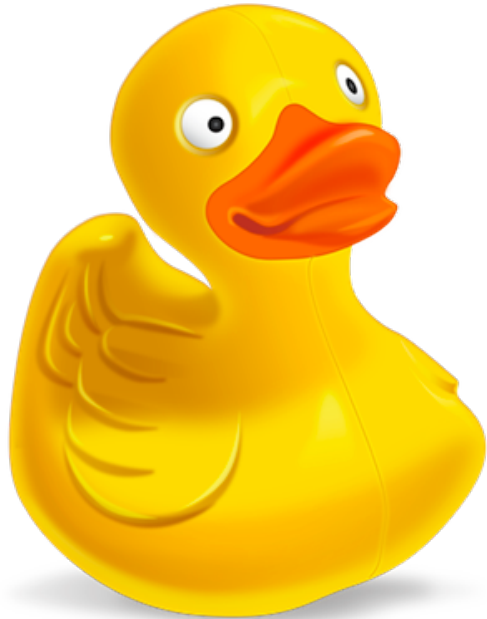


Processing RNA-seq Data

2020-07-23

Getting Files Off the Server

Mac



Cyberduck

Windows



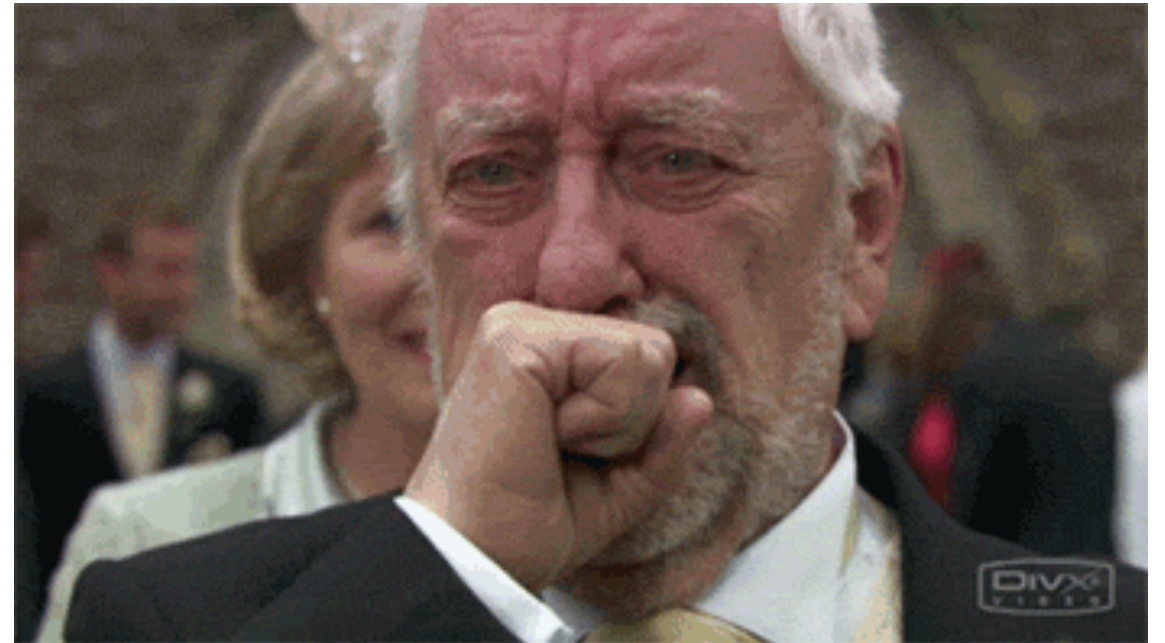
Setting Up a Project

Why do I need to document my work?

**1. For your future
self**

Why do I need to document my work?

1. For your future
self



Why do I need to document my work?

1. For your future
self



Why do I need to document my work?

1. For your future
self

2. For everybody
else



Why do I need to document my work?

1. For your future self
2. For everybody else
For your next step
in your career



Setting Up Project Documentation

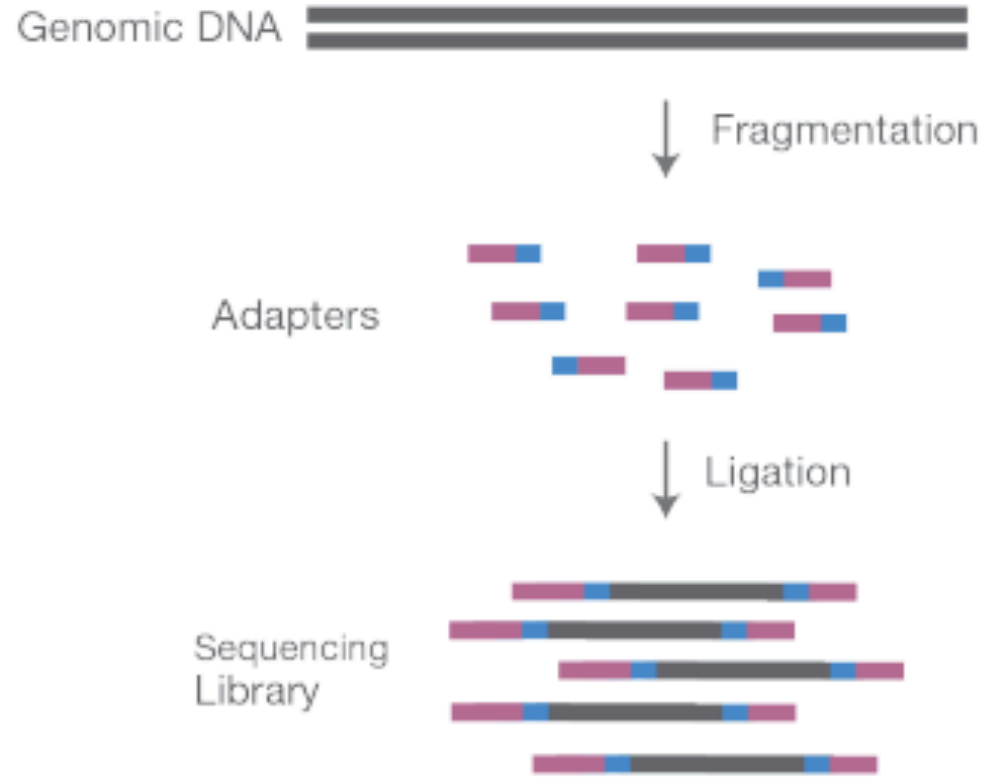
1. Create a new folder
2. Open a plain text file in a new to take notes in / document your work
3. Make that folder into a Git repository and back it up to GitHub

Quick Review: How does Illumina sequencing work?

Illumina Sequencing

- General overview
- For RNA-seq, extract RNA and remove ribosomal RNA as well

A. Library Preparation

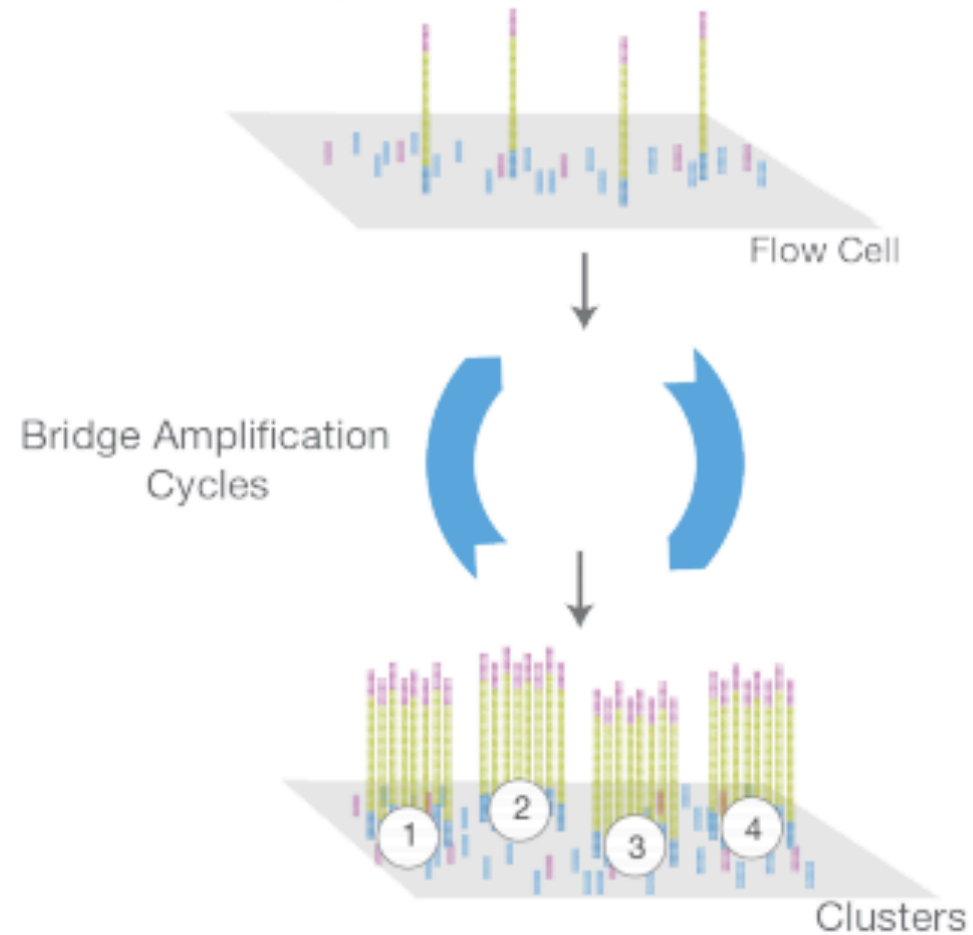


NGS library is prepared by fragmenting a gDNA sample and ligating specialized adapters to both fragment ends.

Illumina Sequencing

Make tiles of identical DNA to read

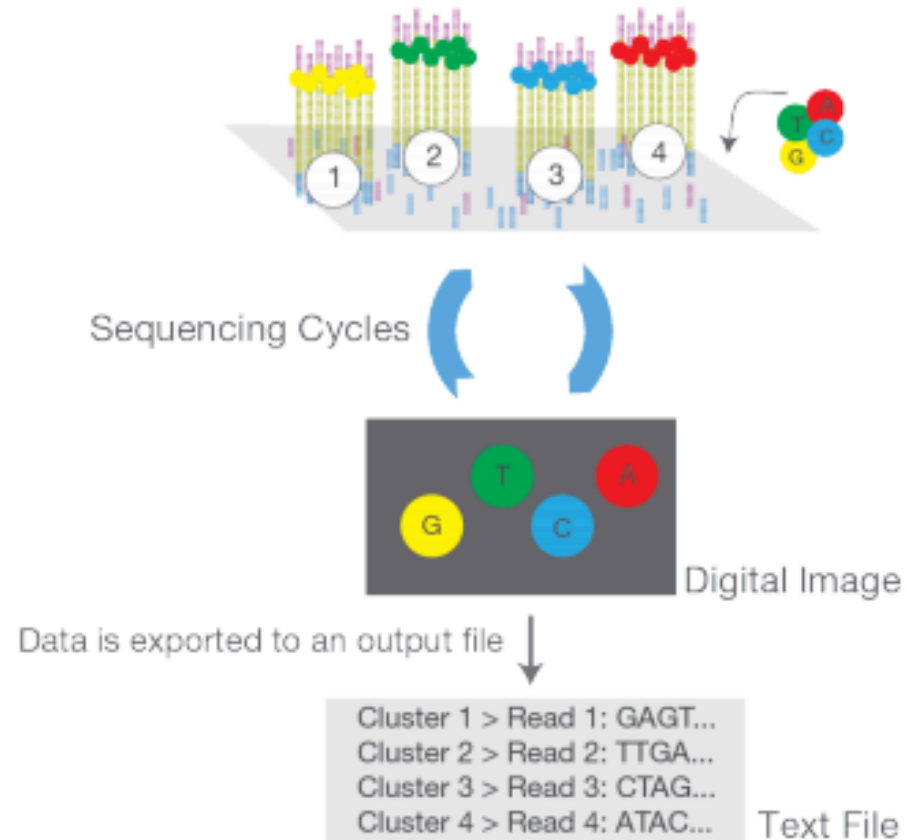
B. Cluster Amplification



Library is loaded into a flow cell and the fragments are hybridized to the flow cell surface. Each bound fragment is amplified into a clonal cluster through bridge amplification.

Illumina Sequencing

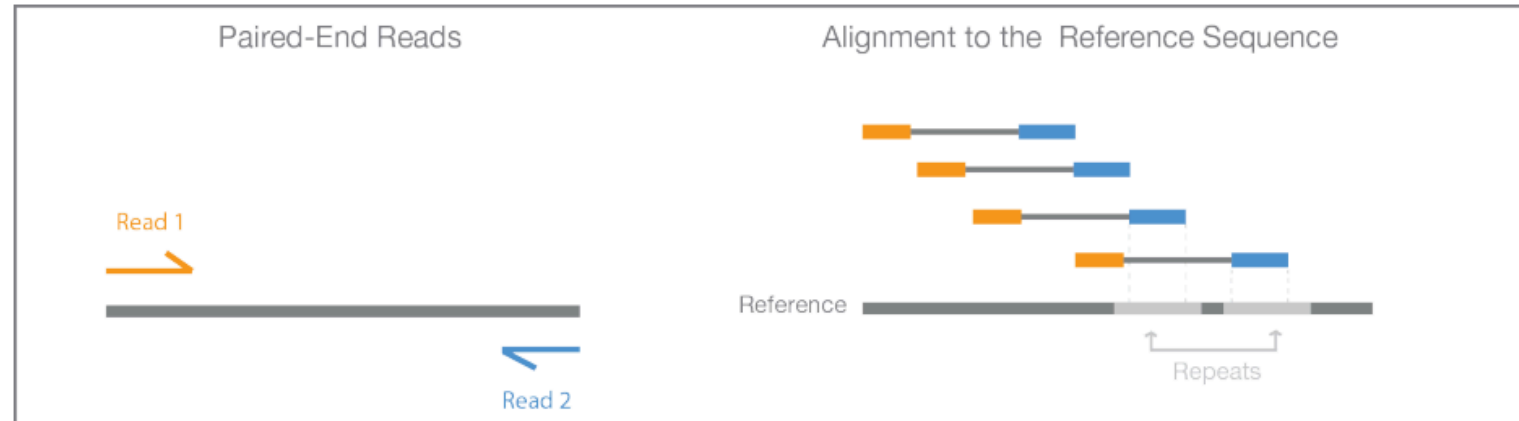
C. Sequencing



Sequencing reagents, including fluorescently labeled nucleotides, are added and the first base is incorporated. The flow cell is imaged and the emission from each cluster is recorded. The emission wavelength and intensity are used to identify the base. This cycle is repeated "n" times to create a read length of "n" bases.

Paired-End Sequencing

- Sequence both ends of the fragment
- Because sequencing is always 5' to 3', the read pairs will be in the opposite orientation
- 90% of the time, the programs you use will be aware of the difference in orientation and take care of it for you
- Because the distance between the pairs is known (depends on the sequence length you asked for) mapping is more accurate, especially in highly repetitive regions of the genome
- For RNA-seq, paired end reads are necessary if you want to look at alternative splicing
- More expensive than single end sequencing



What does raw sequencing data
look like?

FastQ Files

FastQ Files

- Fastq files (usually) end in either `fastq.gz` or `fq.gz` (or they can be missing the `.gz` extension)

FastQ Files

- Fastq files (usually) end in either `fastq.gz` or `fq.gz` (or they can be missing the `.gz` extension)
- File names will have some combination of the following information (depends on the sequencer):
 - Sample ID
 - Lane
 - Read number
 - Unique ID from the company or sequencer

FastQ Files

- Fastq files (usually) end in either `fastq.gz` or `fq.gz` (or they can be missing the `.gz` extension)
- File names will have some combination of the following information (depends on the sequencer):
 - Sample ID
 - Lane
 - Read number
 - Unique ID from the company or sequencer

```
[[kkeith@cbix rnaseq_data]$ ll
total 46700
-rw-r--r--. 1 kkeith research 4276449 Dec 18 10:55 dac1_chr21_R1.fastq.gz
-rw-r--r--. 1 kkeith research 4441834 Dec 18 10:55 dac1_chr21_R2.fastq.gz
-rw-r--r--. 1 kkeith research 4118184 Dec 18 10:55 dac2_chr21_R1.fastq.gz
-rw-r--r--. 1 kkeith research 4296786 Dec 18 10:55 dac2_chr21_R2.fastq.gz
-rw-r--r--. 1 kkeith research 4336091 Dec 18 10:56 dac3_chr21_R1.fastq.gz
-rw-r--r--. 1 kkeith research 4519748 Dec 18 10:56 dac3_chr21_R2.fastq.gz
-rw-r--r--. 1 kkeith research 3652875 Dec 18 10:57 siC1_chr21_R1.fastq.gz
-rw-r--r--. 1 kkeith research 3830628 Dec 18 10:57 siC1_chr21_R2.fastq.gz
-rw-r--r--. 1 kkeith research 3941656 Dec 18 10:58 siC2_chr21_R1.fastq.gz
-rw-r--r--. 1 kkeith research 4103667 Dec 18 10:58 siC2_chr21_R2.fastq.gz
-rw-r--r--. 1 kkeith research 3078529 Dec 18 10:59 siC3_chr21_R1.fastq.gz
-rw-r--r--. 1 kkeith research 3198840 Dec 18 10:59 siC3_chr21_R2.fastq.gz
```

FastQ Files

- Fastq files (usually) end in either `fastq.gz` or `fq.gz` (or they can be missing the `.gz` extension)
- File names will have some combination of the following information (depends on the sequencer):
 - Sample ID
 - Lane
 - Read number
 - Unique ID from the company or sequencer

```
[[kkeith@cbix rnaseq_data]$ ll
total 46700
-rw-r--r--. 1 kkeith research 4276449 Dec 18 10:55 dac1_chr21_R1.fastq.gz
-rw-r--r--. 1 kkeith research 4441834 Dec 18 10:55 dac1_chr21_R2.fastq.gz
-rw-r--r--. 1 kkeith research 4118184 Dec 18 10:55 dac2_chr21_R1.fastq.gz
-rw-r--r--. 1 kkeith research 4296786 Dec 18 10:55 dac2_chr21_R2.fastq.gz
-rw-r--r--. 1 kkeith research 4336091 Dec 18 10:56 dac3_chr21_R1.fastq.gz
-rw-r--r--. 1 kkeith research 4519748 Dec 18 10:56 dac3_chr21_R2.fastq.gz
-rw-r--r--. 1 kkeith research 3652875 Dec 18 10:57 siC1_chr21_R1.fastq.gz
-rw-r--r--. 1 kkeith research 3830628 Dec 18 10:57 siC1_chr21_R2.fastq.gz
-rw-r--r--. 1 kkeith research 3941656 Dec 18 10:58 siC2_chr21_R1.fastq.gz
-rw-r--r--. 1 kkeith research 4103667 Dec 18 10:58 siC2_chr21_R2.fastq.gz
-rw-r--r--. 1 kkeith research 3078529 Dec 18 10:59 siC3_chr21_R1.fastq.gz
-rw-r--r--. 1 kkeith research 3198840 Dec 18 10:59 siC3_chr21_R2.fastq.gz
```

```
[[kkeith@cbix Rawdata]$ ll [fm]*/*.fq.gz
-rw-r--r--. 1 jjelinek research 4994684738 Dec 10 20:02 f45y4/f45y4_CKDL190143587-1a-6_H723FBBXX_L1_1.fq.gz
-rw-r--r--. 1 jjelinek research 5329097223 Dec 10 20:03 f45y4/f45y4_CKDL190143587-1a-6_H723FBBXX_L1_2.fq.gz
-rw-r--r--. 1 jjelinek research 4378414462 Dec 10 20:01 f53y6/f53y6_CKDL190143587-1a-12_H723FBBXX_L1_1.fq.gz
-rw-r--r--. 1 jjelinek research 4688210826 Dec 10 20:02 f53y6/f53y6_CKDL190143587-1a-12_H723FBBXX_L1_2.fq.gz
-rw-r--r--. 1 jjelinek research 4353424157 Dec 10 20:01 f61y8/f61y8_CKDL190143587-1a-19_H723FBBXX_L1_1.fq.gz
-rw-r--r--. 1 jjelinek research 4589695705 Dec 10 20:01 f61y8/f61y8_CKDL190143587-1a-19_H723FBBXX_L1_2.fq.gz
-rw-r--r--. 1 jjelinek research 4595389700 Dec 10 20:00 m38y1/m38y1_CKDL190143587-1a-2_H723FBBXX_L1_1.fq.gz
-rw-r--r--. 1 jjelinek research 4905936742 Dec 10 20:01 m38y1/m38y1_CKDL190143587-1a-2_H723FBBXX_L1_2.fq.gz
-rw-r--r--. 1 jjelinek research 3745371888 Dec 10 20:00 m45y3/m45y3_CKDL190143587-1a-5_H723FBBXX_L1_1.fq.gz
-rw-r--r--. 1 jjelinek research 3963991106 Dec 10 20:00 m45y3/m45y3_CKDL190143587-1a-5_H723FBBXX_L1_2.fq.gz
-rw-r--r--. 1 jjelinek research 4237208070 Dec 10 20:01 m53y5/m53y5_CKDL190143587-1a-7_H723FBBXX_L1_1.fq.gz
-rw-r--r--. 1 jjelinek research 4497568575 Dec 10 20:02 m53y5/m53y5_CKDL190143587-1a-7_H723FBBXX_L1_2.fq.gz
-rw-r--r--. 1 jjelinek research 4061248079 Dec 10 20:00 m61y7/m61y7_CKDL190143587-1a-16_H723FBBXX_L1_1.fq.gz
-rw-r--r--. 1 jjelinek research 4308294343 Dec 10 20:01 m61y7/m61y7_CKDL190143587-1a-16_H723FBBXX_L1_2.fq.gz
```

What does a raw read look like?

What does a raw read look like?

Diagram illustrating the structure of a raw sequencing read, showing the read ID, pair/mate number, sequence, and quality scores.

read ID

pair/mate number

@SN930:673:HT5JVBCXY:2:2103:18909:8888/1

CTTTATTTCTGCCTTCATTTTGTTATGTACCCAGTAGTCATTCAGGAGCAGGTTGTTTCAGTTTCCATGTAGTTGAGCAGTTTTGAGTGAGTTTCTTAATCCTGAGTTCTAGTTTGATTGCACTGTGGTCTGAGAGACAGTTTGTTATAAT

+

GGGGI...IGGGIIGGGIGGGIGIGIGIIG

nothing

sequence

What does a raw read look like?

read ID

```
@SN930:673:HT5JVBCXY:  
CTTTATTTCTGCCTTCATTT  
+  
GGGGGIIIIIIIIIIIIIIIIIIII
```

nothing

sequ

NOTE: YOUR SEQUENCING DATA WILL FREQUENCLY LOOK DIFFERENT

- File names vary from sequencer to sequencer
- Read IDs also depend on the sequencer and will probably be different from the example here
- Quality encoding can be different if you're using older or public data

```
GAGAGACAGTTTGTTATAAT  
IIIGGGIGIGGGIGIGIGIGIIG
```

```
.....  
K.....  
F.....  
J.....  
.....  
hijklmnopqrstuvwxyz{|}~  
|  
126
```

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)

Quality Check

FastQC

- Before going forward, we want to check the quality of the data
 - How much did the sequencer fail?
 - Did we sequence mostly our sample DNA?
- FastQC is a program from the Babraham Institute in the UK that creates an html report on the quality of the sequencing data
 - Has 11 quality control checks that it does

Basic Statistics

Good Quality



Basic Statistics

Measure	Value
Filename	good_sequence_short.txt
File type	Conventional base calls
Encoding	Illumina 1.5
Total Sequences	250000
Sequences flagged as poor quality	0
Sequence length	40
%GC	45

Bad Quality



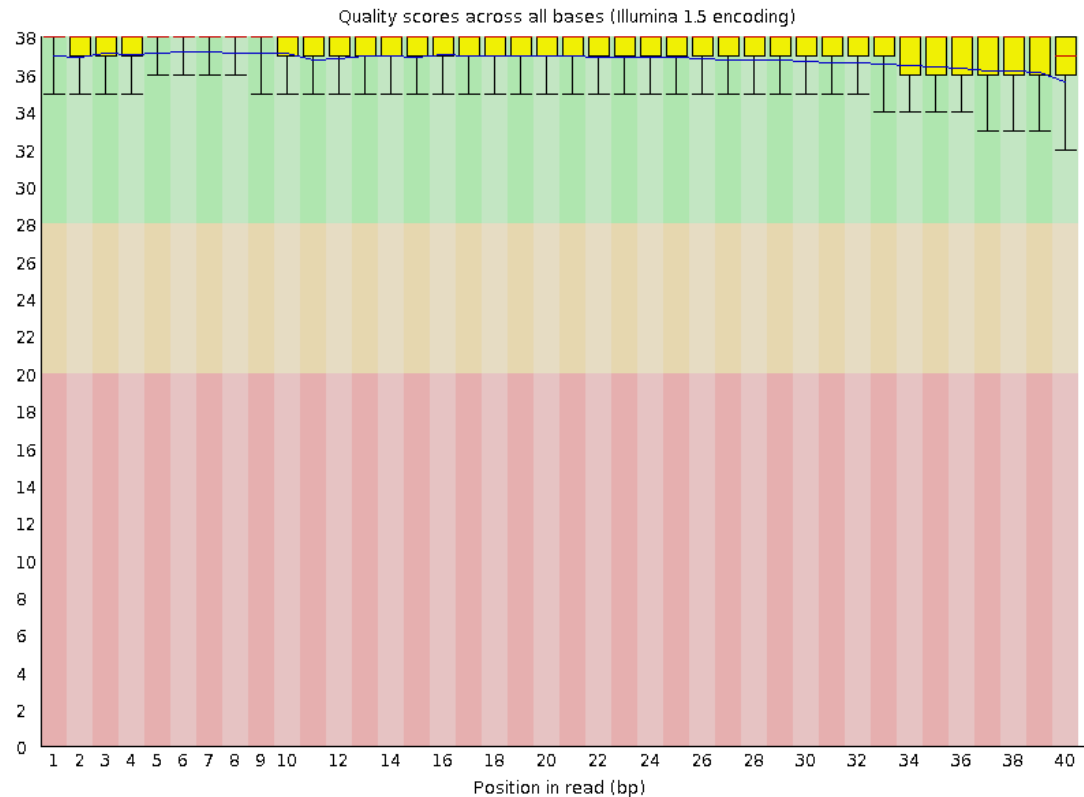
Basic Statistics

Measure	Value
Filename	bad_sequence.txt
File type	Conventional base calls
Encoding	Illumina 1.5
Total Sequences	395288
Sequences flagged as poor quality	0
Sequence length	40
%GC	47

Per base sequence quality

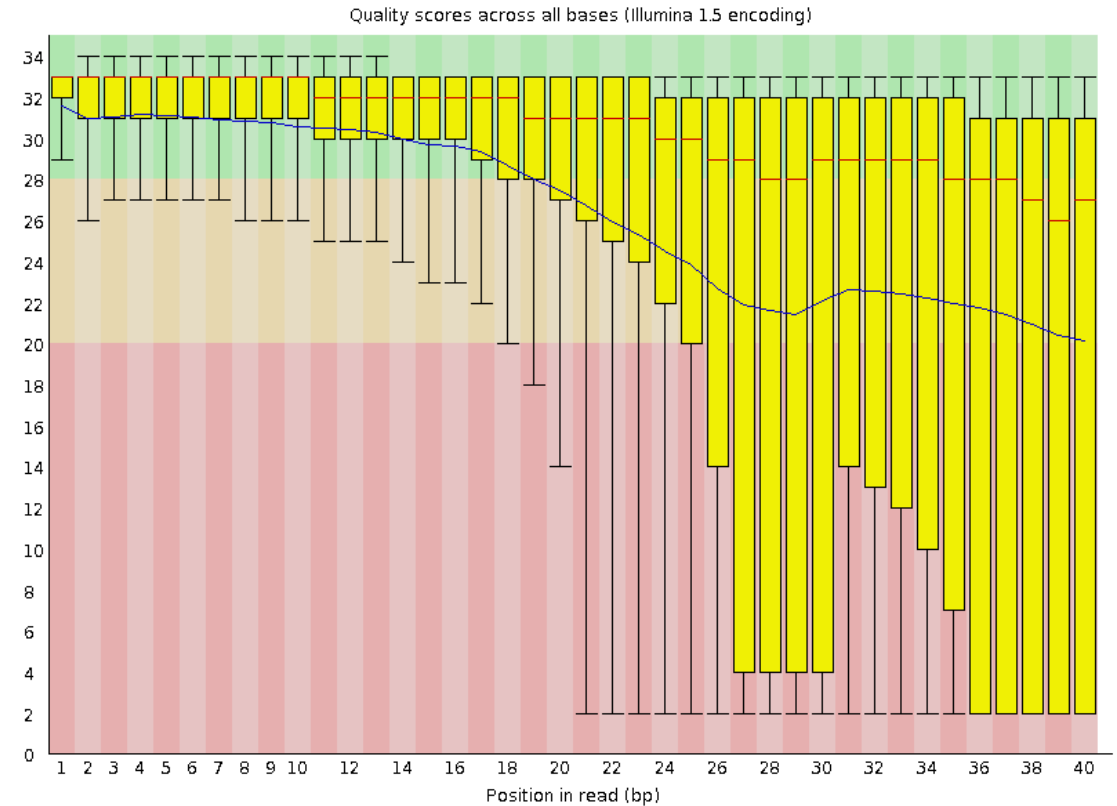
Good Quality

✔ Per base sequence quality



Bad Quality

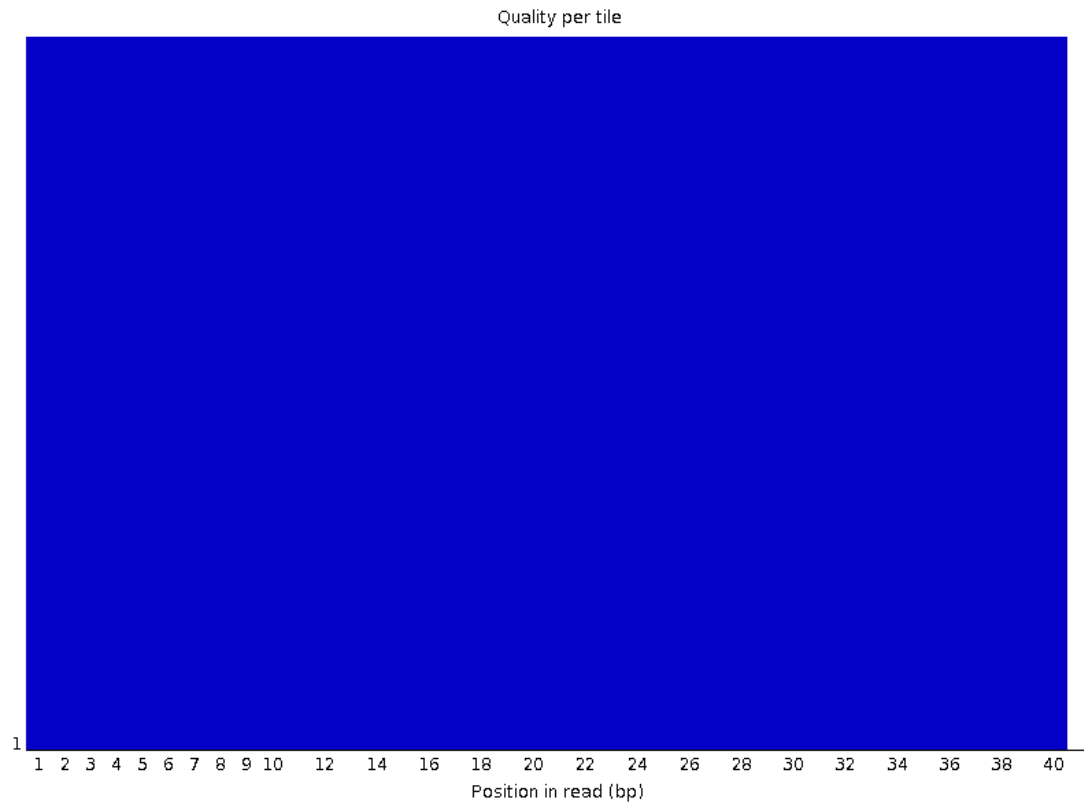
✘ Per base sequence quality



Per tile sequence quality

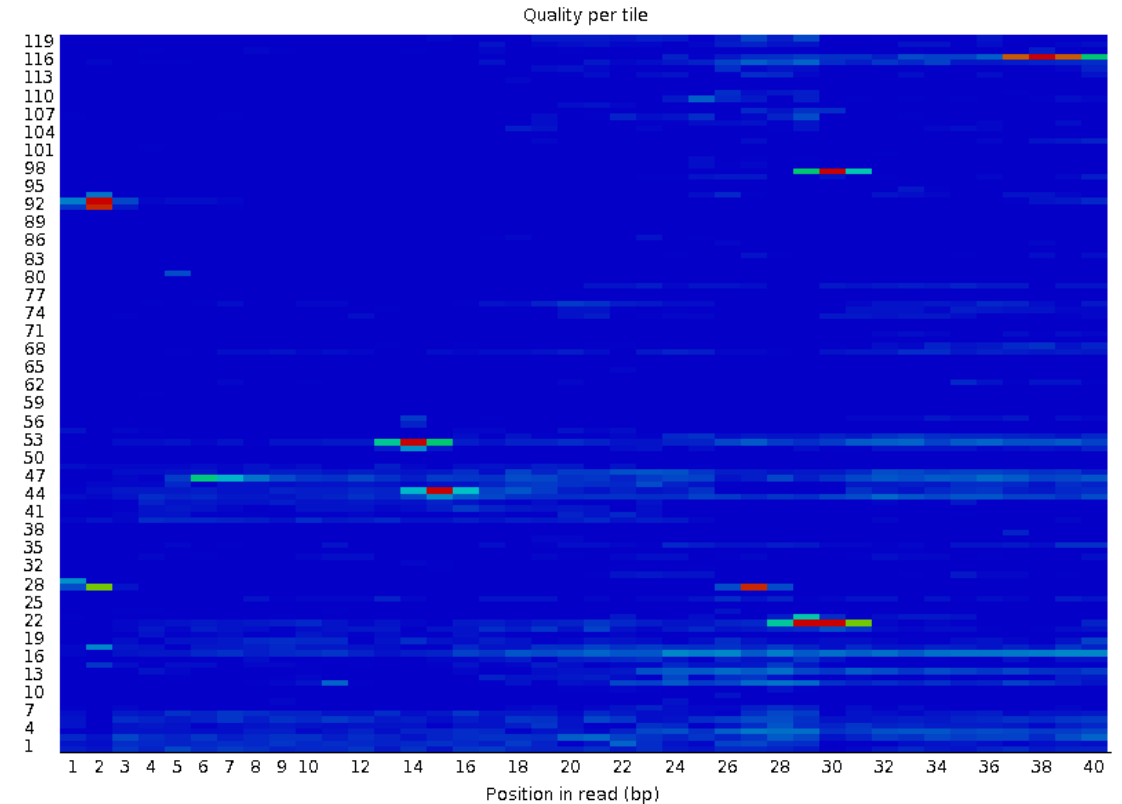
Good Quality

✔ Per tile sequence quality



Bad Quality

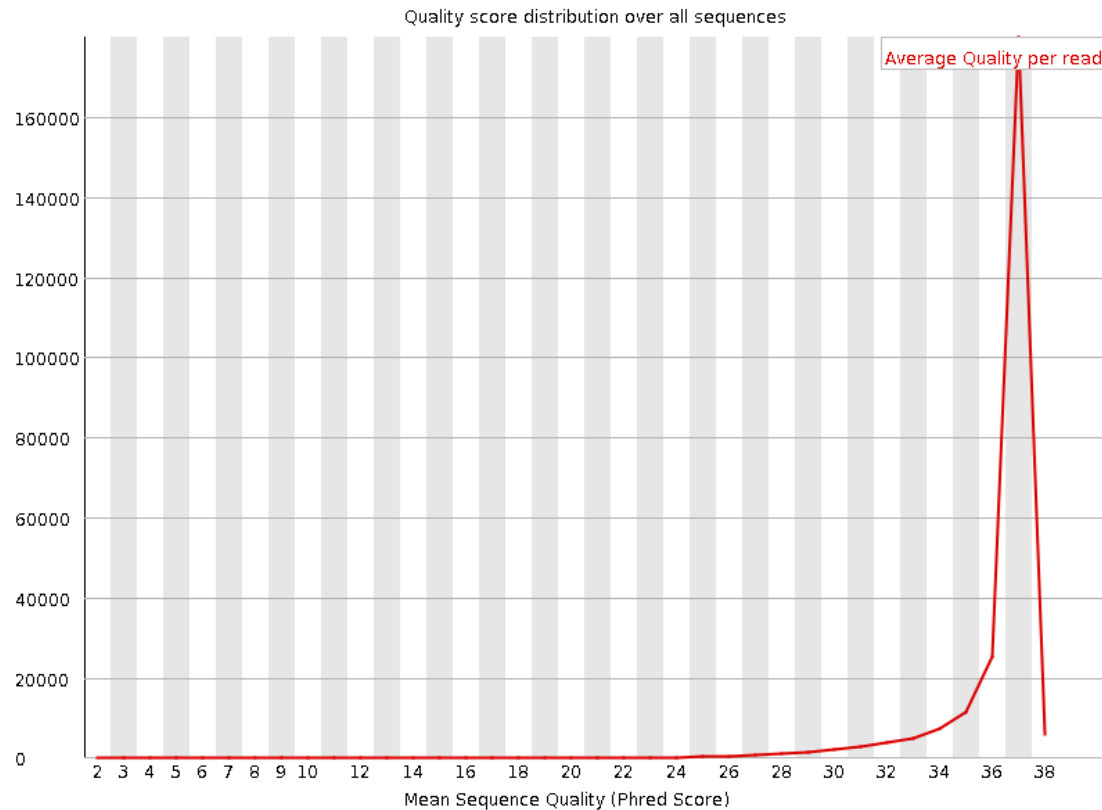
✘ Per tile sequence quality



Per sequence quality scores

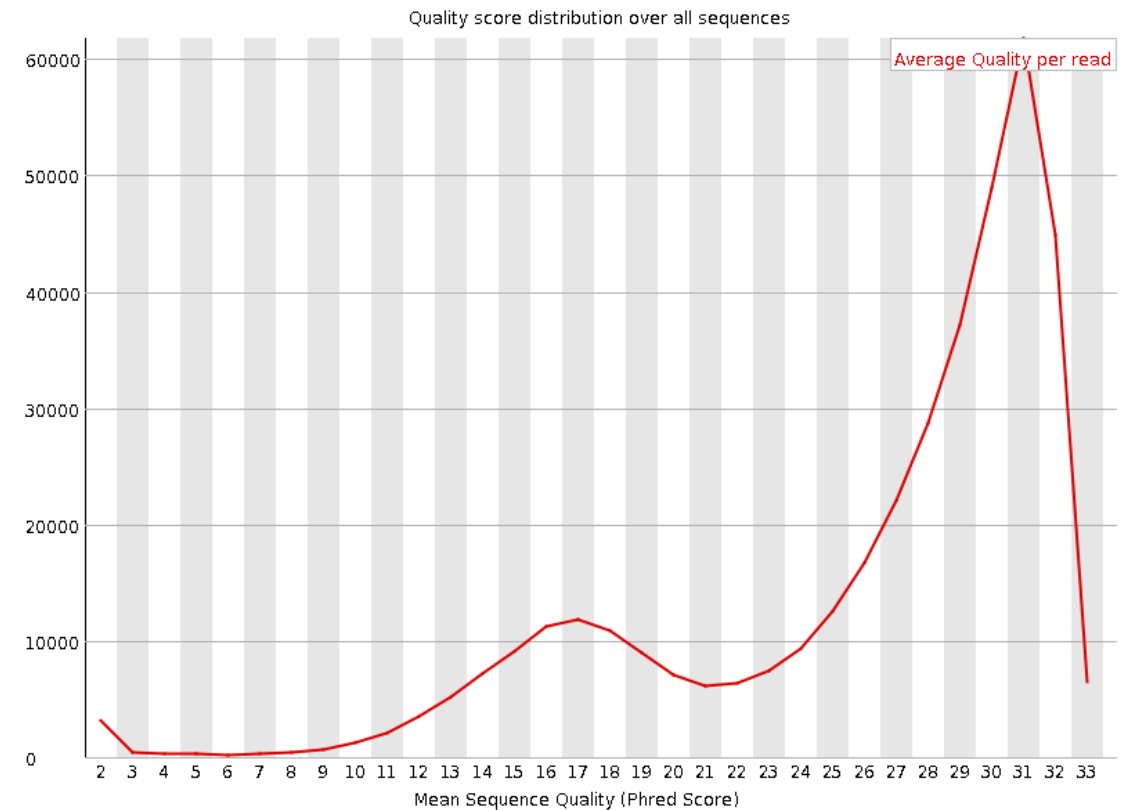
Good Quality

✔ Per sequence quality scores



Bad Quality

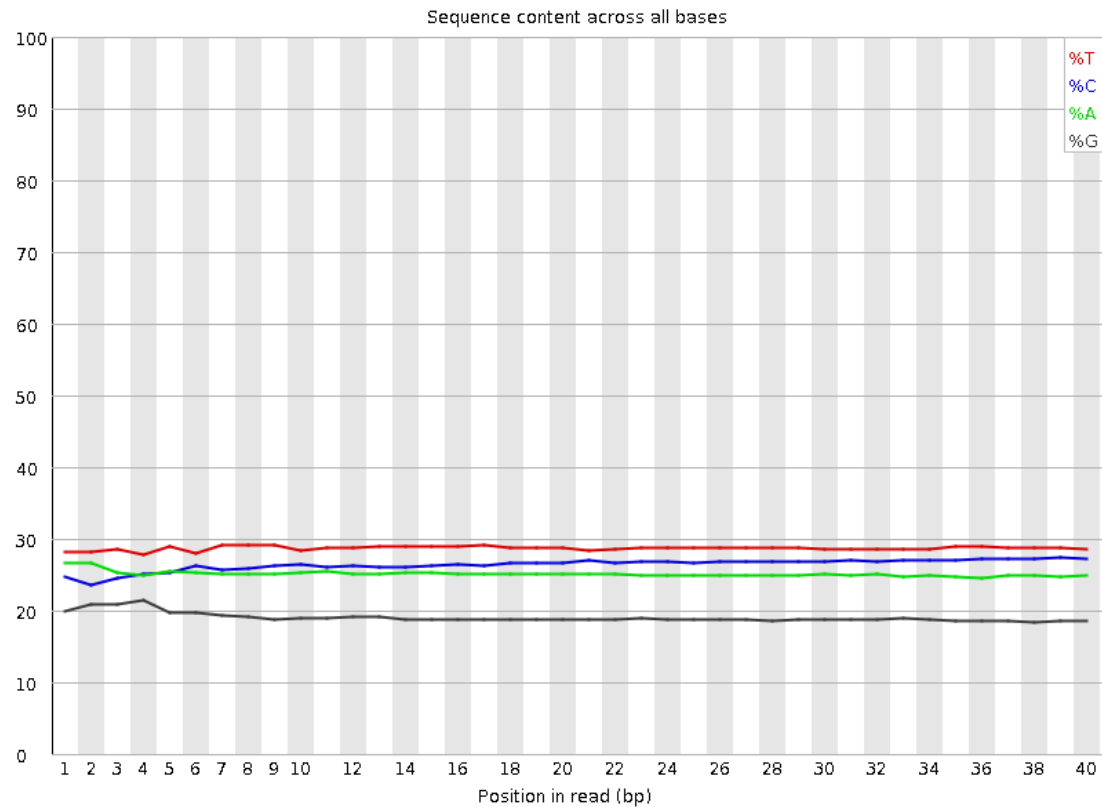
✔ Per sequence quality scores



Per base sequence content

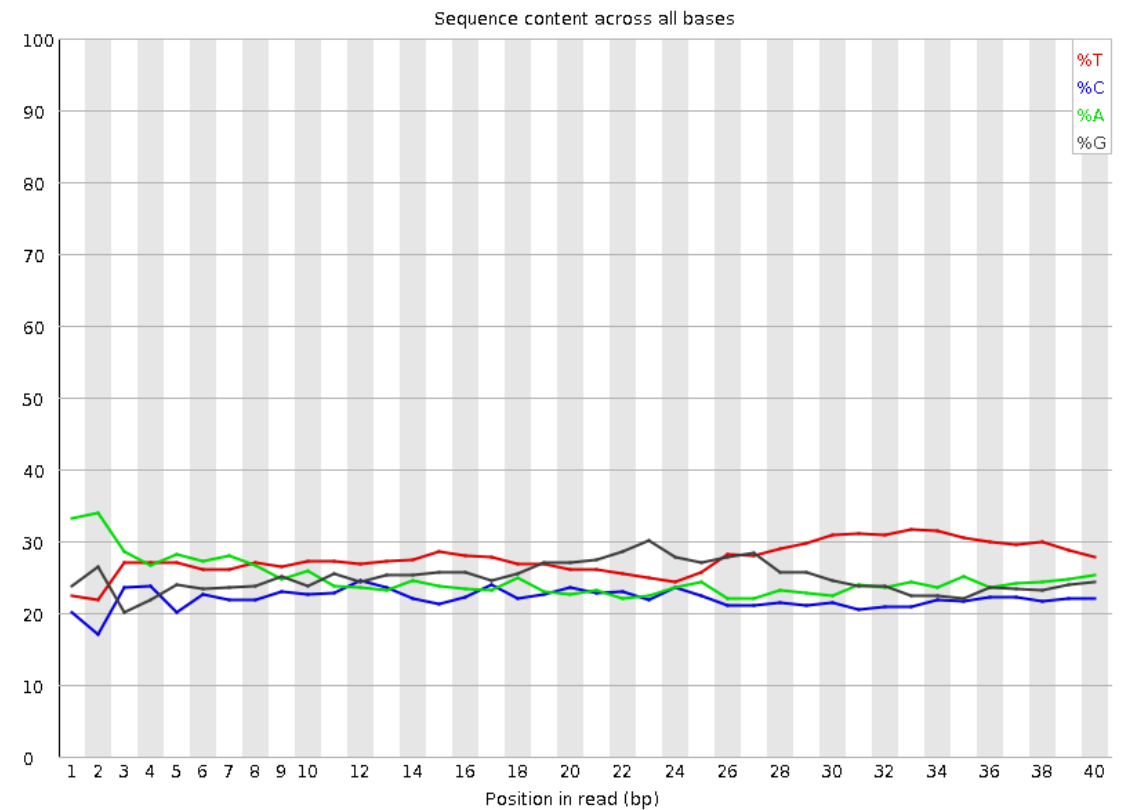
Good Quality

✔ Per base sequence content



Bad Quality

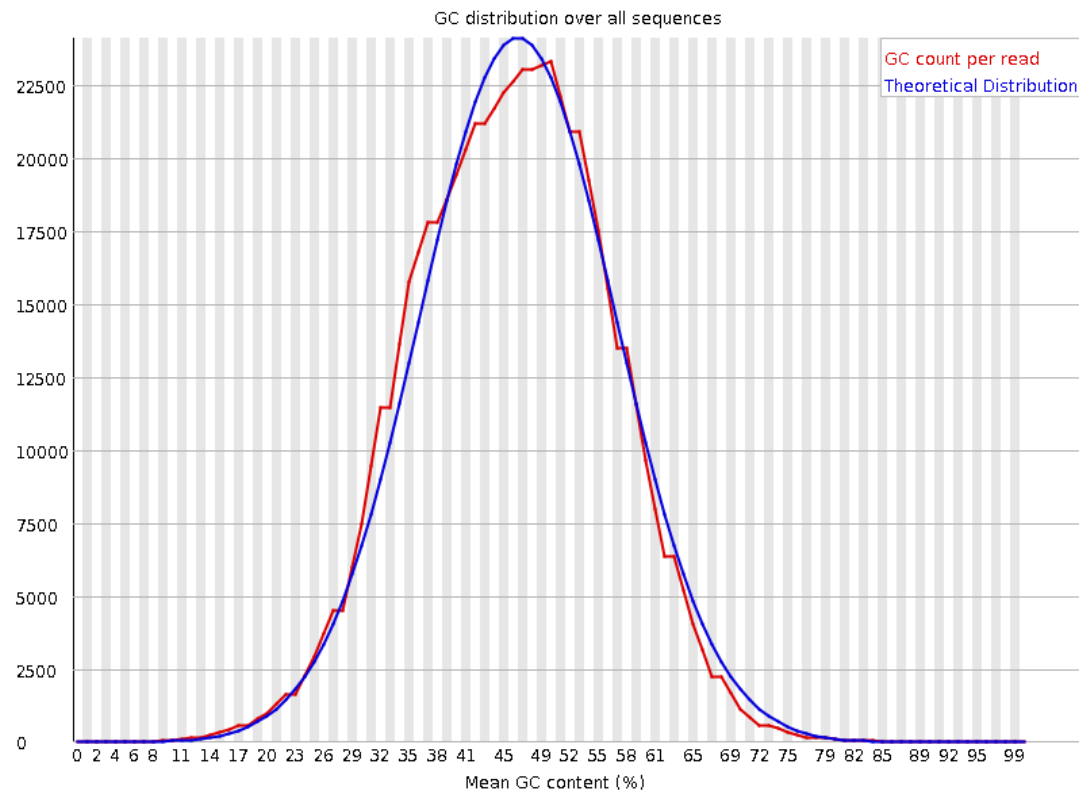
⚠ Per base sequence content



Per sequence GC content

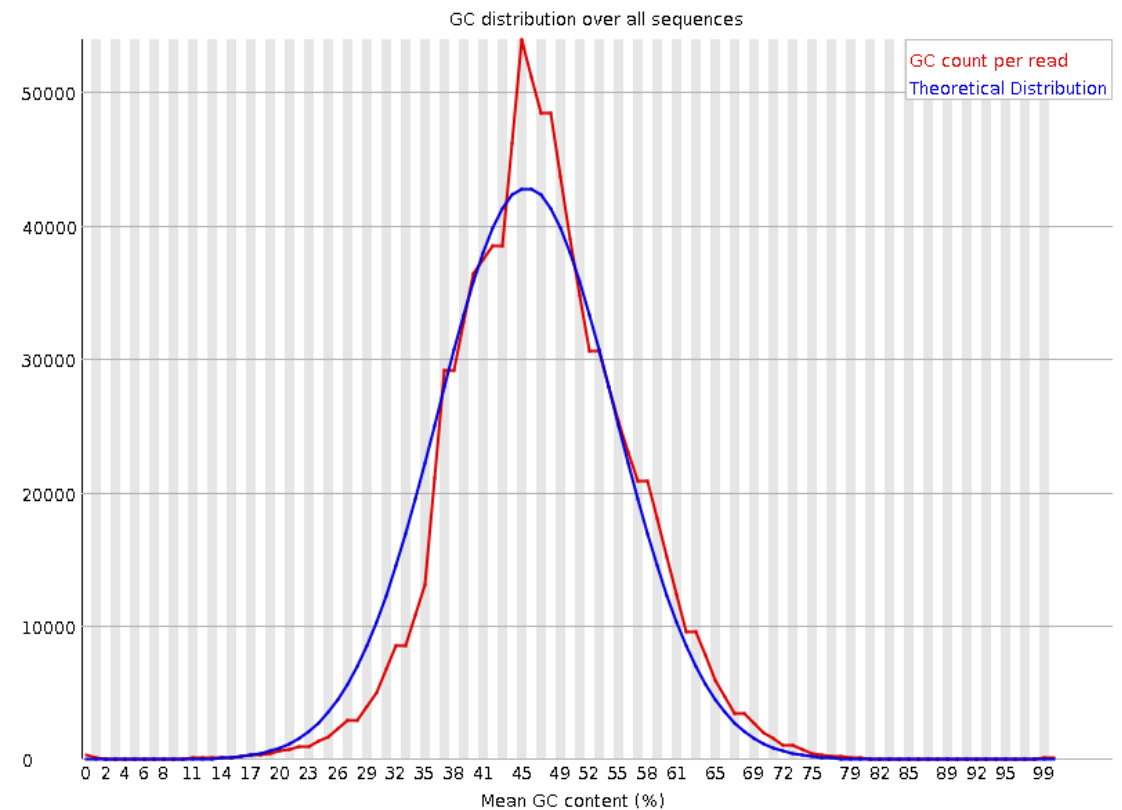
Good Quality

✔ Per sequence GC content



Bad Quality

⚠ Per sequence GC content

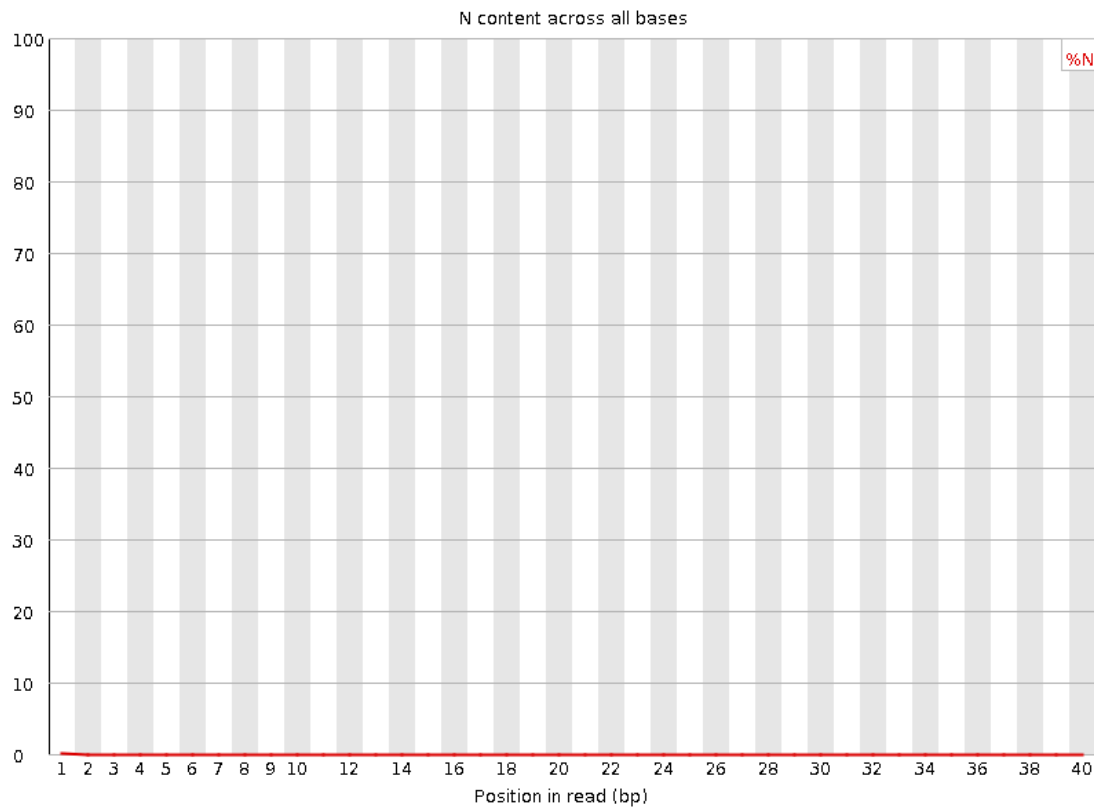


Per base sequence quality

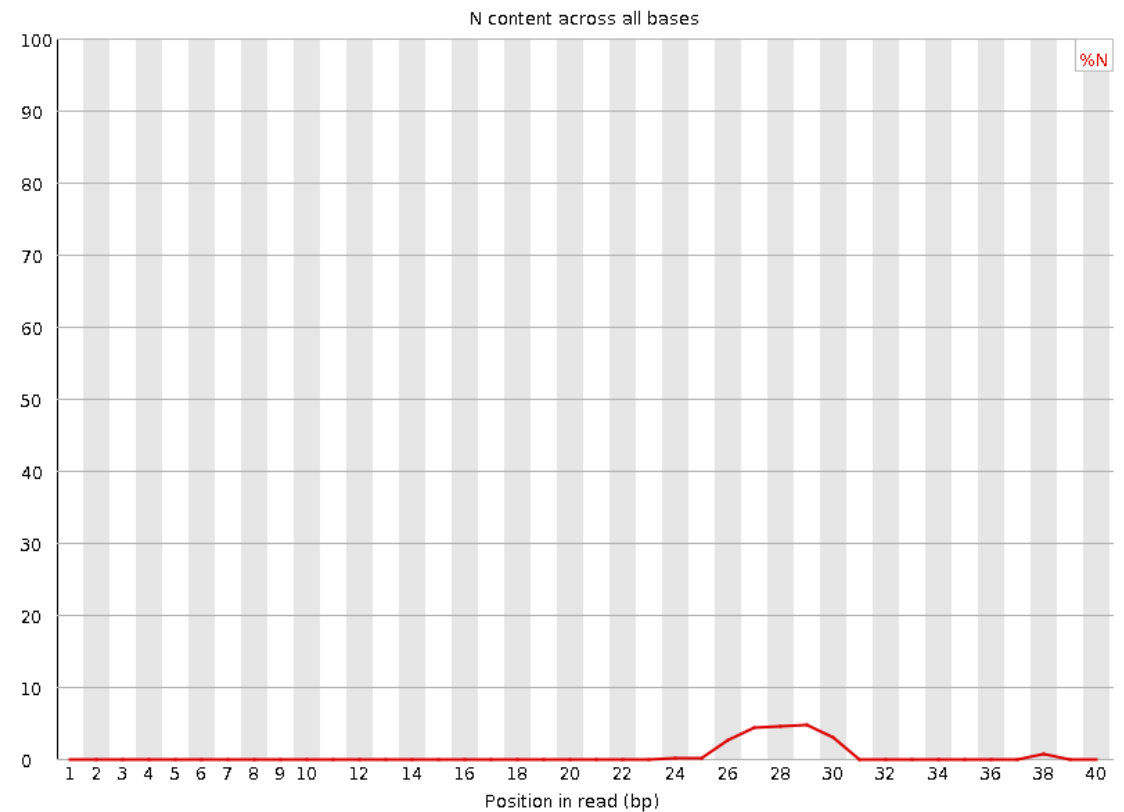
Good Quality

Bad Quality

✔ Per base N content



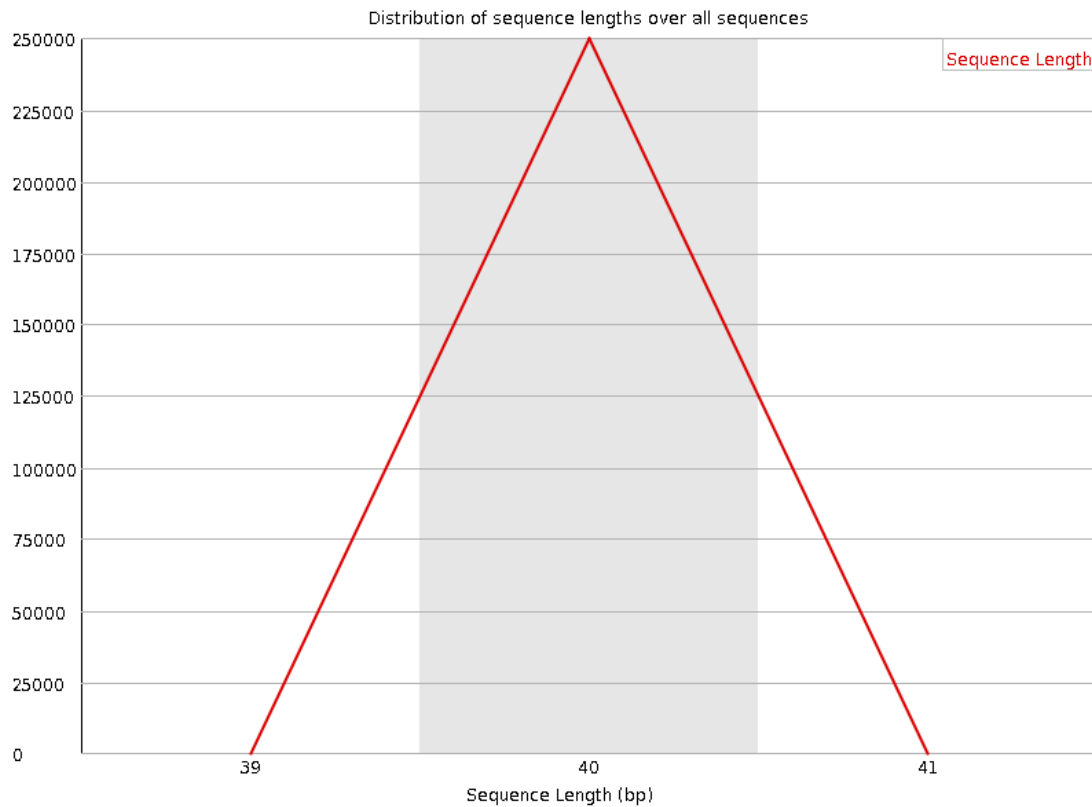
✔ Per base N content



Per base sequence quality

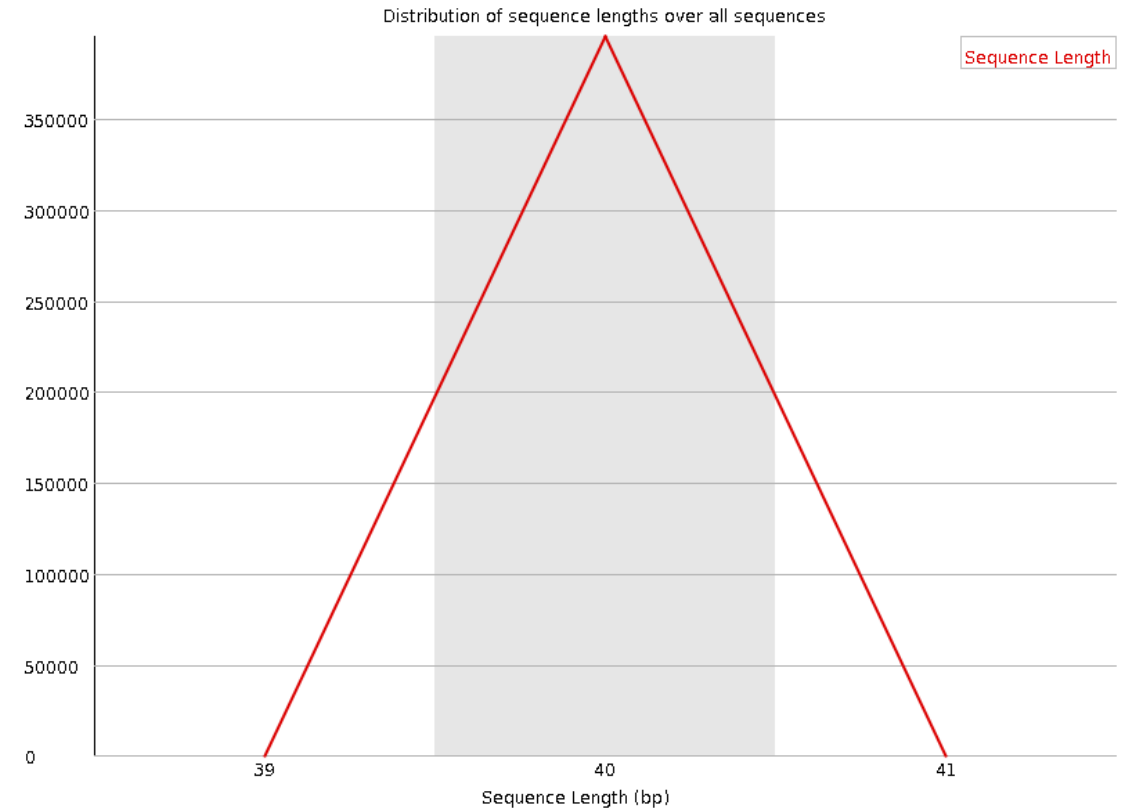
Good Quality

✔ Sequence Length Distribution



Bad Quality

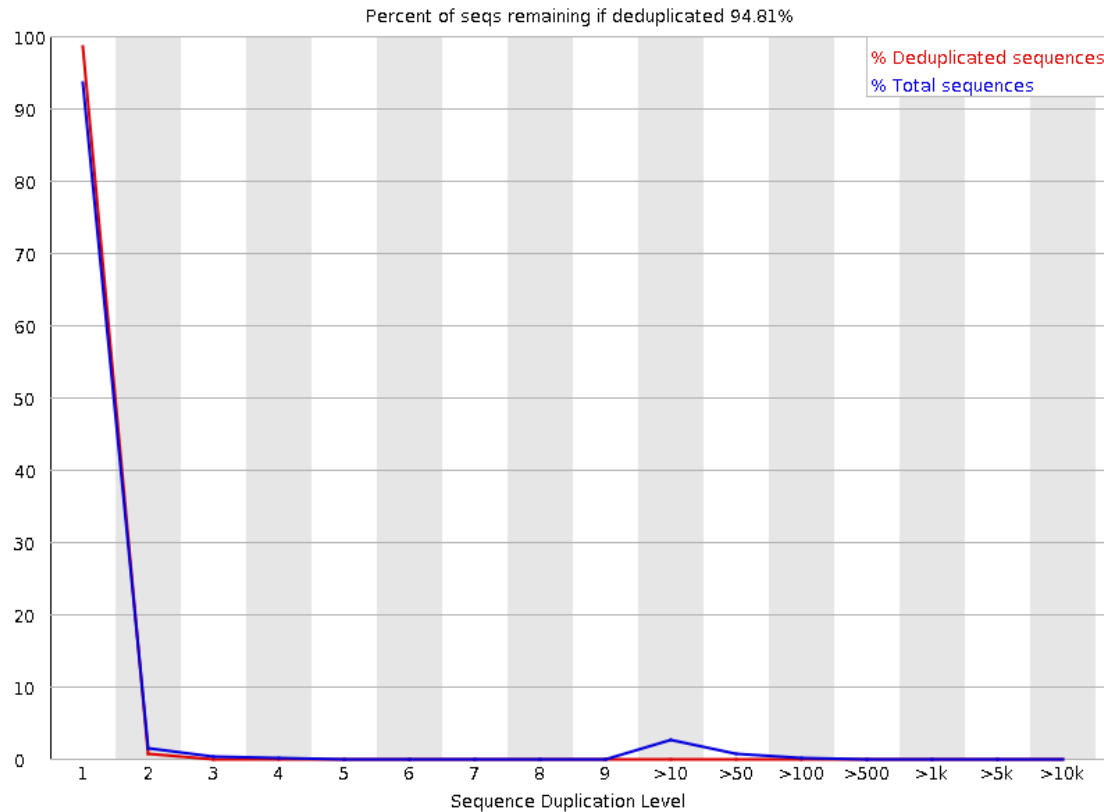
✔ Sequence Length Distribution



Sequence Duplication Levels

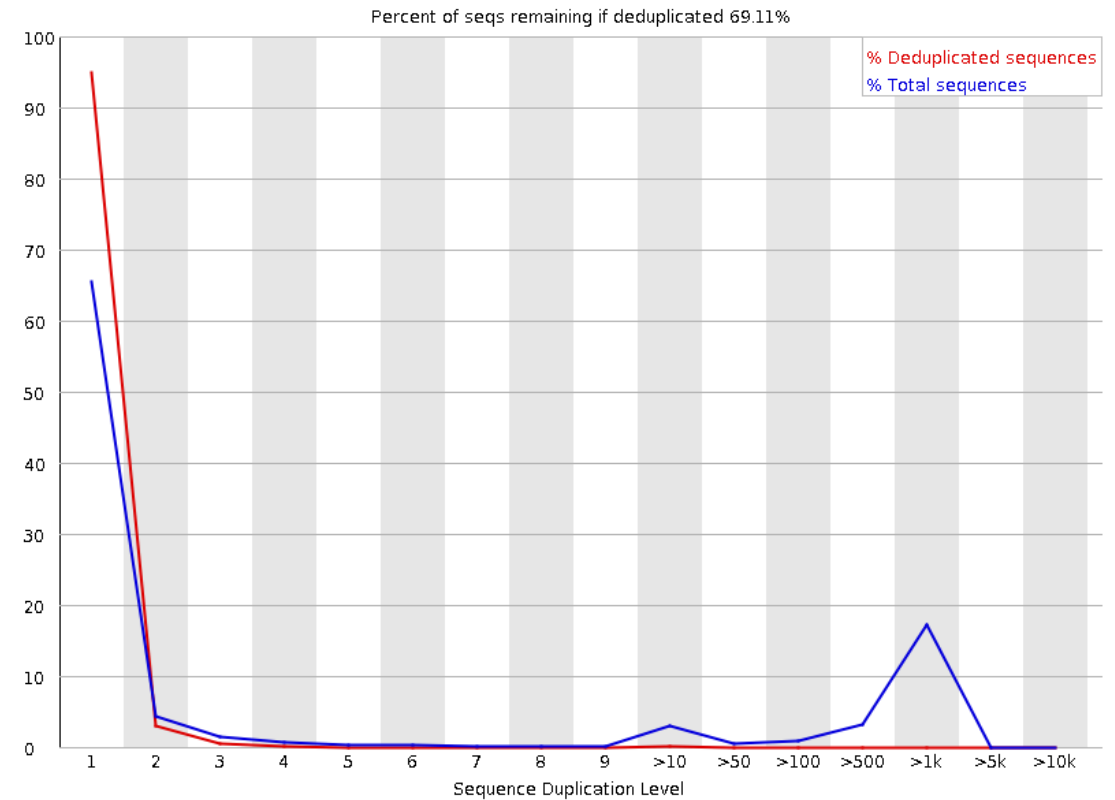
Good Quality

✔ Sequence Duplication Levels



Bad Quality

! Sequence Duplication Levels



Overrepresented sequences

Good Quality



Overrepresented sequences

No overrepresented sequences

Bad Quality

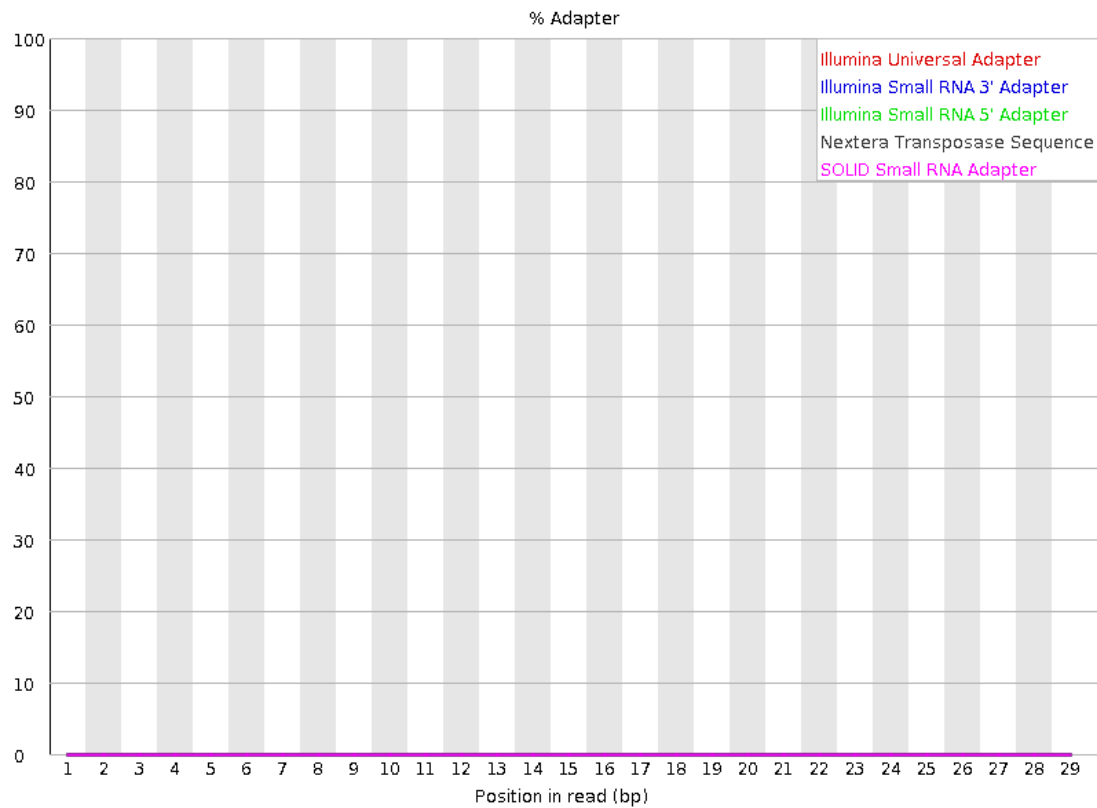
Overrepresented sequences

Sequence	Count	Percentage	Possible Source
AGAGTTTATCGCTTCACAGCCGAGAGTTAACACTTC	2965	0.5224039181558763	No Hit
GATGGCGTATCAACTCCAGGTTTTATCGCTCATG	2947	0.5178592762542754	No Hit
ATGGCGTATCAACTCCAGGTTTTATCGCTTCATGA	2914	0.5098019327480071	No Hit
CGATAAAATGATGGCGTATCCAACTCCAGGTTTTAT	1913	0.4839509420979134	No Hit
GTATCAACTCCAGGTTTTATCGCTTCACAGCCGGA	1879	0.47534961850640066	No Hit
AAAAATGATGGCGTATCAACTCCAGGTTTTATCGCT	1846	0.4670012750197325	No Hit
TGATGGCGTATCAACTCCAGGTTTTATCGCTTCAT	1841	0.46573637449150995	No Hit
AACCTCCAGGTTTTATCGCTTCATGACCGAGATTAA	1836	0.46447147396328753	No Hit
GATAAAATGATGGCGTATCCAACTCCAGGTTTTATC	1831	0.4632065734350651	No Hit
AAATGATGGCGTATCAACTCCAGGTTTTATCGCTTC	1779	0.45005160794155147	No Hit
ATGATGGCGTATCAACTCCAGGTTTTATCGCTTCA	1779	0.45005160794155147	No Hit
AAATGATGGCGTATCCAACTCCAGGTTTTATCGCTTC	1760	0.4452449859343061	No Hit
AAAAATGATGGCGTATCCAACTCCAGGTTTTATCGCT	1729	0.4374026026593269	No Hit
CGTATCCAACTCCAGGTTTTATCGCTTCATGACCGA	1713	0.4332949299691494	No Hit
ATCCAACTCCAGGTTTTATCGCTTCATGACCGAGA	1708	0.4320902044079253	No Hit
CAGAGTTTTATCGCTTCATGACCGAGTTAACACTTC	1684	0.42601848790532476	No Hit
TCCAGTTTTATCGCTTCATGACCGAGTTAACACTTC	1668	0.4219708162150128	No Hit
CAACTCCAGGTTTTATCGCTTCATGACCGAGATTAA	1668	0.4219708162150128	No Hit
TATCCAACTCCAGGTTTTATCGCTTCATGACCGAGA	1630	0.4123575722005221	No Hit
GTATGGAGCGATAAACTCCAGGTTGGATACGCCAA	1620	0.40982777114407726	No Hit
AACTTCGCGTATGAGGATAAACTCCAGGTTGG	1616	0.4088158507214993	No Hit
GCAGAGTTTTATCGCTTCATGACCGAGATTAACTTC	1580	0.39970856691829754	No Hit
TGGCGTATCAACTCCAGGTTTTATCGCTTCATGACG	1569	0.3969257857562082	No Hit
GGCGTATCAACTCCAGGTTTTATCGCTTCATGACG	1542	0.39009532290380683	No Hit
ATAAAATGATGGCGTATCAACTCCAGGTTTTATCG	1481	0.37466353645949285	No Hit
ACTCCAGGTTTTATCGCTTCACAGCCGAGAGTTAAC	1479	0.37415757624820384	No Hit
ATGAAAGCGATAAACTCCAGGTTGGATACGCCAA	1452	0.3673271133988026	No Hit
GATAAACTCCAGGTTGGATACGCCAACTTTATCG	1420	0.35923175001517876	No Hit
CGTATGAGCGATAAACTCCAGGTTGGATACGCCAA	1412	0.3572079091700229	No Hit
ACTTCGCGTATGAGGATAAACTCCAGGTTGGA	1368	0.34607678452166524	No Hit
TAACTTCGCGTATGAGGATAAACTCCAGGTTGG	1363	0.34481180399344276	No Hit
CATGAGCGATAAACTCCAGGTTGGATACGCCAACT	1333	0.337222480824108	No Hit
CGATAAACTCCAGGTTGGATACGCCAACTTTAT	1304	0.32988605776041774	No Hit
TAAAAATGATGGCGTATCCAACTCCAGGTTTTATCG	1277	0.32305559490801644	No Hit
GGTATCCAACTCCAGGTTTTATCGCTTCATGACCGA	1262	0.31926089332334906	No Hit
TGGCGTATGAGGATAAACTCCAGGTTGGATACCG	1233	0.3119244702596588	No Hit
GGAGCGATAAACTCCAGGTTGGATACGCCAACTATT	1182	0.2990224848717897	No Hit
AGCGATAAACTCCAGGTTGGATACGCCAACTATT	1136	0.2873854000121431	No Hit
ACTTCGAGTTTTATCGCTTCACAGCCGAGAGTTAAC	1133	0.28646454969520956	No Hit
AAAACTCCAGGTTGGATACGCCAACTTTATCGGAA	1131	0.28612049494839206	No Hit
AAAACTCCAGGTTGGATACGCCAACTTTATCGAA	1129	0.2856145392726316	No Hit
AGCGATAAACTCCAGGTTGGATACGCCAACTATT	1113	0.2815668575823197	No Hit
ATAAACTCCAGGTTGGATACGCCAACTATTATCG	1111	0.28106089737103074	No Hit
AACTTCGAGTTGGATACGCCAACTATTATCGAAGC	1083	0.273977454412985	No Hit
CTCCAGGTTTTATCGCTTCATGACCGAGATTAACT	1055	0.2668940114549392	No Hit
TTCGCGTATGAGGATAAACTCCAGGTTGGATA	947	0.23957216004533402	No Hit
TGGAAGCGATAAACTCCAGGTTGGATACGCCAACT	946	0.23931917993968954	No Hit
TAAAACTCCAGGTTGGATACGCCAACTTTATCGA	912	0.2307178563477768	No Hit
GAGCGATAAACTCCAGGTTGGATACGCCAACTATT	898	0.224646333812309	No Hit
GGTATGAGGATAAACTCCAGGTTGGATACGCC	805	0.20364898504381615	No Hit
CGGATAAACTCCAGGTTGGATACGCCAACTTTTAA	785	0.19898938293992632	No Hit
TTCGCGTATCAACTCCAGGTTTTATCGCTTCATGAC	784	0.198336409825818	No Hit
CTTCGCGTATGAGGATAAACTCCAGGTTGGAT	762	0.192770840501103	No Hit
TCCAACTCCAGGTTTTATCGCTTCATGACCGAGATT	752	0.1902410394445806	No Hit
CCAACTCCAGGTTTTATCGCTTCATGACCGAGATT	744	0.1882171985950212	No Hit
TGATGAGCGATAAACTCCAGGTTGGATACGCCAACT	665	0.1682317025358726	No Hit
TTCGCGTATGAGGATAAACTCCAGGTTGGATAC	627	0.15861852623909656	No Hit
CTCCAGGTTTTATCGCTTCATGACCGAGATTAACT	613	0.15507680476007366	No Hit
CGGTTAGCGGATAAGCGAGATCCGAGGCGGCTCCAG	599	0.15153508328105078	Illumina Paired End PCR Primer 2 (96% over 25bp)
TCTCCAGGTTGGATACGCCAACTATTATCGAAGCGG	585	0.1479933618020279	No Hit
CGCTTAAAGCTACCAAGTATATGCGTGGGGGGTTTT	552	0.13964501831575965	No Hit
CTTCAGGTTGGATACGCCAACTTTTATCGAAGCGG	532	0.1345854162028698	No Hit
TTCGCGTATGAGGATAAACTCCAGGTTGGATACG	515	0.13028475440691342	No Hit
CTCCAGGTTGGATACGCCAACTTTTATCGAAGCGG	505	0.12775495335044852	No Hit
CGCTTAAAGCTACCAAGTATATGCGTGGGGGGTTTT	411	0.10397482341988626	No Hit

Adapter Content

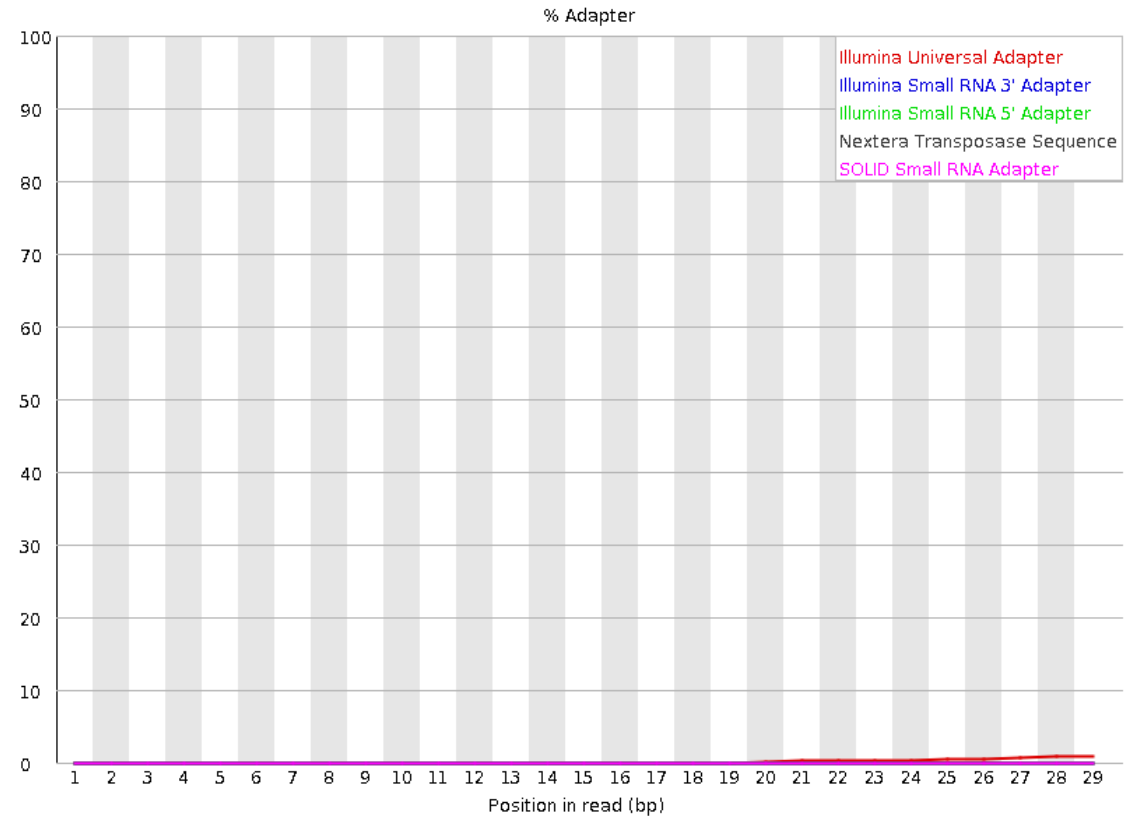
Good Quality

✓ Adapter Content



Bad Quality

✓ Adapter Content



Run FastQC

1. Go to the RNA-seq data directory
2. Make a directory to put the FastQC reports into, `fastqc`
3. Run `fastqc` on the samples

```
for i in *.fastq.gz; do fastqc $i -o fastqc/;  
done
```

Trim Bad Quality Sequences

What is trimming and why do it?

What is trimming and why do it?

- Trimming removes sequencing adapters, bad quality sequences, and/or other biased sequence information

What is trimming and why do it?

- Trimming removes sequencing adapters, bad quality sequences, and/or other biased sequence information
- Why is that important?
 - Helps prevent incorrect base calls by removing poor quality information
 - Increases speed and accuracy of alignment by removing artificial sequences and low quality sequences

What is trimming and why do it?

- Trimming removes sequencing adapters, bad quality sequences, and/or other biased sequence information
- Why is that important?
 - Helps prevent incorrect base calls by removing poor quality information
 - Increases speed and accuracy of alignment by removing artificial sequences and low quality sequences
- Trimming does two complementary things:
 1. Removes any sequence information that comes from library preparation or sequencing
 2. Removes low quality bases / low quality reads

Trim Sequences

1. Go back up to the rnaseq directory
2. Make a folder to put the analysis results in, `analysis`
3. Make a folder inside the analysis folder to put the trimmed reads in, `analysis/01_trim`

Trim Sequences

```
for i in rnaseq_data/*R1.fastq.gz;
do trim_galore
    --paired
    --fastqc
    --illumina
    --output analysis/01_trim/
    --retain_unpaired
    $i
    ${i/R1/R2};
done
```

Trim Sequences

```
for i in rnaseq_data/*R1.fastq.gz;  
do trim_galore  
    --paired  
    --fastqc  
    --illumina  
    --output analysis/01_trim/  
    --retain_unpaired  
    $i  
    ${i/R1/R2};  
  
done
```



loop condition

Trim Sequences

```
for i in rnaseq_data/*R1.fastq.gz;  
do trim_galore  
    --paired  
    --fastqc  
    --illumina  
    --output analysis/01_trim/  
    --retain_unpaired  
    $i  
    ${i/R1/R2};  
  
done
```

loop condition

call the program

Trim Sequences

```
for i in rnaseq_data/*R1.fastq.gz;  
do trim_galore  
  --paired  
  --fastqc  
  --illumina  
  --output analysis/01_trim/  
  --retain_unpaired  
  $i  
  ${i/R1/R2};  
  
done
```

loop condition

call the program

reads are paired-end

Trim Sequences

```
for i in rnaseq_data/*R1.fastq.gz;
do trim_galore
  --paired
  --fastqc
  --illumina
  --output analysis/01_trim/
  --retain_unpaired
  $i
  ${i/R1/R2};
done
```

Annotations:

- loop condition
- call the program
- reads are paired-end
- run FastQC again after trimming

Trim Sequences

```
for i in rnaseq_data/*R1.fastq.gz;
do trim_galore
  --paired
  --fastqc
  --illumina
  --output analysis/01_trim/
  --retain_unpaired
  $i
  ${i/R1/R2};
done
```

Annotations for the script:

- loop condition
- call the program
- reads are paired-end
- run FastQC again after trimming
- trim Illumina adapters

Trim Sequences

```
for i in rnaseq_data/*R1.fastq.gz;
do trim_galore
  --paired
  --fastqc
  --illumina
  --output analysis/01_trim/
  --retain_unpaired
  $i
  ${i/R1/R2};
done
```

Annotations for the script:

- loop condition
- call the program
- reads are paired-end
- run FastQC again after trimming
- trim Illumina adapters
- output goes here

Trim Sequences

```
for i in rnaseq_data/*R1.fastq.gz;
do trim_galore
  --paired
  --fastqc
  --illumina
  --output analysis/01_trim/
  --retain_unpaired
  $i
  ${i/R1/R2};
done
```

Annotations for the script:

- loop condition
- call the program
- reads are paired-end
- run FastQC again after trimming
- trim Illumina adapters
- output goes here
- keep reads where one mate fails trimming but the other doesn't

Trim Sequences

```
for i in rnaseq_data/*R1.fastq.gz;
do trim_galore
  --paired
  --fastqc
  --illumina
  --output analysis/01_trim/
  --retain_unpaired
  $i
  ${i/R1/R2};
done
```

Annotations:

- loop condition
- call the program
- reads are paired-end
- run FastQC again after trimming
- trim Illumina adapters
- output goes here
- keep reads where one mate fails trimming but the other doesn't
- read files

Trim Sequences

```
for i in rnaseq_data/*R1.fastq.gz;
do trim_galore
--paired
--fastqc
--illumina
--output analysis/01_trim/
--retain_unpaired
$i
${i/R1/R2};
done
```

By default bases quality less than 20 will be trimmed and if the read falls below 20 bp, it will be discarded

loop condition

call the program

reads are paired-end

run FastQC again after trimming

trim Illumina adapters

output goes here

keep reads where one mate fails trimming but the other doesn't

read files

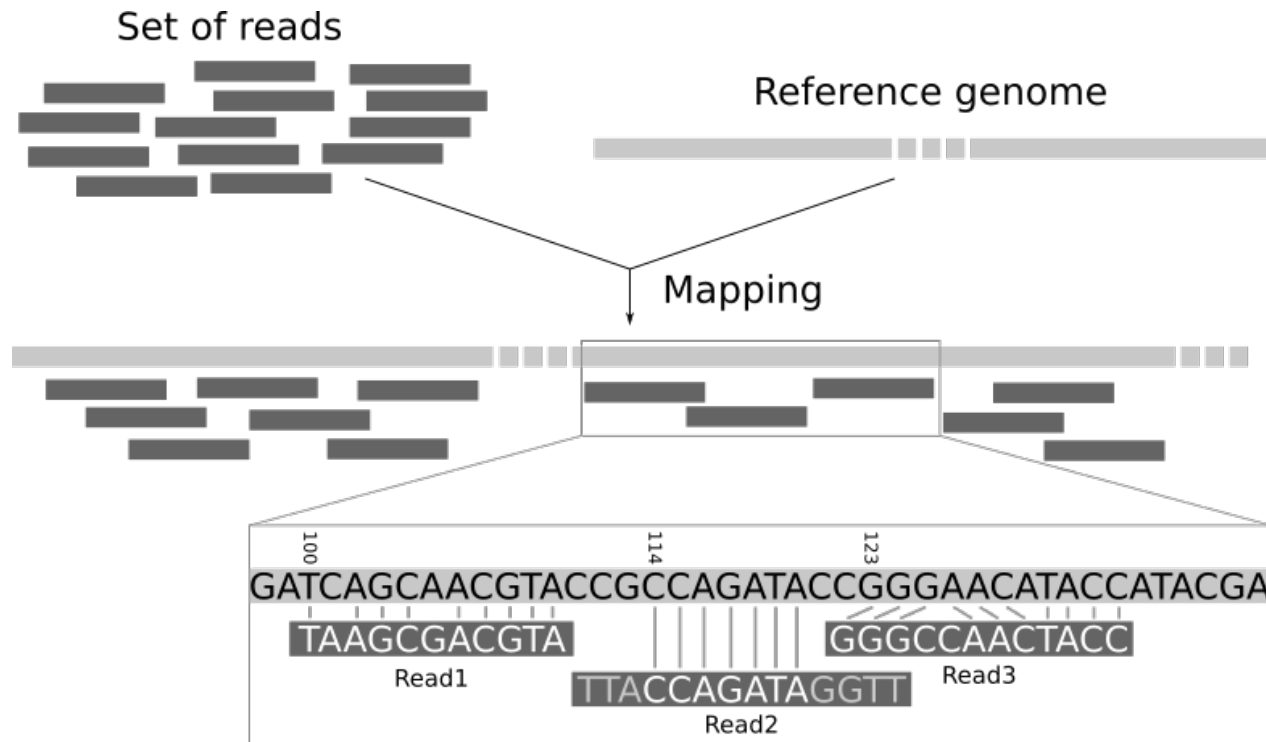
Trim Sequences

```
for i in rnaseq_data/*R1.fastq.gz; do trim_galore  
--paired --fastqc --illumina --output analysis/01_trim/  
--retain_unpaired $i ${i/R1/R2}; done
```

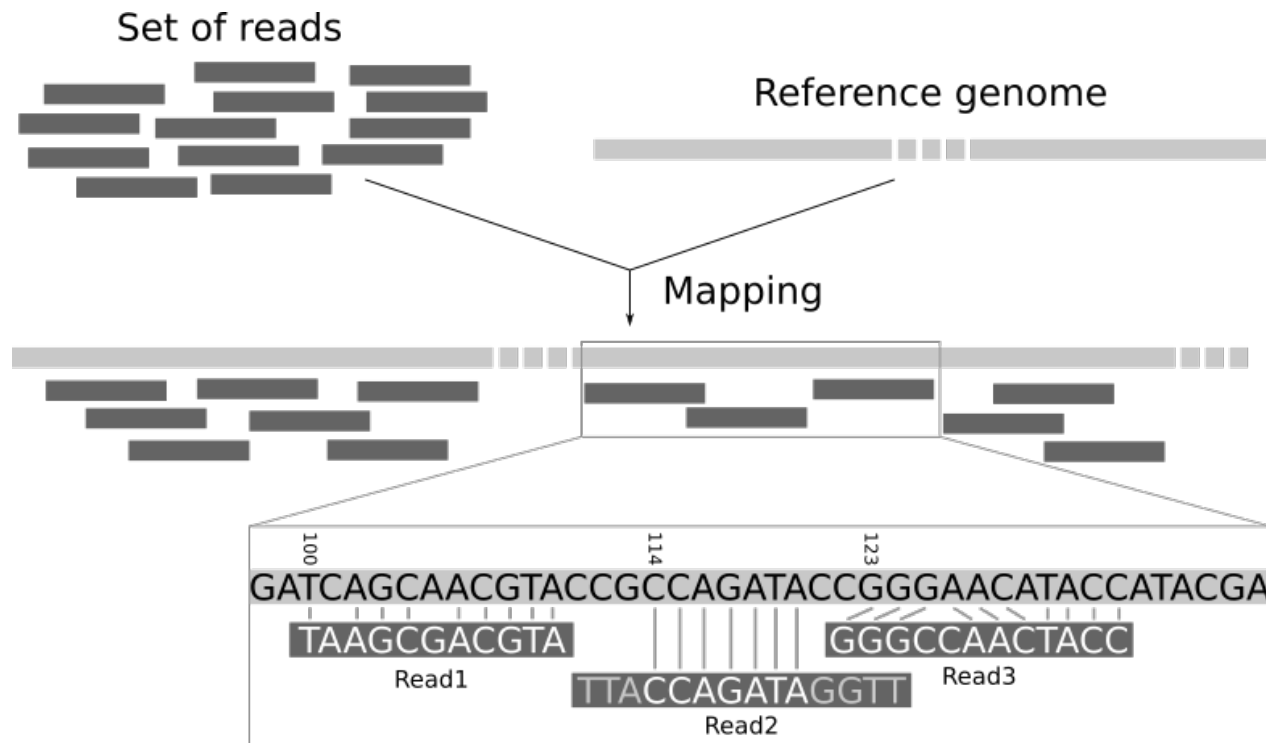
Align

How does aligning work?

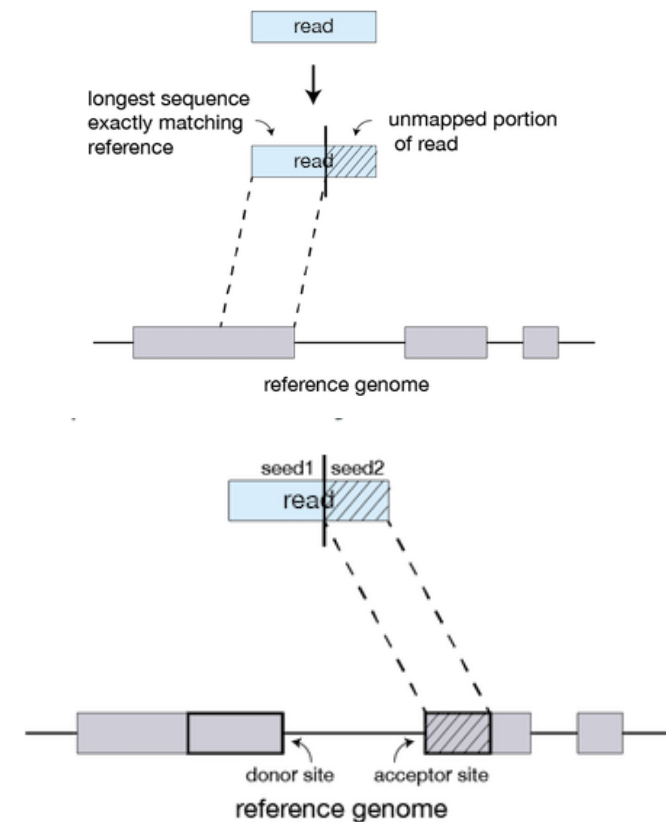
How does aligning work?



How does aligning work?



STAR (Spliced Transcripts Alignment to a Reference)



Align Sequences

1. Make a folder inside the analysis folder to put the aligned reads in, `analysis/02_align`
2. Change to the trimmed reads folder `analysis/01_trim`

Align Sequences

```
for i in *val_1.fq.gz;
do STAR
    --genomeDir /mnt/data/gdata/human \
                /hg38/chr21/STAR_index
    --readFilesIn $i ${i/R1_val_1/R2_val_2}
    --readFilesCommand zcat
    --outFileNamePrefix ../02_align/${i/R1*/}
    --outSAMtype BAM SortedByCoordinate;
done
```

Align Sequences

```
for i in *val_1.fq.gz; ← loop condition
do STAR
    --genomeDir /mnt/data/gdata/human \
                /hg38/chr21/STAR_index
    --readFilesIn $i ${i/R1_val_1/R2_val_2}
    --readFilesCommand zcat
    --outFileNamePrefix ../02_align/${i/R1*/}
    --outSAMtype BAM SortedByCoordinate;
done
```

Align Sequences

```
for i in *val_1.fq.gz;
do STAR
    --genomeDir /mnt/data/gdata/human \
                /hg38/chr21/STAR_index
    --readFilesIn $i ${i/R1_val_1/R2_val_2}
    --readFilesCommand zcat
    --outFileNamePrefix ../02_align/${i/R1*/}
    --outSAMtype BAM SortedByCoordinate;
done
```

Diagram illustrating the loop structure and the call to the aligner (STAR) in the provided code snippet. The code is annotated with red boxes and arrows:

- The `for i in *val_1.fq.gz;` line is annotated as the **loop condition**.
- The `do STAR` line is annotated as the **call aligner**.

Align Sequences

```
for i in *val_1.fq.gz;
```

loop condition

```
do STAR
```

call aligner

path to reference
genome

```
--genomeDir /mnt/data/gdata/human \
              /hg38/chr21/STAR_index
--readFilesIn $i ${i/R1_val_1/R2_val_2}
--readFilesCommand zcat
--outFileNamePrefix ../02_align/${i/R1*/}
--outSAMtype BAM SortedByCoordinate;
```

```
done
```

Align Sequences

```
for i in *val_1.fq.gz;
```

loop condition

```
do STAR
```

call aligner

path to reference
genome

```
--genomeDir /mnt/data/gdata/human \  
             /hg38/chr21/STAR_index
```

trimmed read files

```
--readFilesIn $i ${i/R1_val_1/R2_val_2}
```

```
--readFilesCommand zcat
```

```
--outFileNamePrefix ../02_align/${i/R1*/}
```

```
--outSAMtype BAM SortedByCoordinate;
```

```
done
```

Align Sequences

```
for i in *val_1.fq.gz;
```

loop condition

```
do STAR
```

call aligner

path to reference
genome

```
--genomeDir /mnt/data/gdata/human \  
             /hg38/chr21/STAR_index
```

trimmed read files

```
--readFilesIn $i ${i/R1_val_1/R2_val_2}
```

zipped files

```
--readFilesCommand zcat
```

```
--outFileNamePrefix ../02_align/${i/R1*/}
```

```
--outSAMtype BAM SortedByCoordinate;
```

```
done
```

Align Sequences

```
for i in *val_1.fq.gz;
```

loop condition

```
do STAR
```

call aligner

path to reference
genome

```
--genomeDir /mnt/data/gdata/human \  
             /hg38/chr21/STAR_index
```

trimmed read files

```
--readFilesIn $i ${i/R1_val_1/R2_val_2}
```

zipped files

```
--readFilesCommand zcat
```

write the files here

```
--outFileNamePrefix ../02_align/${i/R1*/}
```

```
--outSAMtype BAM SortedByCoordinate;
```

```
done
```

Align Sequences

```
for i in *val_1.fq.gz;
```

loop condition

```
do STAR
```

call aligner

path to reference
genome

```
--genomeDir /mnt/data/gdata/human \  
             /hg38/chr21/STAR_index
```

trimmed read files

```
--readFilesIn $i ${i/R1_val_1/R2_val_2}
```

zipped files

```
--readFilesCommand zcat
```

write the files here

```
--outFileNamePrefix ../02_align/${i/R1*/}
```

write a sorted BAM

```
--outSAMtype BAM SortedByCoordinate;
```

```
done
```


Align Sequences

```
for i in *val_1.fq.gz; do STAR --genomeDir  
/mnt/data/gdata/human/hg38/chr21/STAR_index --  
readFilesIn $i ${i/R1_val_1/R2_val_2} --  
readFilesCommand zcat --outFileNamePrefix  
../02_align/${i/R1*/} --outSAMtype BAM  
SortedByCoordinate; done
```

Count Features

What do you mean by count features?

- We're going to count genes, but you could also count:

- transcripts
- non-coding RNA

- Need an annotation file for whatever feature you want to count

<u>Col 1</u>	<u>Col 2</u>	<u>Col 3</u>	<u>Col 4</u>	<u>Col 5</u>	<u>Col 6</u>	<u>Col 7</u>	<u>Col 8</u>	<u>Col 9</u>
chr21	HAVANA	transcript	10862622	10863067	.	+	.	gene_id "ENSG00000169..
chr21	HAVANA	exon	10862622	10862667	.	+	.	gene_id "ENSG00000169..
chr21	HAVANA	CDS	10862622	10862667	.	+	0	gene_id "ENSG00000169..
chr21	HAVANA	start_codon	10862622	10862624	.	+	0	gene_id "ENSG00000169..
chr21	HAVANA	exon	10862751	10863067	.	+	.	gene_id "ENSG00000169..
chr21	HAVANA	CDS	10862751	10863064	.	+	2	gene_id "ENSG00000169..
chr21	HAVANA	stop_codon	10863065	10863067	.	+	0	gene_id "ENSG00000169..
chr21	HAVANA	UTR	10863065	10863067	.	+	.	gene_id "ENSG00000169..

- Going to use a gene transfer format (GTF) file for annotations


Count Features

1. Make a folder inside the analysis folder to put the aligned reads in, `../03_count`
2. Change to the trimmed reads folder `../02_align/`



Count Features

```
for i in *.bam;
do featureCounts
    -a /mnt/data/gdata/human/hg38/chr21/ \
    homo_sapiens_hg38_chr21.gtf
    -o ../03_count/${i/ \
    Aligned.sortedByCoord.out.bam/ \
    counts.txt}
    -R BAM
    $i;
done
```

Count Features

```
for i in *.bam; 
do featureCounts
    -a /mnt/data/gdata/human/hg38/chr21/ \
        homo_sapiens_hg38_chr21.gtf
    -o ../03_count/${i/ \
        Aligned.sortedByCoord.out.bam/ \
        counts.txt}
    -R BAM
    $i;
done
```

Count Features

```
for i in *.bam;  loop condition  
do featureCounts  call program  
    -a /mnt/data/gdata/human/hg38/chr21/ \  
        homo_sapiens_hg38_chr21.gtf  
    -o ../03_count/${i/ \  
        Aligned.sortedByCoord.out.bam/ \  
        counts.txt}  
    -R BAM  
    $i;  
  
done
```

Count Features

```
for i in *.bam;
do featureCounts
-a /mnt/data/gdata/human/hg38/chr21/ \
homo_sapiens_hg38_chr21.gtf
-o ../03_count/${i/ \
Aligned.sortedByCoord.out.bam/ \
counts.txt}
-R BAM
$i;
done
```

Annotations in the diagram:

- loop condition**: Points to the `for i in *.bam;` line.
- call program**: Points to the `do featureCounts` line.
- path to genome annotation file**: Points to the `-a /mnt/data/gdata/human/hg38/chr21/ \` and `homo_sapiens_hg38_chr21.gtf` lines.

Count Features

```
for i in *.bam;
```

loop condition

```
do featureCounts
```

call program

path to genome
annotation file

```
-a /mnt/data/gdata/human/hg38/chr21/ \
homo_sapiens_hg38_chr21.gtf
```

where to write
the output

```
-o ../03_count/${i/ \
Aligned.sortedByCoord.out.bam/ \
counts.txt}
```

```
-R BAM
```

```
done
```

```
done
```

Count Features

```
for i in *.bam;
```

loop condition

```
do featureCounts
```

call program

path to genome
annotation file

```
-a /mnt/data/gdata/human/hg38/chr21/ \
homo_sapiens_hg38_chr21.gtf
```

where to write
the output

```
-o ../03_count/${i/ \
Aligned.sortedByCoord.out.bam/ \
counts.txt}
```

input files are BAM

```
-R BAM
$i;
```

```
done
```

Count Features

```
for i in *.bam;
```

loop condition

```
do featureCounts
```

call program

path to genome
annotation file

```
-a /mnt/data/gdata/human/hg38/chr21/ \
homo_sapiens_hg38_chr21.gtf
```

where to write
the output

```
-o ../03_count/${i/ \
Aligned.sortedByCoord.out.bam/ \
counts.txt}
```

input files are BAM

```
-R BAM
```

input file

```
 $i;
```

```
done
```

Count Features

```
for i in *.bam; do featureCounts -a  
/mnt/data/gdata/human/hg38/chr21/homo_sapiens_hg  
38_chr21.gtf -o  
../03_count/${i/Aligned.sortedByCoord.out.bam/co  
unts.txt} -R BAM $i; done
```

General Steps

1. Check quality
2. Trim
3. Align
4. Count features
5. Statistics