Introduction to Bioinformatics using Python

Assignment Brief

Dr. Alexey Larionov

01 November 2024

www.cranfield.ac.uk

FastQC Report Generator

Assignment_Instructions_and_Marking_Criteria_I-BIX-PYT-24.pdf

The aim of this assignment is to write a Python program to parse FastQC text files, and generate reports and plots as required by user

FastQ files: Raw unaligned reads

This information is given here for general background only

- De-facto standard for reporting results of raw short-reads sequencing data (Illumina)
- Text file describing the sequence of bases and the quality of each base
- The quality scores are presented by symbols (see next slide)

Read 1	Identifier	@WTCHG_20998_02:1:1108:4990:182444#CGATGT/1
	Sequence	AACCTGGAAACCCCTGCTTTGAGTGGTTCTGGCTTTCTGGACAAAACCAA
	Empty line	+
	Quality	>>==?>>??????>>@?>???>?>????@?@???????@??@?@@@???@@
Read 2		@WTCHG_22290_02:7:2201:11568:182063#CGATGT/1
		AGGGGCTGGGAGAGGCCCAGAAGGCTCTGAAGGAGTTTTGGTTTGGCTGG
		+
		>==>;>?>>>===>==>??>??>>???>?=??>?><?@??>?>>>;>>>>
::		

- FASTQ specification: http://maq.sourceforge.net/fastq.shtml

Quality scores in FASTQ files

This information is given here for general background only

Calculating

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

Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%
60	1 in 1,000,000	99.9999%

Encoding

Symbol	Phred Quality Score	Probability of Incorrect Base Call
!	0	1.000
"	1	0.794
#	2	0.631
$	3	0.501
%	4	0.398
&	5	0.316
		...
D	35	0.0003
E	36	0.0002
F	37	0.0002
G	38	0.0002
H	39	0.0001
I	40	0.0001

- Phred quality scores description: https://en.wikipedia.org/wiki/Phred_quality_score

FastQC

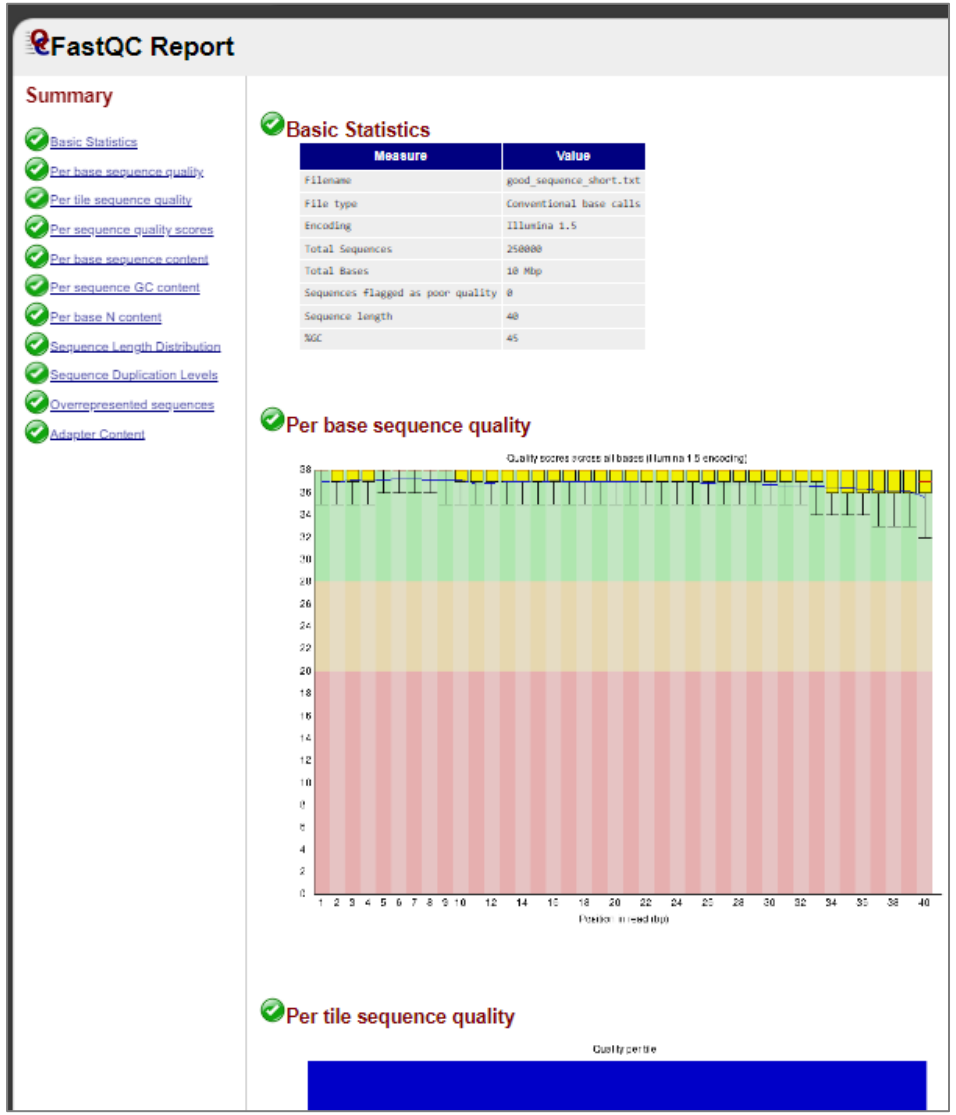
This information is given here for general background only

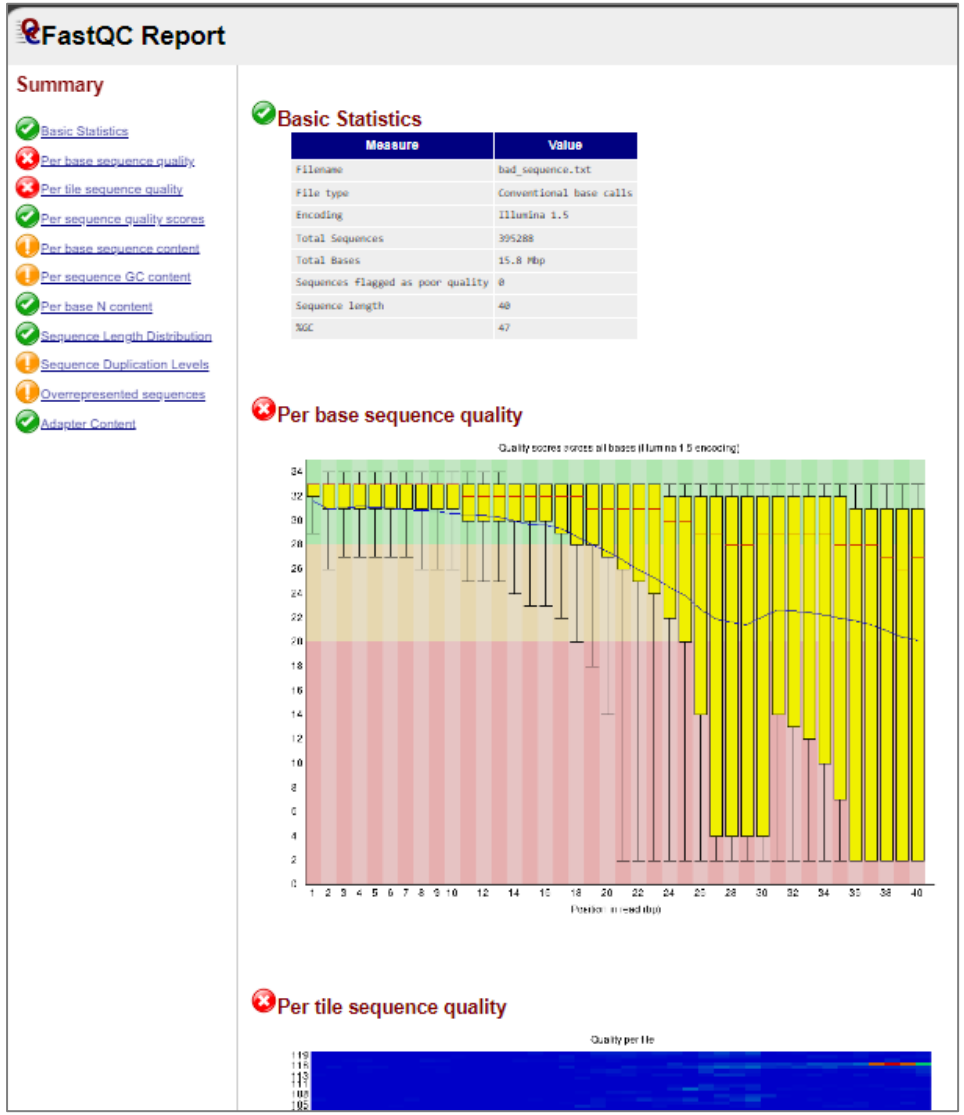
- FastQC is a bioinformatics tool developed in Babraham Bioinformatics Institute
- It generates QC report for raw sequencing data, usually reported in FASQ format
- In addition to FASTQ files FastQC may also accept data in some other formats, including:
  - GZip compressed FastQ
  - SAM
  - BAM
  - etc

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

FastQC HTML reports

This information is given here for general background only

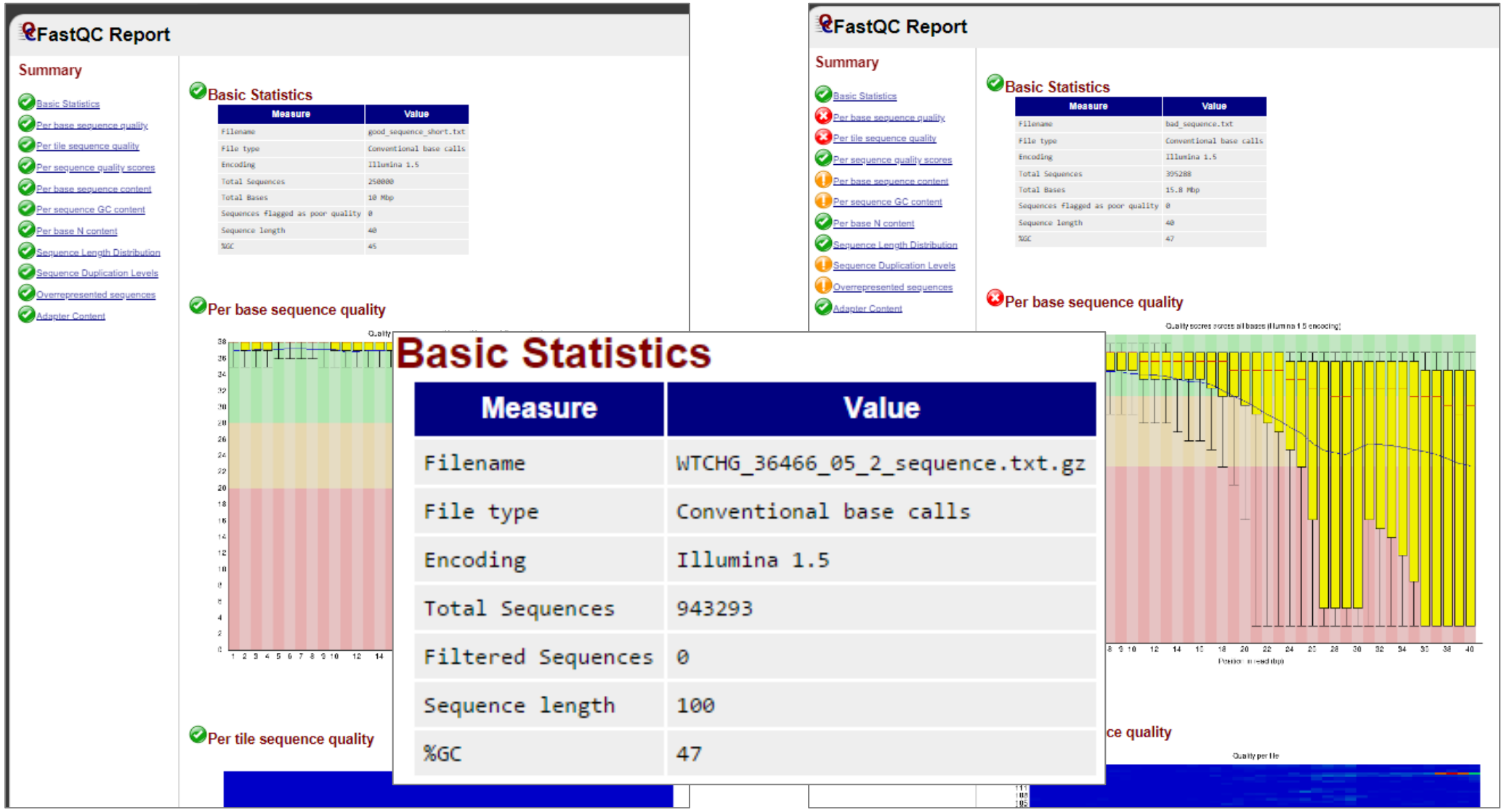




http://www.bioinformatics.babraham.ac.uk/projects/fastqc

FastQC HTML reports

This information is given here for general background only



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

FastQC text file

FastQC makes a text file with QC information

```
##FastQC    0.11.3
>>Basic Statistics  pass
#Measure    Value
Filename    4_age21_S12_L001_R1_001_concat.fastq.gz
File type   Conventional base calls
Encoding    Sanger / Illumina 1.9
Total Sequences 37287903
Sequences flagged as poor quality   0
Sequence length 75
%GC 55
>>END_MODULE
>>Per base sequence quality pass
#Base   Mean    Median  Lower Quartile  Upper Quartile  10th Percentile 90th Percentile
1   31.639491526246463  32.0    32.0    32.0    32.0    32.0
2   31.618002787660117  32.0    32.0    32.0    32.0    32.0
3   35.66454209023232   37.0    37.0    37.0    32.0    37.0
4   36.135634015138905  37.0    37.0    37.0    37.0    37.0
5   36.33817981129162   37.0    37.0    37.0    37.0    37.0
6   39.89045790534265   41.0    41.0    41.0    37.0    41.0
7   39.98228999362072   41.0    41.0    41.0    37.0    41.0
8   40.09800371450226   41.0    41.0    41.0    37.0    41.0
9   40.09410346835541   41.0    41.0    41.0    37.0    41.0
10  40.1102169515942    41.0    41.0    41.0    37.0    41.0
11  40.14826572575025   41.0    41.0    41.0    37.0    41.0
12  40.12512352866827   41.0    41.0    41.0    37.0    41.0
13  40.069397493337185  41.0    41.0    41.0    37.0    41.0
14  40.08390292154536   41.0    41.0    41.0    37.0    41.0
15  40.085222357502914  41.0    41.0    41.0    37.0    41.0
16  40.087172453757994  41.0    41.0    41.0    37.0    41.0
17  40.05952587357889   41.0    41.0    41.0    37.0    41.0
18  40.08766706457051   41.0    41.0    41.0    37.0    41.0
19  40.077069123463446  41.0    41.0    41.0    37.0    41.0
20  40.09115953771924   41.0    41.0    41.0    37.0    41.0
21  40.04739781156371   41.0    41.0    41.0    37.0    41.0
22  40.02398643871177   41.0    41.0    41.0    37.0    41.0
23  40.03484234551887   41.0    41.0    41.0    37.0    41.0
```

Basic statistics

The QC information is divided in following sections :

- Basic statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- K-mer Content

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

FastQC text file

FastQC makes a text file with QC information

The flag information could be **Pass**, **Warn** or **Fail**

```
##FastQC    0.11.3
>>Basic Statistics    pass
#Measure    Value
Filename    4_age21_S12_L001_R1_001_concat.fastq.gz
File type    Conventional base calls
Encoding    Sanger / Illumina 1.9
Total Sequences 37287903
Sequences flagged as poor quality    0
Sequence length 75
%GC 55
>>END_MODULE
>>Per base sequence quality pass
#Base    Mean    Median  Lower Quartile  Upper Quartile  10th Percentile 90th Percentile
1    31.639491526246463  32.0    32.0    32.0    32.0    32.0
2    31.618002787660117  32.0    32.0    32.0    32.0    32.0
3    35.66454209023232   37.0    37.0    37.0    32.0    37.0
4    36.135634015138905  37.0    37.0    37.0    37.0    37.0
5    36.33817981129162   37.0    37.0    37.0    37.0    37.0
6    39.89045790534265   41.0    41.0    41.0    37.0    41.0
7    39.98228999362072   41.0    41.0    41.0    37.0    41.0
8    40.09800371450226   41.0    41.0    41.0    37.0    41.0
9    40.09410346835541   41.0    41.0    41.0    37.0    41.0
10   40.1102169515942    41.0    41.0    41.0    37.0    41.0
11   40.14826572575025   41.0    41.0    41.0    37.0    41.0
12   40.12512352866827   41.0    41.0    41.0    37.0    41.0
13   40.069397493337185  41.0    41.0    41.0    37.0    41.0
14   40.08390292154536   41.0    41.0    41.0    37.0    41.0
15   40.085222357502914  41.0    41.0    41.0    37.0    41.0
16   40.087172453757994  41.0    41.0    41.0    37.0    41.0
17   40.05952587357889   41.0    41.0    41.0    37.0    41.0
18   40.08766706457051   41.0    41.0    41.0    37.0    41.0
19   40.077069123463446  41.0    41.0    41.0    37.0    41.0
20   40.09115953771924   41.0    41.0    41.0    37.0    41.0
21   40.04739781156371   41.0    41.0    41.0    37.0    41.0
22   40.02398643871177   41.0    41.0    41.0    37.0    41.0
23   40.03484234551887   41.0    41.0    41.0    37.0    41.0
```

Basic statistics

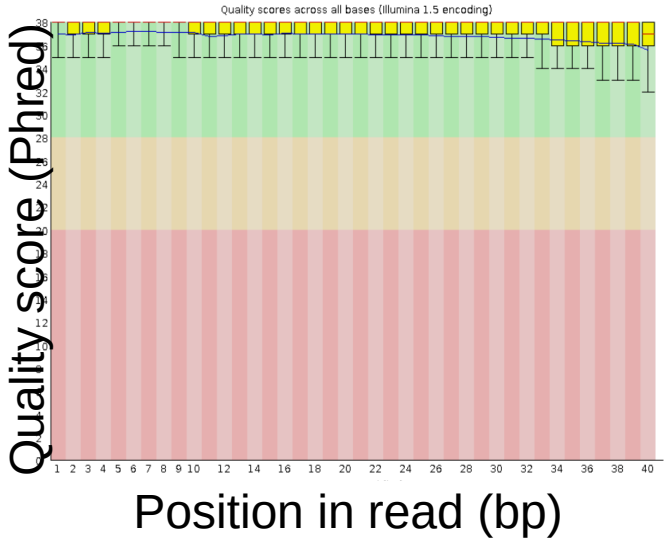
The QC information is divided in following sections :

- Basic statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- K-mer Content

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

Plots can be made for most of the FastQC file sections

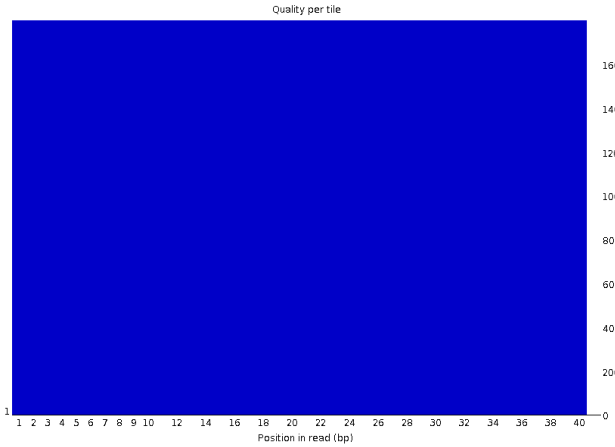
- The QC information is also presented using different graphs



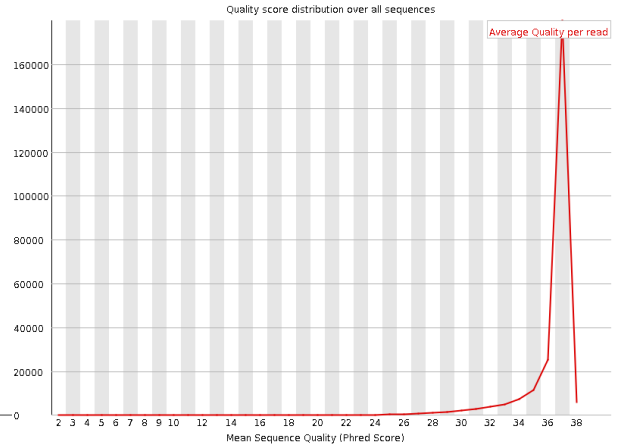
Per base sequence quality

```
##FastQC    0.11.3
>>Basic Statistics  pass
#Measure    Value
Filename    4_age21_S12_L001_R1_001_concat.fastq.gz
File type   Conventional base calls
Encoding    Sanger / Illumina 1.9
Total Sequences 37287903
Sequences flagged as poor quality   0
Sequence length 75
%GC 55
>>END_MODULE
>>Per base sequence quality pass
#Base   Mean    Median  Lower Quartile  Upper Quartile  10th Percentile 90th Percentile
1   31.639491526246463  32.0    32.0    32.0    32.0    32.0
2   31.618002787660117  32.0    32.0    32.0    32.0    32.0
3   35.66454209023232   37.0    37.0    37.0    32.0    37.0
4   36.135634015138905  37.0    37.0    37.0    37.0    37.0
5   36.33817981129162   37.0    37.0    37.0    37.0    37.0
6   39.89045790534265   41.0    41.0    41.0    37.0    41.0
7   39.98228999362072   41.0    41.0    41.0    37.0    41.0
8   40.09800371450226   41.0    41.0    41.0    37.0    41.0
9   40.09410346835541   41.0    41.0    41.0    37.0    41.0
10  40.1102169515942    41.0    41.0    41.0    37.0    41.0
11  40.14826572575025   41.0    41.0    41.0    37.0    41.0
12  40.12512352866827   41.0    41.0    41.0    37.0    41.0
13  40.069397493337185  41.0    41.0    41.0    37.0    41.0
14  40.08390292154536   41.0    41.0    41.0    37.0    41.0
15  40.085222357502914  41.0    41.0    41.0    37.0    41.0
16  40.087172453757994  41.0    41.0    41.0    37.0    41.0
17  40.05952587357889   41.0    41.0    41.0    37.0    41.0
18  40.08766706457051   41.0    41.0    41.0    37.0    41.0
19  40.077069123463446  41.0    41.0    41.0    37.0    41.0
20  40.09115953771924   41.0    41.0    41.0    37.0    41.0
21  40.04739781156371   41.0    41.0    41.0    37.0    41.0
22  40.02398643871177   41.0    41.0    41.0    37.0    41.0
23  40.03484234551887   41.0    41.0    41.0    37.0    41.0
```

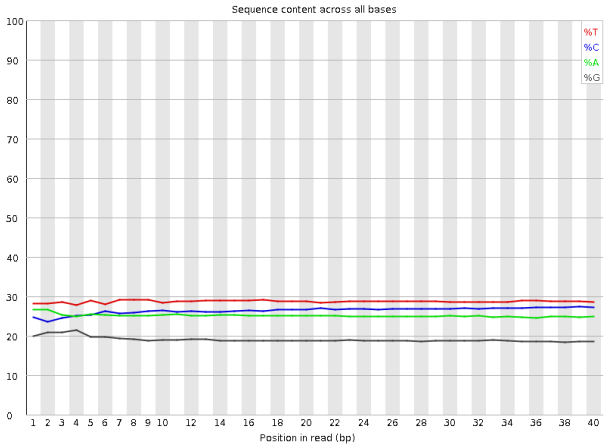
FastQC graphs



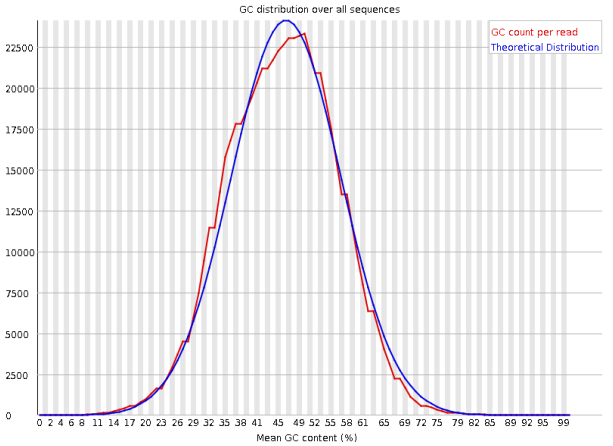
Per tile sequence quality



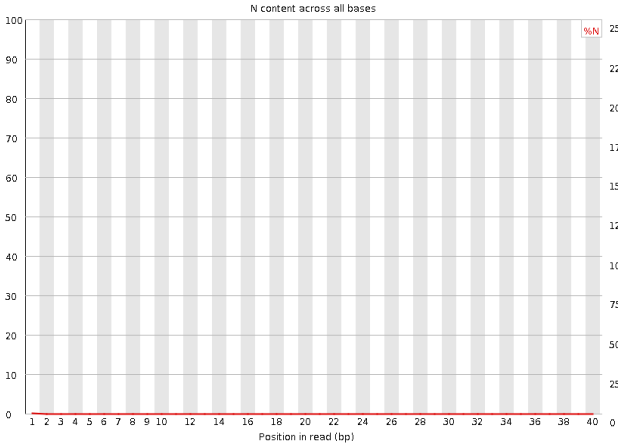
Per sequence quality score



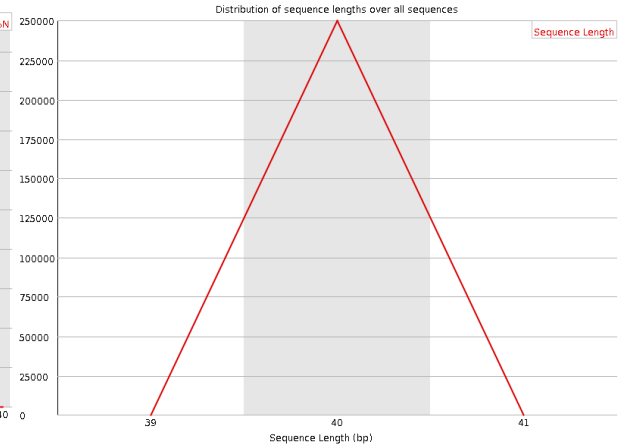
Per base sequence content



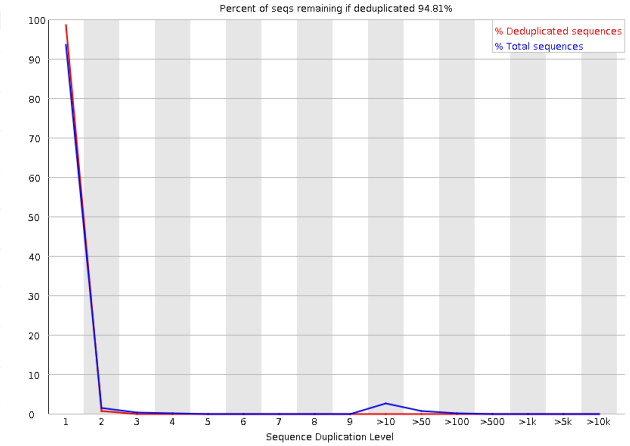
Per sequence GC content



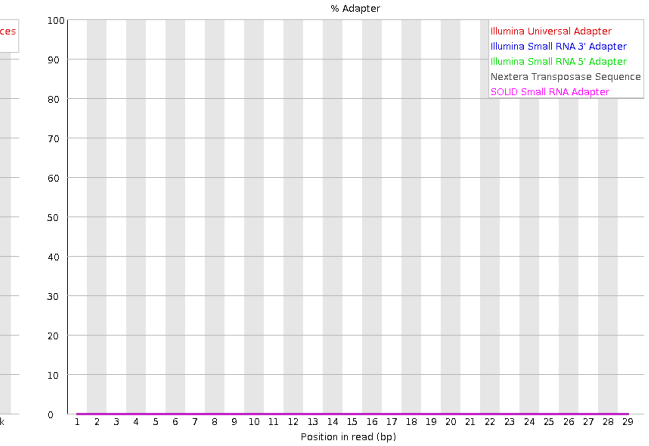
Per base N content



Sequence Length Distribution



Sequence Duplication Levels



Adapter Content

More information about FastQC report and graphs

More information about FastQC reports and graphs can be found in the:

Web:
http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
https://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules/

Provided file: FastQC_Notes.pdf

Your task

Write a Python program to parse FastQC text files,
and generate reports and plots as required by user

Your program should:

- ***Run in the terminal*** accepting ***command line arguments***
  (do NOT assume that users have any IDE like VS Code, Pycharm etc)

- Produce for each
  - ***Command-line*** output
  - ***Text files*** (summary reports and file with flags)
  - ***Plots*** (for most of the sections)

Command line arguments

Mandatory arguments:

1. The input file name
2. The output folder name

Optional arguments:

3. Any combination of optional arguments (or none) describing which sections of FastQC report should be processed

Table 1: Optional arguments

Short arg.	Long argument	File section	Required outputs
-b	--per_base_seq_qual	Per base sequence quality	R, P, F
-t	--per_tile_seq_qual	Per tile sequence quality	R, P, F
-s	--per_seq_qual_scores	Per sequence quality scores	R, P, F
-c	--per_base_seq_content	Per base sequence content	R, P, F
-g	--per_seq_GC_cont	Per sequence GC content	R, P, F
-n	--per_base_N_cont	Per base N content	R, P, F
-l	--seq_len_dist	Sequence Length Distribution	R, F
-d	--seq_dup	Sequence Duplication Levels	R, P, F
-o	--over_seq	Overrepresented sequences	R, F
-p	--adap_cont	Adapter Content	R, P, F
-k	--kmer_cont	K-mer Content	R, P, F
-a	--all	All the above	As above

R: Report, **P**: Plot, and **F**: File with Flag

Command line arguments examples

- User provides **-c** argument:

  ```
  python your_script.py input_file output_folder -c
  ```

  The content of the "***Basic Statistics***" section is printed to the terminal.

  The ***Report***, ***Plot*** and file with ***Flag*** are generated for the **"Per base sequence content"**.

  No other outputs are generated.

- User provides **-c** and **-o** arguments:

  ```
  python your_script.py input_file output_folder -c -o
  ```

  The content of the "Basic Statistics" section is printed to the terminal.

  The ***Report***, ***Plot*** and file with ***Flag*** are generated for the **"Per base sequence content"**.

  The ***Report*** and file with ***Flag*** are generated for the **"Overrepresented sequences"** section.

  No other outputs are generated.

Command line arguments examples

- User provides argument --**all**:

```
python your_script.py input_file output_folder -all
```

The content of the “Basic Statistics” section is printed to the terminal.
The required outputs are generated for all sections listed in Table 1.

- Use this option when generating the output, which you include into your submission

Outputs

Always print the content of the “Basic Statistics” section to the terminal

```
python your_script.py input_file output_folder -all
>>Basic Statistics  pass
#Measure      Value
Filename      4_age21_S12_L001_R1_001_concat.fastq.gz
File type     Conventional base calls
Encoding      Sanger / Illumina 1.9
Total Sequences     37287903
Sequences flagged as poor quality 0
Sequence length     75
%GC    55
>>END_MODULE
```

Outputs

Always print the content of the “Basic Statistics” section to the terminal

```
python your_script.py input_file output_folder -all
>>Basic Statistics  pass
#Measure      Value
Filename      4_age21_S12_L001_R1_001_concat.fastq.gz
File type     Conventional base calls
Encoding      Sanger / Illumina 1.9
Total Sequences     37287903
Sequences flagged as poor quality 0
Sequence length     75
%GC    55
>>END_MODULE
```

It doesn’t matter whether you include the opening and closing lines

Outputs

Report

The specified section should be extracted from the input file and saved into a separate text file. The report doesn't have to include ">>MODULE NAME" and ">>END_MODULE" lines.

File with Flag

A text file containing a single word: *pass*, *fail* or *warn*, extracted from the first line of the section.

Plot

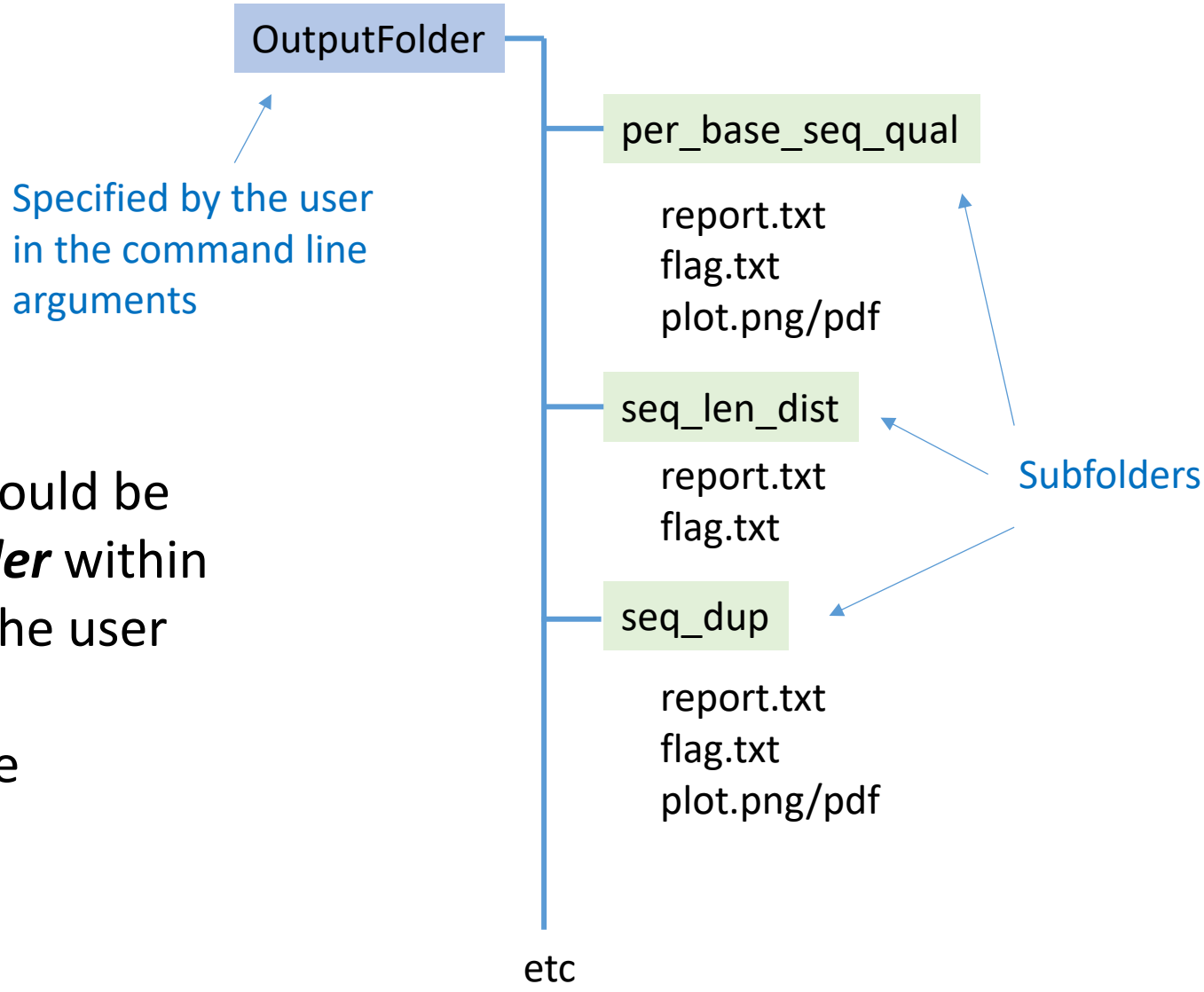
The plot to illustrate section's content (except for the "Sequence Length Distribution" and "Overrepresented sequences" sections).

The graphs do not need to be exactly like the ones generated by the FastQC tool.

Professional looking graphs will carry bonus marks.

The graphs may be generated in any image format (though PNG or PDF are preferred).

Output folder structure



- Output files for each section should be placed into ***a separate sub-folder*** within the output folder specified by the user
- The sub-folder names should be informative

Code features

Clear, structured, and understandable code

Modularising code, use of functions, meaningful variable names;
Classes are not necessary, but appropriate use of classes may bring additional marks, appropriate comments (in addition to the Docstrings!), etc

Error / exception handling

Using specialised Python methods for error handling.
Just use of *if-else* for error handling is discouraged.

Proper use of Docstring

It’s easy !

In addition to the Code

Technical documetation

The technical documentation should be written to explain how the program works to **a professional bioinformatician.** Should not exceed **2000 words**.

Concept diagram

The diagram to illustrate the general structure of your code, inputs/outputs, the used functions or classes etc. You may refer to the diagram in your technical documentation.

User manual

A **one-page** user manual for a user with biological background (**a non-bioinformatician**)

In addition to the Code and Documentation

Testing results

You should provide the results of your program testing generated with the optional flag **-all** (with the output files in sub-folders as required)

FASTQ files used for testing

You should include to the submission the FASTQ files that you used for testing (and any other materials that you might have used for testing)

Marking Scheme

Task	Maximal Mark
Code (70%)	
Passing command line arguments	5
Required output to the command line is generated	5
The required text files are generated in the required folders	15
The required plots are generated in the required folders	15
Clean readable/understandable code (with comments)	10
Error and exception handling	5
Proper use of Docstring	5
Professional looking graphs	5
Test results (including the input files that you used for testing)	5
Documentation (30%)	
Technical documentation	20
Concept diagram	5
User Manual	5
Total	**100**

Submission

- Assignment should be submitted via **CANVAS**
- You need to submit a **Zip archive** containing
  - The Python scripts
  - Documentation: Technical documentation + Concept diagram + User manual
  - The testing output: files in designated folders
  - The input files used for testing (FASTQ files provided to you with this assignment)
- The file name should include your student number and name like the following:
  - ***StudentName_StudentNumber_IBIX_PYT_24_ assignment.zip***
- **Submission deadline**: as shown in CANVAS
  - Different deadlines for Full- and Part- time (incl. apprenticeships) students

Python related questions: alexey.larionov@cranfield.ac.uk

Organisational questions: SAS & the Course Director