4 PE=2 SV=1
MDVHDLSEEAKRGVIHTSEESLDDLCVDFRGRPCRPSKHGGTRAALFVLGFQAFEMMAIA
AVGNNLITYVFNEMHFPLSKSANLVTNFIGTVFLLSLLGGFLSDSYLGSFRTMLVFGVIE
ISGFILLSVQAHLPELRPPECNMKSTTIHCVEANGYKAATLYTALCLVALGSGCLKPNII
SHGANQFQRKDLRKLSSFFNAAYFAFSMGQLIALTLLVWVQTHSGMDVGFGVSAAVMAAG
MISLVAGTSFYRNKPPSGSIFTPIAQVFVAAITKRKQICPSNPNMVHQPSTDLVRVKPLL
HSNKFRFLDKACIKTQGKAMESPWRLCTIEQVHQVKILLSVIPIFACTIIFNTILAQLQT
FSVQQGSSMNTHITKTFQIPPASLQAIPYIILIFFVPLYETFFVPLARKLTGNDSGISPL
QRIGTGLFLATFSMVAAALVEKKRRESFLEQNVMLSIFWIAPQFLIFGLSEMFTAVGLVE
FFYKQSSQSMQSFLTAMTYCSYSFGFYLSSVLVSTVNRVTSSNGSGTKEGWLGDNDLNKD
RLDHFYWLLASLSFINFFNYLFWSRWYSCDPSATHHSAEVNSLEALENGEIKDSTTEKPR
I
```

Sequence

```
>sp|Q9SX20|PTR18_ARATH Protein NRT1/ PTR FAMILY 3.1 OS=Arabidopsis thaliana OX=3702 GN=NPF3.1 PE=2 SV=1
MEEQSKNKISEEEKQLHGRPNRPKGGLITMPFIFANEICEKLAVVGFHANMISYLTTQLH
LPLTKAANTLTNFAGTSSLTPLLGAFIADSFAGRFWTITFASIIYQIGMTLLTISAIIPT
LRPPPCKGEEVCVVADTAQLSILYVALLLGALGSGGIRPCVVAFGADQFDESDPNQTTKT
WNYFNWYYFCMGAAVLLAVTVLVWIQDNVGWGLGLGIPTVAMFLSVIAFVGGFQLYRHLV
PAGSPFTRLIQVGVAAFRKRKLRMVSDPSLLYFNDEIDAPISLGGKLTHTKHMSFLDKAA
IVTEEDNLKPGQIPNHWRLSTVHRVEELKSVIRMGPIGASGILLITAYAQQGTFSLQQAK
TMNRHLTNSFQIPAGSMSVFTTVAMLTTIIFYDRVFVKVARKFTGLERGITFLHRMGIGF
VISIIATLVAGFVEVKRKSVAIEHGLLDKPHTIVPISFLWLIPQYGLHGVAEAFMSIGHL
EFFYDQAPESMRSTATALFWMAISIGNYVSTLLVTLVHKFSAKPDGSNWLPDNNLNRGRL
EYFYWLITVLQAVNLVYYLWCAKIYTYKPVQVHHSKEDSSPVKEELQLSNRSLVDE
```

Which of the following commands will output all fasta headers?

1. `grep @ file.fasta | less`

2. `grep > file.fasta | less`

3. `grep “>” file.fasta | less`

4. `grep file.fasta “>” | less`

QA-ing raw sequencing data with FastQC

- A common first step when getting your fastq data
- Generates a simple HTML report
- Can help detect issues with the data

Running:

```
fastqc <file1.fastq> <file2.fastq> … <file n.fastq>
```

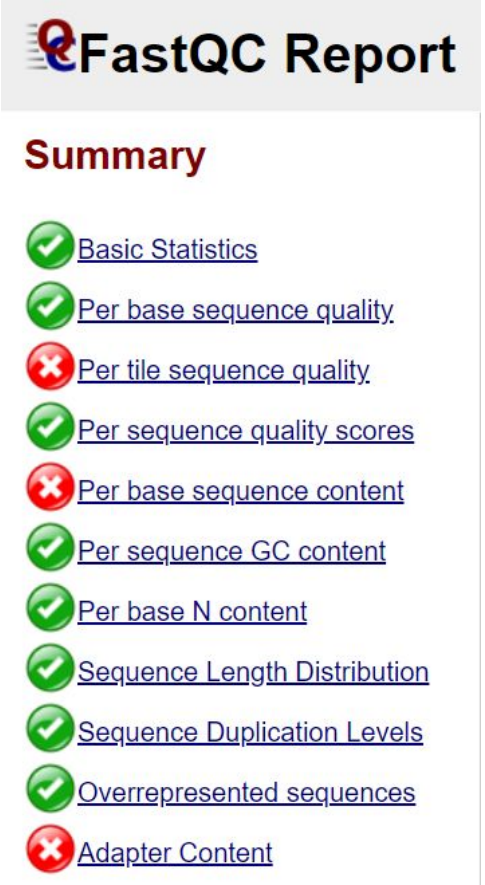
For more options:

```
fastqc -h | less
```

The FastQC report

- View using your favorite web browser
- Contains multiple analysis modules
- Issues “Warning” and “Failure” messages per module
- Remember to look both at R1 and R2

www.bioinformatics.babraham.ac.uk/projects/fastqc

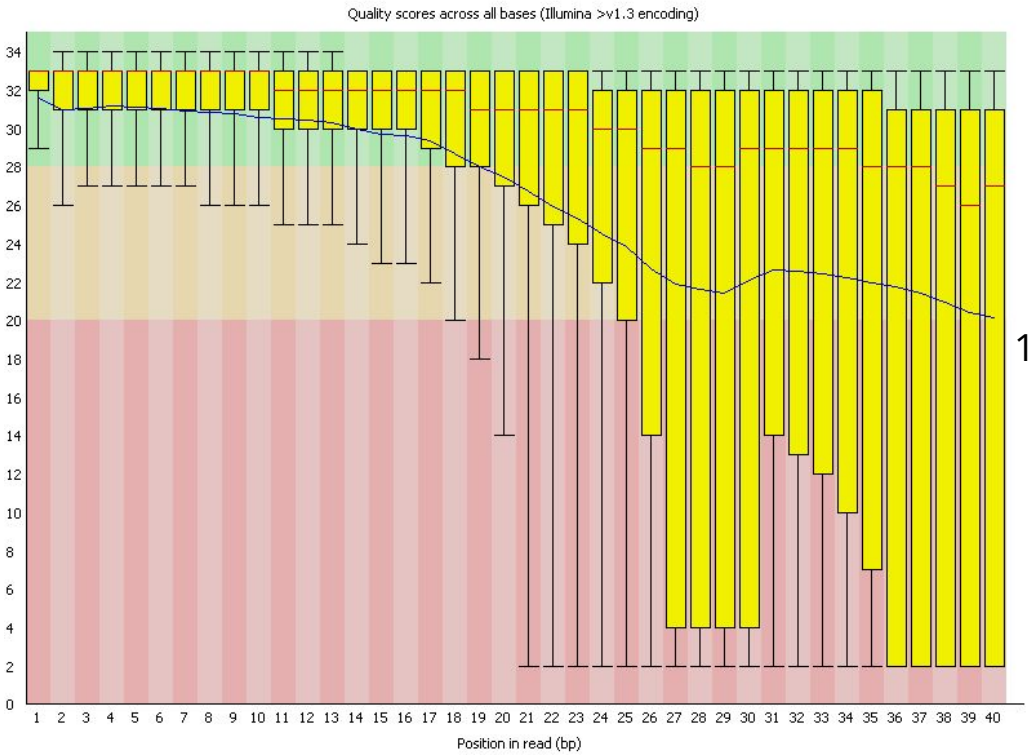


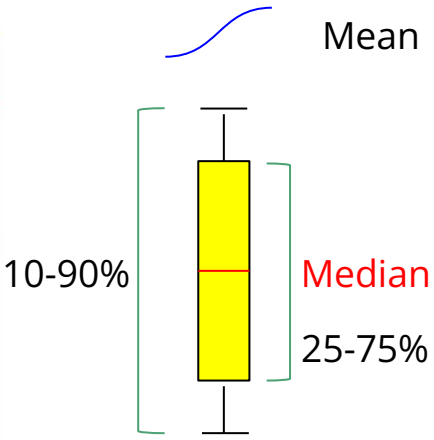
Basic statistics

Basic Statistics

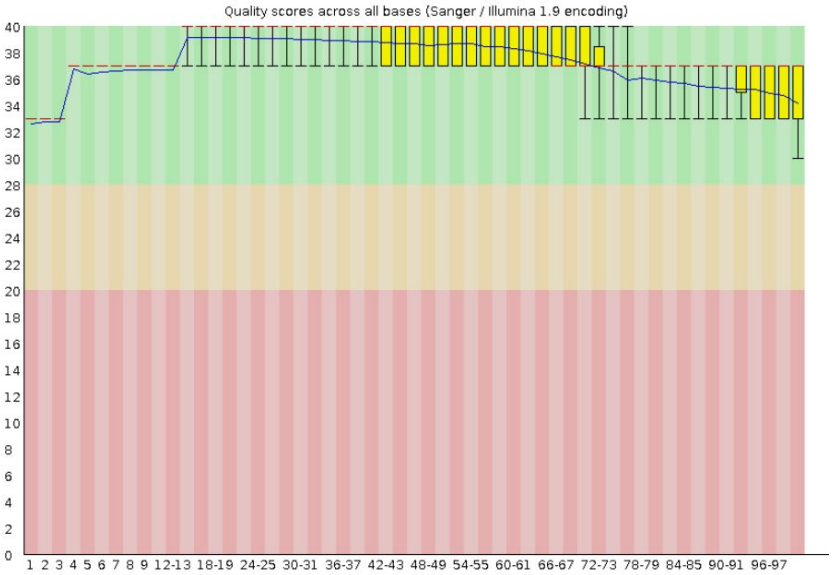
Measure	Value
Filename	ERR2834525_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	3161562
Sequences flagged as poor quality	0
Sequence length	101
%GC	41

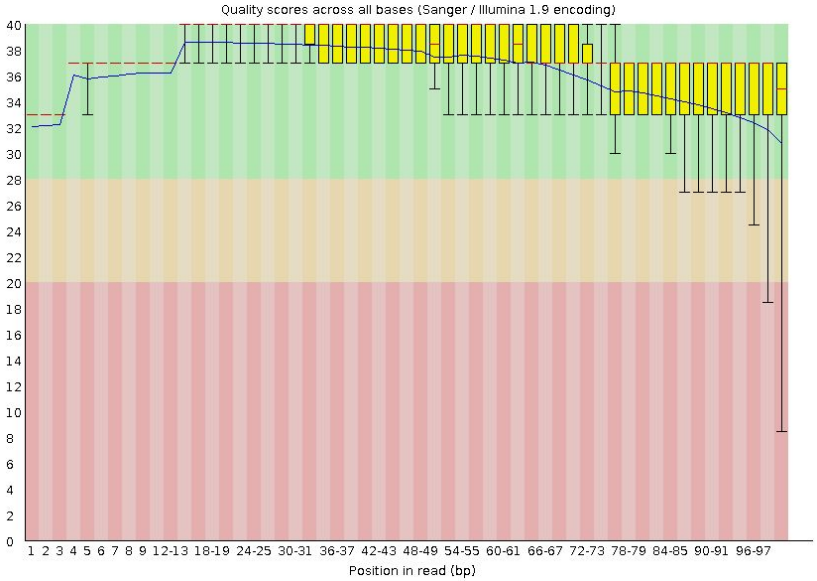
Per base sequence quality



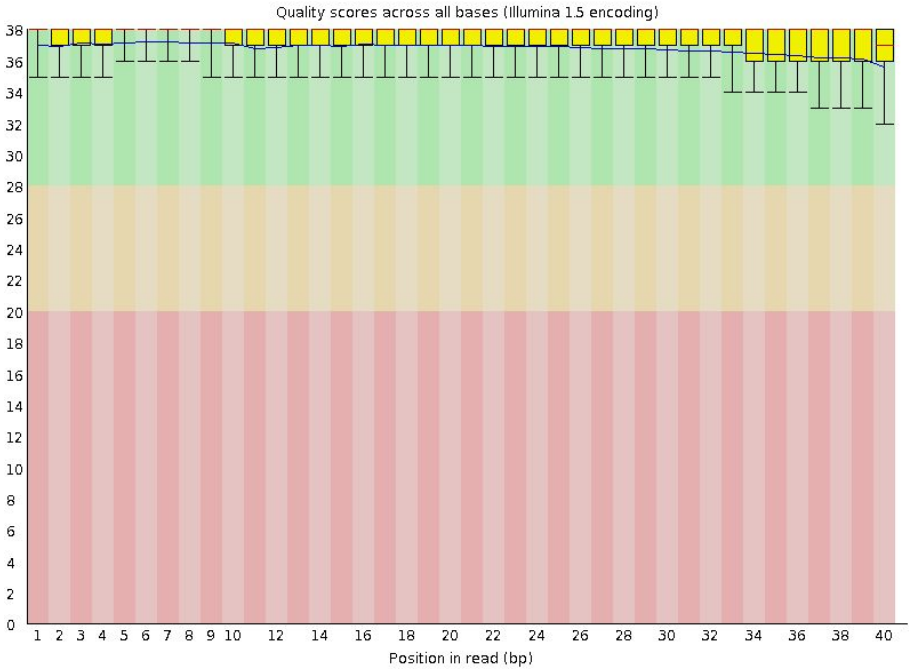


Per base sequence quality - R1 vs. R2

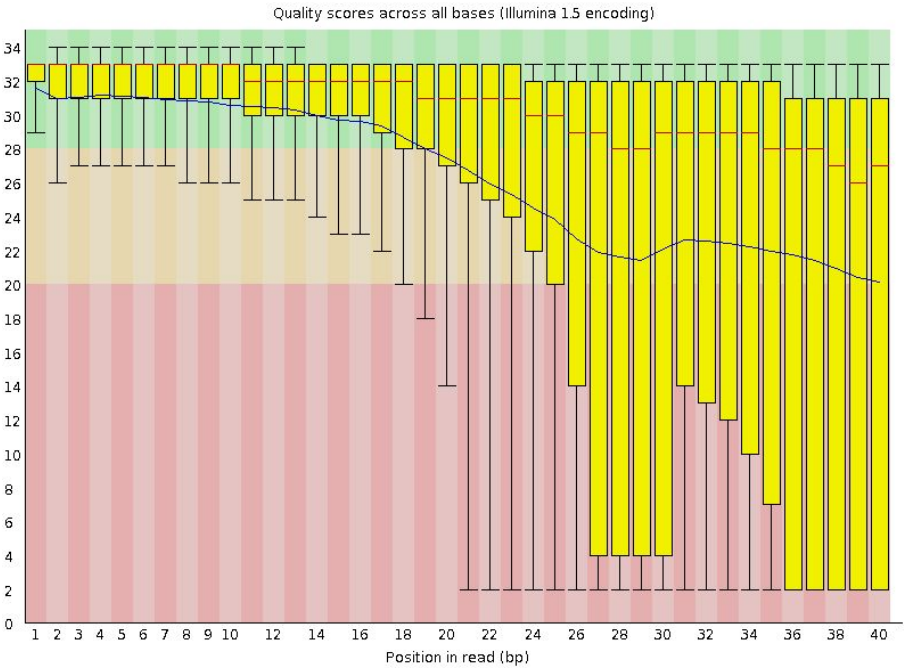




Good

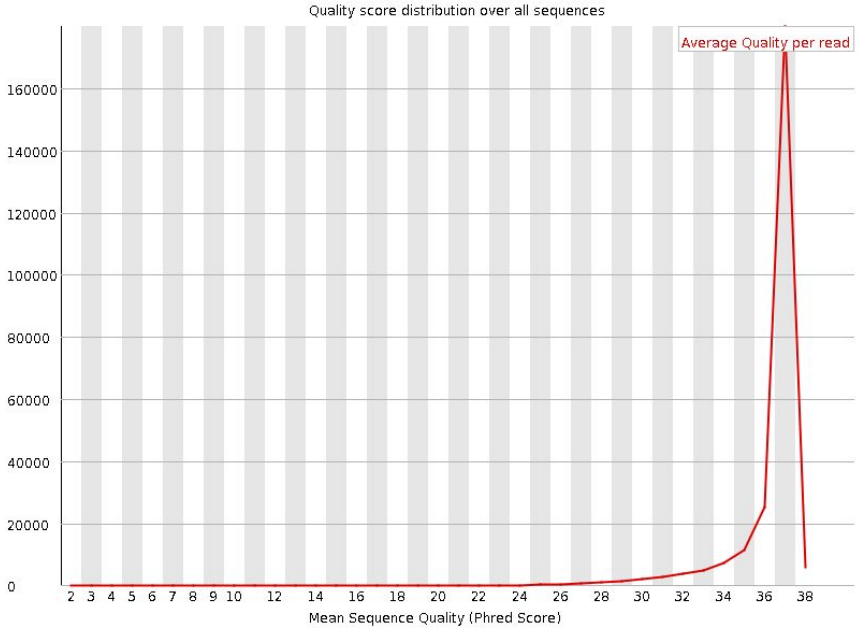

Quality scores across all bases (Illumina 1.5 encoding)
Position in read (bp)

Bad

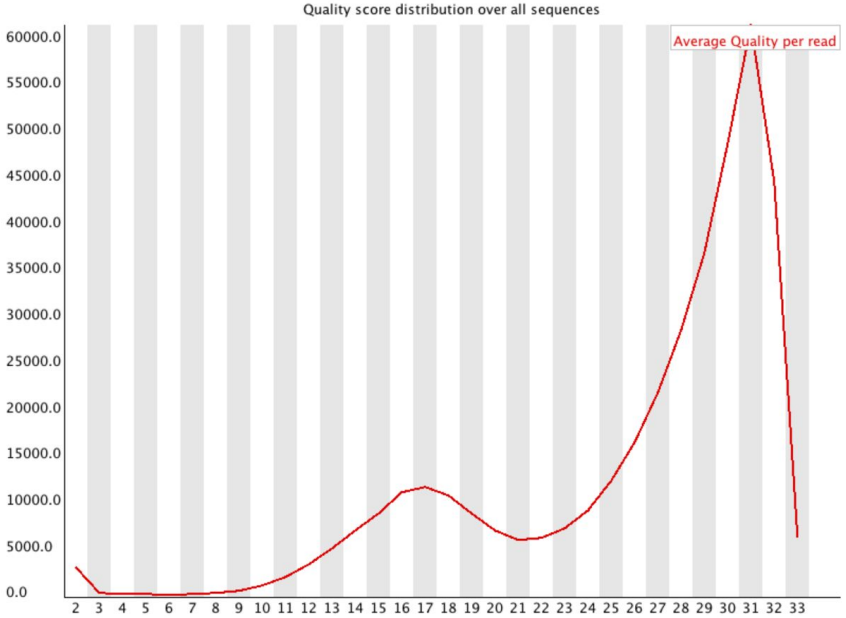

Quality scores across all bases (Illumina 1.5 encoding)
Position in read (bp)

Histogram of mean read quality

Good

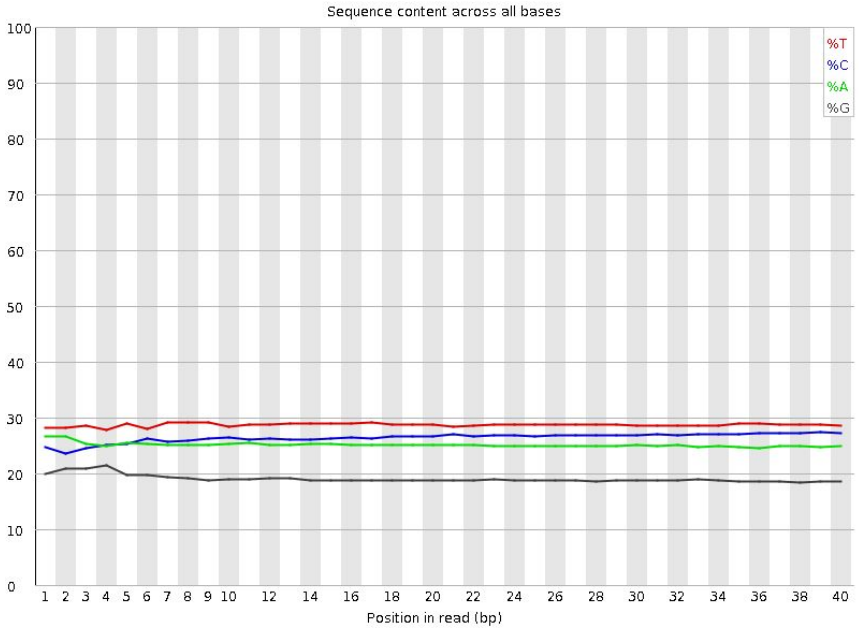


Bad

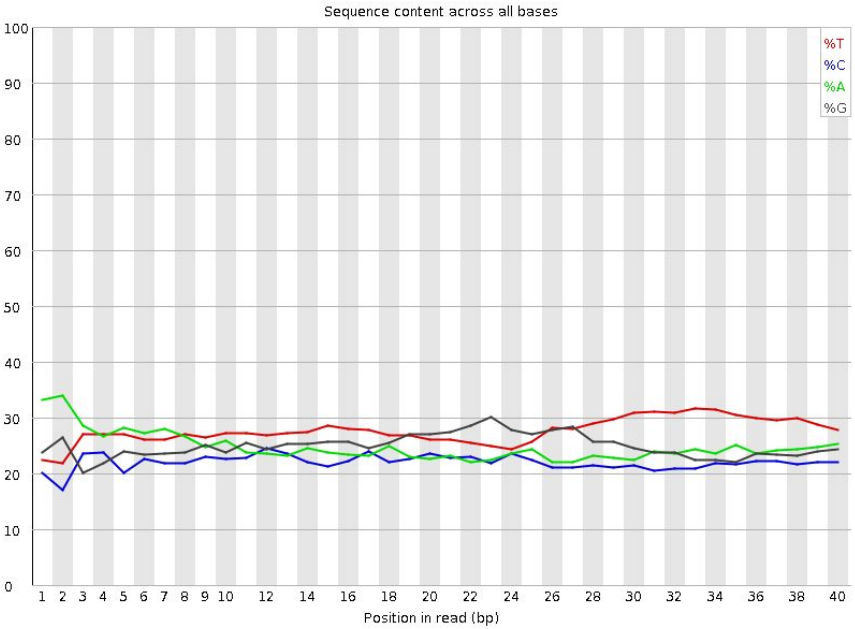


Per base sequence content

Good

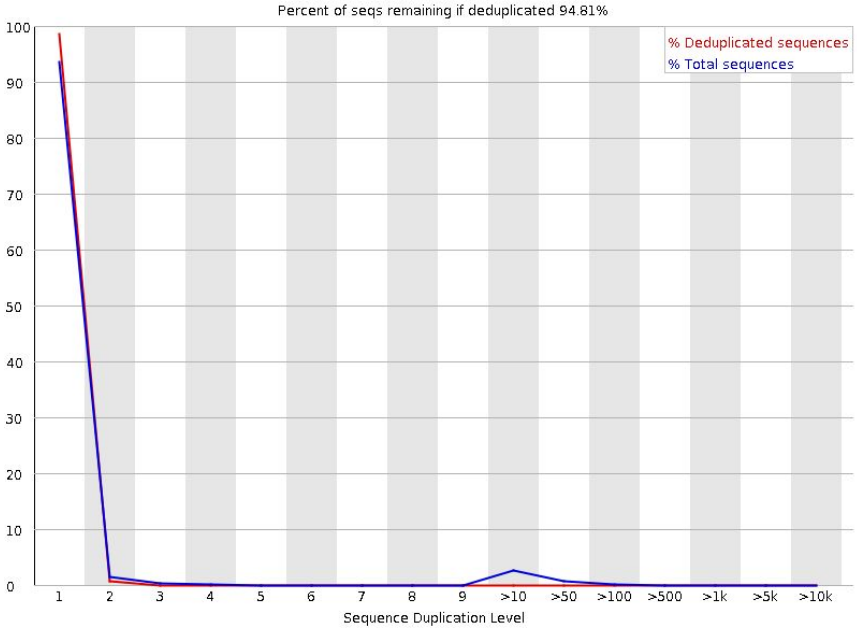


Bad

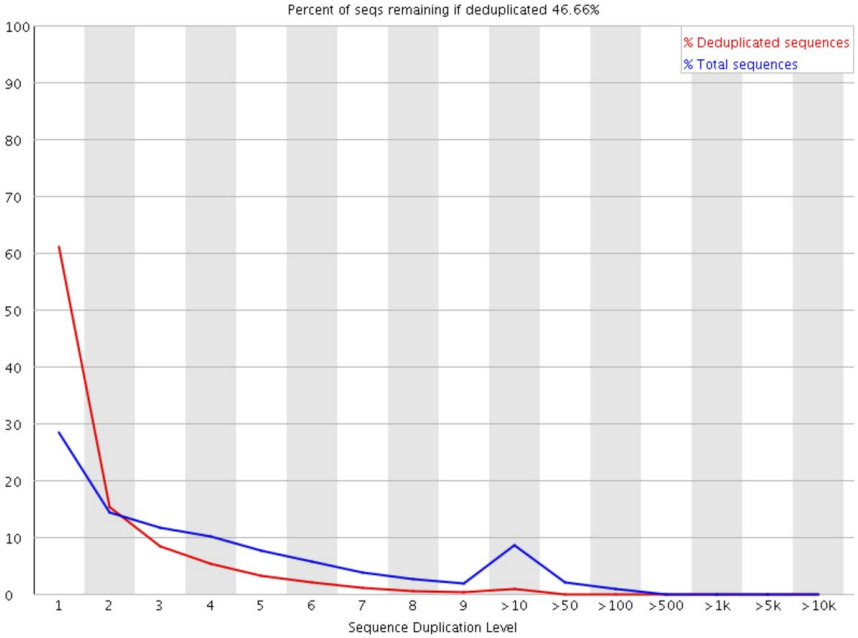


Duplicate reads

Good

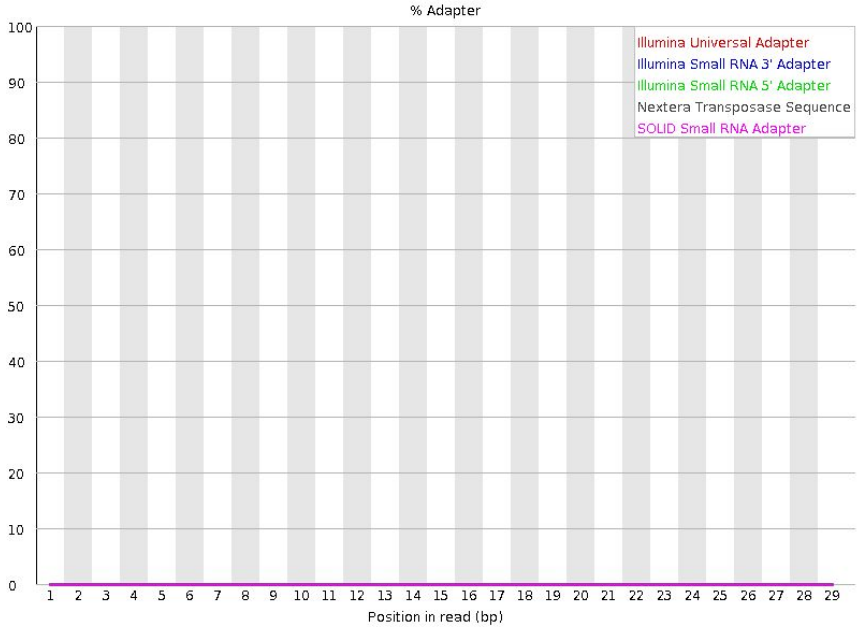


Bad

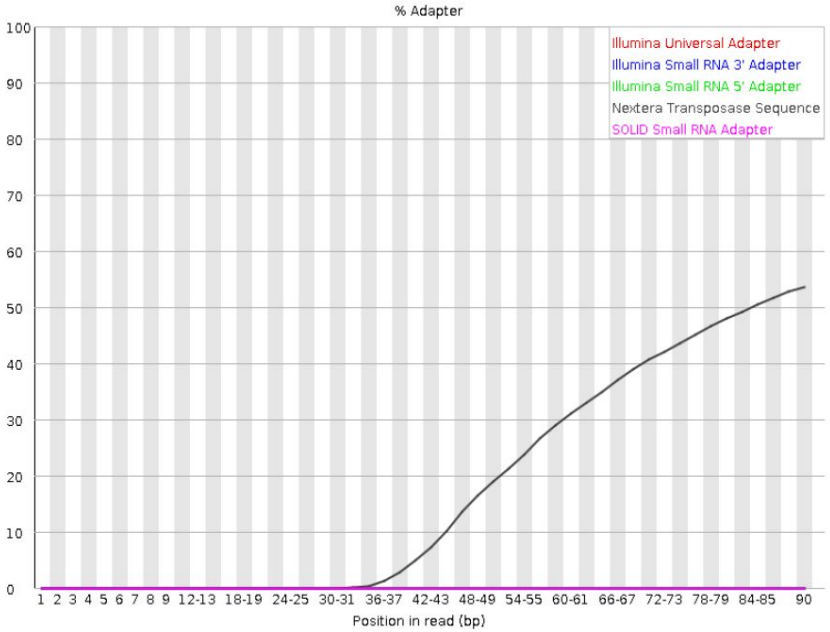


Adapter sequence

Good



Bad



General advice

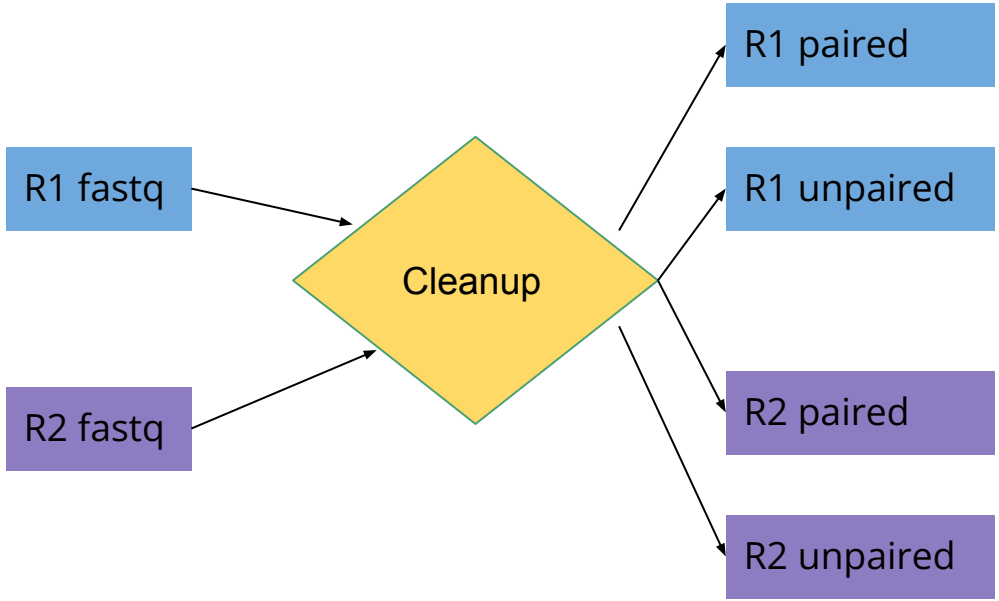
- Use FastQC report to decide on next steps
- Warnings might not affect downstream analysis
- Mainly useful when comparing results

Data cleanup with Trimmomatic - QC

- Takes single- or paired- end fastq
- **Remove:**
  - Low quality reads
  - Very short reads
- **Trim:**
  - Low quality bases
  - Adapter sequences
- Contains multiple cleanup modules

www.usadellab.org/cms/?page=trimmomatic

General workflow (paired end)



- Takes two files
- Performs one or more cleanup steps
- Outputs **four** files

Cleanup modules

- Find and trim adapter sequences
- Trim *n* bases from 5’ and 3’ ends
- Trim *n* bases from 5’ and 3’ ends if quality is too low
- Trim 3’ end with a flexible window size
- Remove reads shorter than a specific length
- Remove reads with mean quality below a specific Q

Running Trimmomatic

```
trimmomatic PE <R1 path> <R2 path> -baseout <output base>
```

Follow this with one or more cleanup modules + parameters

Use the format:

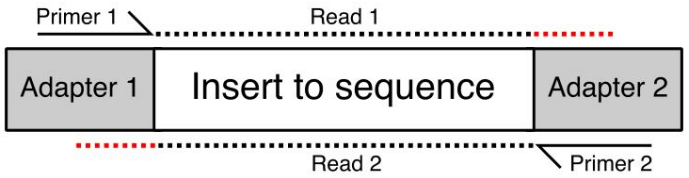
```
<module>:<parameter 1>:<parameter 2>:...:<parameter n>
```

Each module has its own set of parameters.

Remove adapters - `ILLUMINACLIP`

Mandatory parameters:

- *fastaWithAdaptersEtc*
- *seedMismatches* - usually 2-3
- *palindromeClipThreshold* - usually ~30
- *simpleClipThreshold* - usually 7-15



Command example:

```
trimmomatic PE ERR2834525_1.fastq ERR2834525_2.fastq
-baseout ERR2834525 ILLUMINACLIP:NexteraPE-PE.fa:2:30:10
```

Trim low quality 3’ - SLIDINGWINDOW

```
@ERR2348785.1 1/1
CAATGGGATGAATATAAATTCACCCAAGAACTTAGCAACAACAGACAAGTACAATAGAACTCTGCACAACGATGACGC
+
BBBFFFFFFFFFFIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIFIFIIIIIIIIIIIIIIIIIII//*/.+.
```

B= 33; F = 37

W = (33*3 + 37*2)/5 = 34.6

35.4

12.8

Parameters:

- windowSize - usually 4-5
- *requiredQuality* - usually ~15

```
trimmomatic PE ERR2834525_1.fastq ERR2834525_2.fastq -baseout
ERR2834525 SLIDINGWINDOW:5:15
```

Remove short reads - `MINLEN`

Parameters:

- Length - minimum read length (usually 50-70)

```
trimmomatic PE ERR2834525_1.fastq ERR2834525_2.fastq
-baseout ERR2834525 MINLEN:60
```

Combining cleanup modules

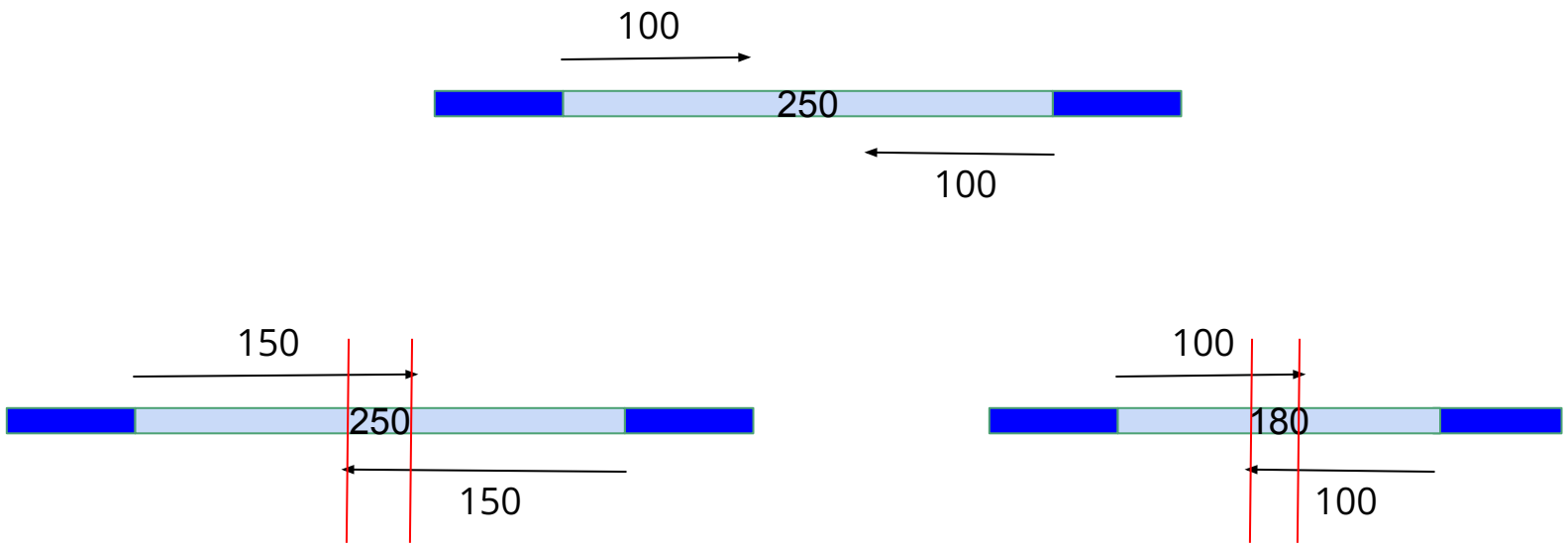
Order matters!

Usually start with adapter trimming and end with length filtration

Example:

```
trimmomatic PE ERR2834525_1.fastq ERR2834525_2.fastq
-baseout ERR2834525 ILLUMINACLIP:NexteraPE-PE.fa:2:30:10
SLIDINGWINDOW:5:15 MINLEN:70
```

Paired-end read merging



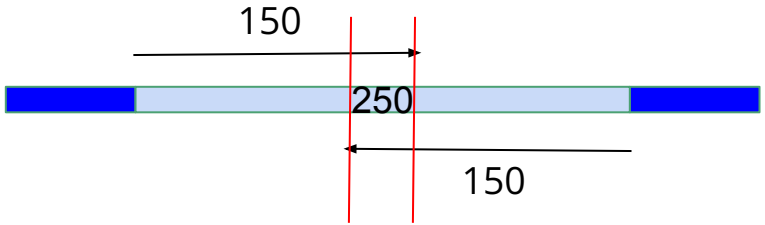
In what cases will PE merging occur, and what will the merged read length be?

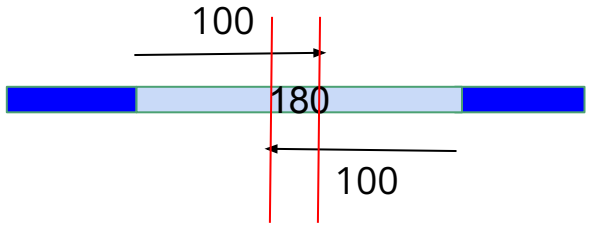
1. insert size < 2 x read size - required overlap ; L = insert size

2. insert size < 2 x read size - 2 x required overlap ; L = insert size

3. insert size < 2 x read size - required overlap ; L = insert size - required overlap

4. insert size < 2 x read size - 2 x required overlap ; L = insert size - required overlap





Paired-end read merging with FLASH

- Input: two fastq files
- Output: Merged fastq + two not-combined fastq
- Important options:
  - -m - min overlap - usually 10bp
  - -x - max mismatch ratio - usually 0.25

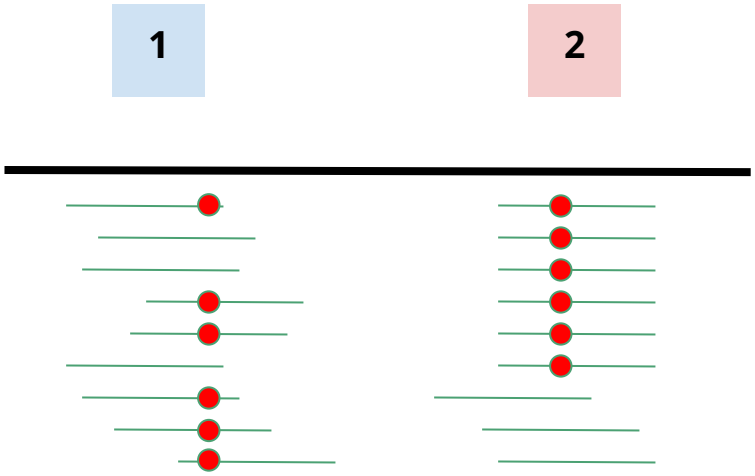
Running FLASH:

```
flash <R1.fastq> <R2.fastq> -d <out dir> -m <n> -x <ratio>
```

ccb.jhu.edu/software/FLASH

Duplicate read removal

- Why and when is it important?
- Causes:
  - Same region sequenced multiple times - low probability
  - Repetitive regions
  - PCR duplicates
  - Sequencing duplicates
  - Optical artifacts



Duplicate read removal with fastuniq

- Input: a file with the list of input fastq files
- Output: de-duplicated fastq files

  Note: fastuniq will concat all input files

  Do not mix libraries with different insert sizes

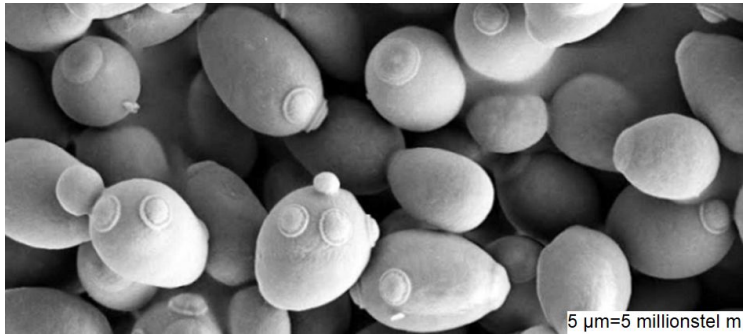
Running fastuniq:

```
fastuniq -i <list> -o <output R1> -p <output R2>
```

sourceforge.net/projects/fastuniq

Introducing: our exercise data set

- Baker's yeast / budding yeast - *S. cerevisiae*
- Genome size: 12.1 Mb
- Usually haploid
- Industrial importance
- Model organism



The RM11 strain

- *S. cerevisiae* has many strains
- Reference strain - **S288C** - used in beer manufacturing
- **RM11** - used in wine manufacturing
- Derived from a californian isolate

You were hired by a large winery to learn more about the RM11 strain.

Can you make the world a drunker place?