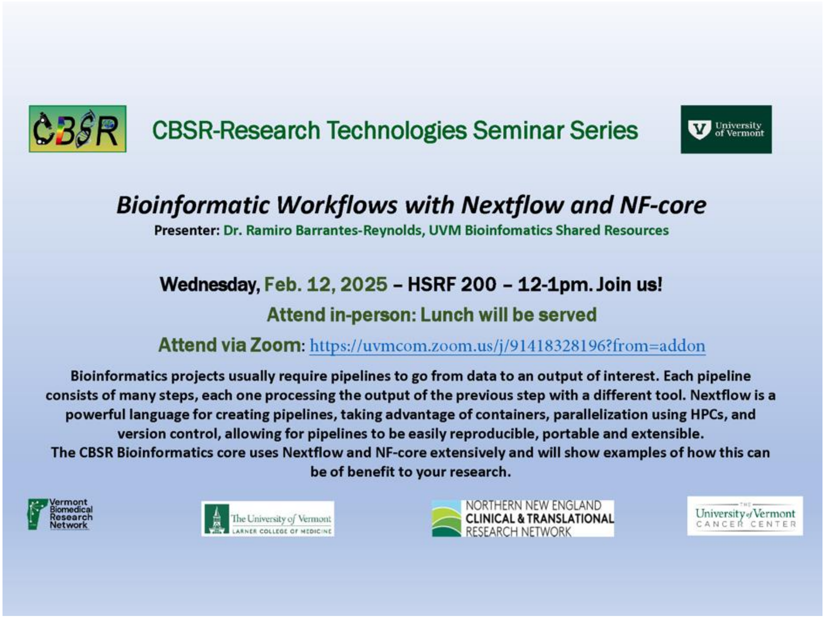
Assessing FASTQC Outputs

Dr. Princess Rodriguez

2025-01-29



CBSR-Research Technologies Seminar Series

Bioinformatic Workflows with Nextflow and NF-core

Presenter: **Dr. Ramiro Barrantes-Reynolds, UVM Bioinfomatics Shared Resources**

Wednesday, Feb. 12, 2025 – HSRF 200 – 12-1pm. Join us!

Attend in-person: Lunch will be served

Attend via Zoom: https://uvmcom.zoom.us/j/91418328196?from=addon

Bioinformatics projects usually require pipelines to go from data to an output of interest. Each pipeline consists of many steps, each one processing the output of the previous step with a different tool. Nextflow is a powerful language for creating pipelines, taking advantage of containers, parallelization using HPCs, and version control, allowing for pipelines to be easily reproducible, portable and extensible.
The CBSR Bioinformatics core uses Nextflow and NF-core extensively and will show examples of how this can be of benefit to your research.

Learning Objectives:

- Describe the contents and format of a FASTQ file
- Create a quality report using FASTQC
- Evaluate the quality of your NGS data using FastQC

Recap: Advantages of Batch Job Submissions

Batch job submission on an HPC (High-Performance Computing) system offers several advantages, particularly for computationally intensive tasks like bioinformatics, genomics, and large-scale data analysis.

1. **Efficient Resource Management:**
   - Jobs are queued and scheduled based on resource availability, ensuring optimal utilization of CPUs, memory, and GPUs.
   - Users can specify resource requirements (e.g., nodes, cores, memory) to avoid wasting computational power.

Recap: Advantages of Batch Job Submissions

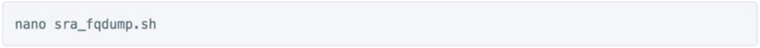
2. **Scalability:**
   - HPC clusters handle jobs of varying sizes, from single-threaded processes to massively parallel workloads.
   - Batch processing supports running multiple jobs concurrently, improving overall throughput.

Recap: Advantages of Batch Job Submissions

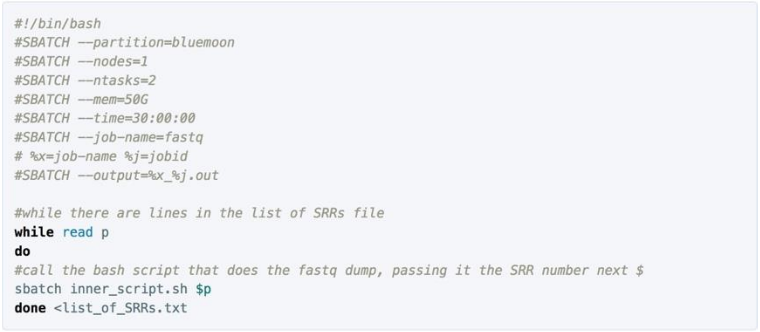
3. **Parallel Execution:**
    - Batch submission allows running thousands of jobs in parallel (e.g., processing multiple sequencing samples).
4. **Job Monitoring:**
    - Provides insights into job status, resource usage, and debugging.

Looking inside of `sra_fqdump.sh`

Purpose: Is to download FASTQ files from the SRA. FASTQ files to be downloaded are listed in a text file with accession numbers provided by you!



```
nano sra_fqdump.sh
```

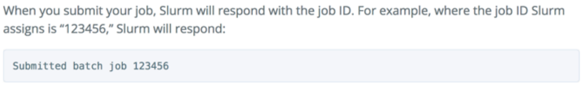


```
#!/bin/bash
#SBATCH --partition=bluemoon
#SBATCH --nodes=1
#SBATCH --ntasks=2
#SBATCH --mem=50G
#SBATCH --time=30:00:00
#SBATCH --job-name=fastq
# %x=job-name %j=jobid
#SBATCH --output=%x_%j.out

#while there are lines in the list of SRRs file
while read p
do
#call the bash script that does the fastq dump, passing it the SRR number next $
sbatch inner_script.sh $p
done <list_of_SRRs.txt
```

To submit a script use the command:

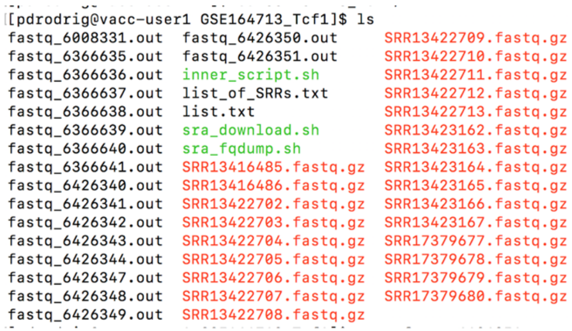
```
sbatch your-script.sh
```



When you submit your job, Slurm will respond with the job ID. For example, where the job ID Slurm assigns is “123456,” Slurm will respond:

```
Submitted batch job 123456
```

After submitting this script you will see `.out` files:



```
[pdrodrig@vacc-user1 GSE164713_Tcf1]$ ls
fastq_6008331.out  fastq_6426350.out     SRR13422709.fastq.gz
fastq_6366635.out  fastq_6426351.out     SRR13422710.fastq.gz
fastq_6366636.out  inner_script.sh       SRR13422711.fastq.gz
fastq_6366637.out  list_of_SRRs.txt      SRR13422712.fastq.gz
fastq_6366638.out  list.txt              SRR13422713.fastq.gz
fastq_6366639.out  sra_download.sh       SRR13423162.fastq.gz
fastq_6366640.out  sra_fqdump.sh         SRR13423163.fastq.gz
fastq_6366641.out  SRR13416485.fastq.gz  SRR13423164.fastq.gz
fastq_6426340.out  SRR13416486.fastq.gz  SRR13423165.fastq.gz
fastq_6426341.out  SRR13422702.fastq.gz  SRR13423166.fastq.gz
fastq_6426342.out  SRR13422703.fastq.gz  SRR13423167.fastq.gz
fastq_6426343.out  SRR13422704.fastq.gz  SRR17379677.fastq.gz
fastq_6426344.out  SRR13422705.fastq.gz  SRR17379678.fastq.gz
fastq_6426347.out  SRR13422706.fastq.gz  SRR17379679.fastq.gz
fastq_6426348.out  SRR13422707.fastq.gz  SRR17379680.fastq.gz
fastq_6426349.out  SRR13422708.fastq.gz
```

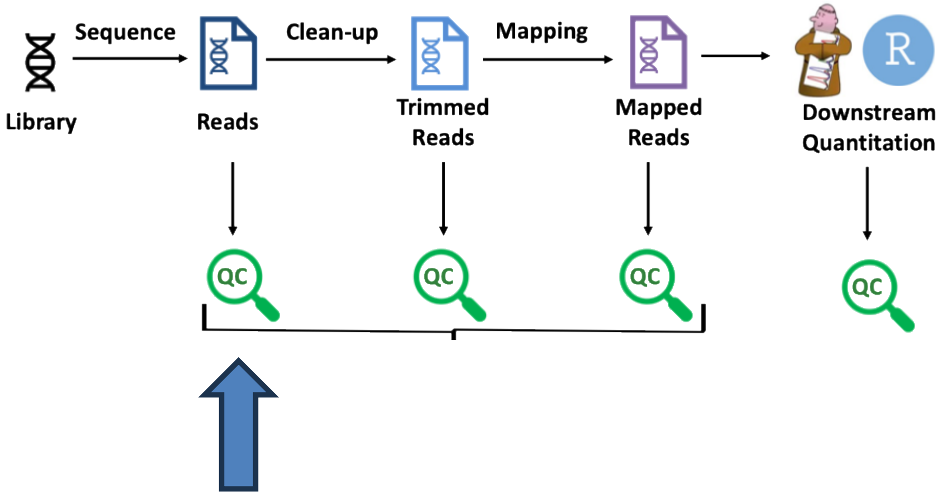
STDOUT records the output of programs

Three data streams exist for all Linux programs:

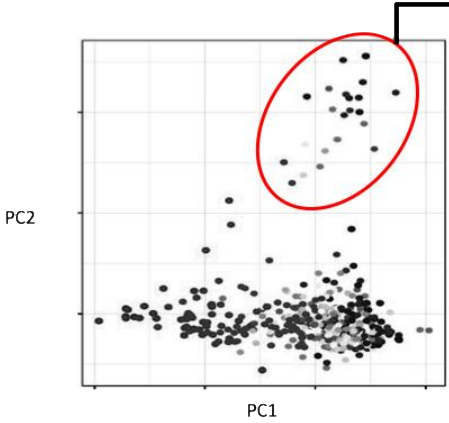
- STDIN (Standard Input - a way to send data into the program)
- STDOUT (Standard Output - a way to send expected data out of the program)
- STDERR (Standard Error - a way to send errors or warnings out of the program)

Next Step in Processing Sequencing Data

Processing Sequencing Data



What is the Point of QC? An Example…

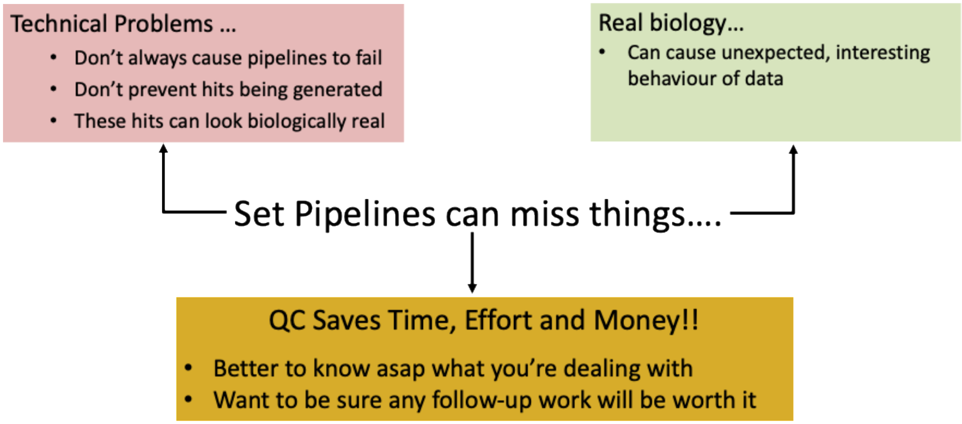


PC2 Genes (85 total)

- No clear biological theme
- No clear connection to system

What is going on?

What is the Point of QC?



Technical Problems …

- Don’t always cause pipelines to fail
- Don’t prevent hits being generated
- These hits can look biologically real

Real biology…

- Can cause unexpected, interesting behaviour of data

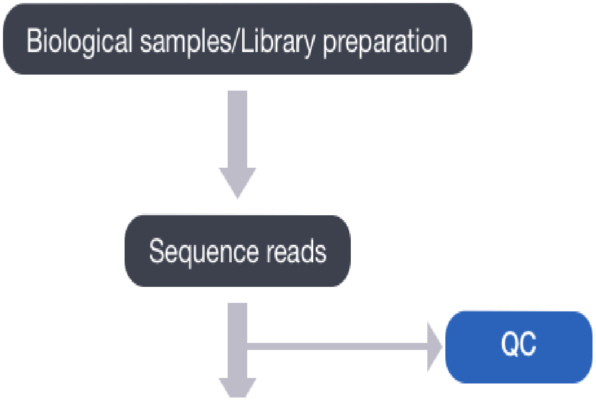
Set Pipelines can miss things….

QC Saves Time, Effort and Money!!

- Better to know asap what you’re dealing with
- Want to be sure any follow-up work will be worth it

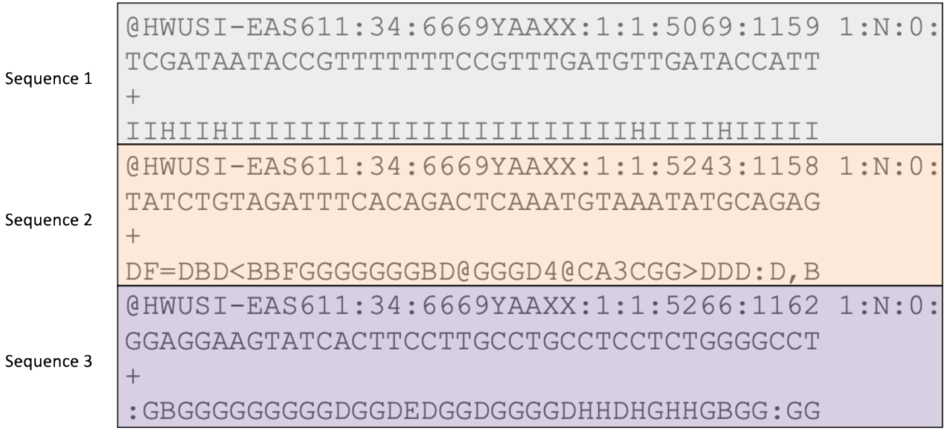
Quality Control of FASTQ files

The first step in the bioinformatic pipeline is to assess the quality of the sequence reads retrieved from the sequencing facility.



FASTQ files

Similar to FASTA, the FASTQ file begins with a header line. The difference is that the FASTQ header is denoted by a `@` character.

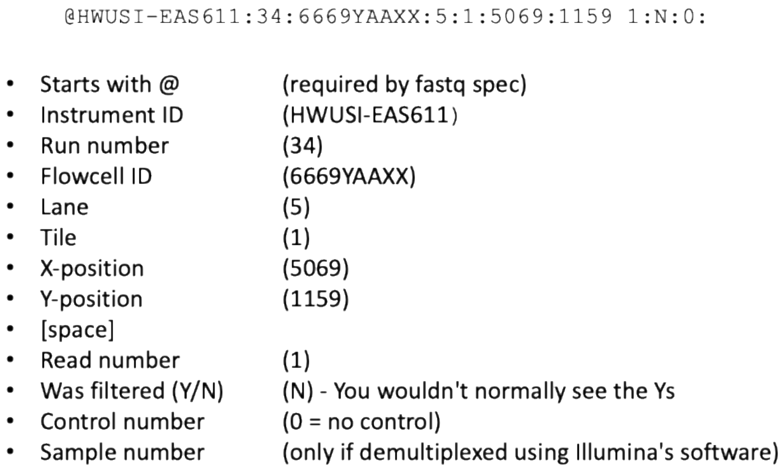


Sequence 1
```
@HWUSI-EAS611:34:6669YAAXX:1:1:5069:1159 1:N:0:
TCGATAATACCGTTTTTTTCCGTTTGATGTTGATACCATT
+
IIHIIHIIIIIIIIIIIIIIIIIIIIIIIHIIIIHIIIII
```

Sequence 2
```
@HWUSI-EAS611:34:6669YAAXX:1:1:5243:1158 1:N:0:
TATCTGTAGATTTCACAGACTCAAATGTAAATATGCAGAG
+
DF=DBD<BBFGGGGGGGBD@GGGD4@CA3CGG>DDD:D,B
```

Sequence 3
```
@HWUSI-EAS611:34:6669YAAXX:1:1:5266:1162 1:N:0:
GGAGGAAGTATCACTTCCTTGCCTGCCTCCTCTGGGGCCT
+
:GBGGGGGGGGDGGDEDGGDGGGGDHHDHGHHGBGG:GG
```

Illumina Header Sections (Line 1)



```
@HWUSI-EAS611:34:6669YAAXX:5:1:5069:1159 1:N:0:
```

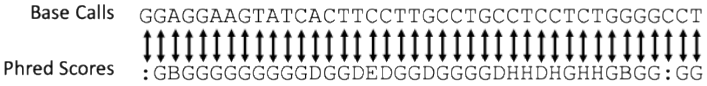
• Starts with @	(required by fastq spec)
• Instrument ID	(HWUSI-EAS611)
• Run number	(34)
• Flowcell ID	(6669YAAXX)
• Lane	(5)
• Tile	(1)
• X-position	(5069)
• Y-position	(1159)
• [space]	
• Read number	(1)
• Was filtered (Y/N)	(N) - You wouldn't normally see the Ys
• Control number	(0 = no control)
• Sample number	(only if demultiplexed using Illumina's software)

A Single FASTQ Entry

```
1. @HWUSI-EAS611:34:6669YAAXX:1:1:5266:1162 1:N:0:
2. GGAGGAAGTATCACTTCCTTGCCTGCCTCCTCTGGGGCCT
3. +
4. :GBGGGGGGGGDGGDEDGGDGGGGDHHDHGHHGBGG:GG
```

1. Header - starts with @
2. Base calls (can include N or IUPAC codes)
3. Mid-line - starts with + usually empty
4. Quality scores (= Phred Scores)

Phred Scores (Line 4)



```
Base Calls     GGAGGAAGTATCACTTCCTTGCCTGCCTCCTCTGGGGCCT
Phred Scores   :GBGGGGGGGGGDGGDEDGGDGGGGDHHDHGHHGBGG:GG
```

- Each quality score represents the probability that the corresponding nucleotide call is incorrect.

_ This quality score is logarithmically based and is calculated as:

```
Q = -10 x log10(P)
```

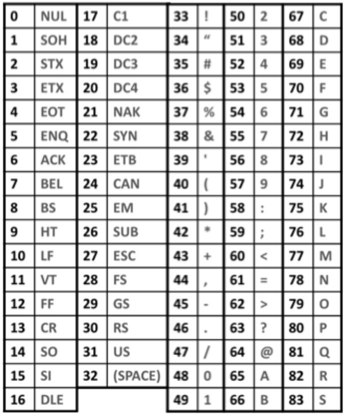
- P is the probability that a base call is erroneous
  - Phred = -10 * (int)$\log_{10}$(p)
    - p=0.1 Phred = 10
    - p=0.01 Phred = 20
    - p=0.001 Phred = 30

Higher Phred Score
Higher Confidence

Phred Score Encoding

- Translation of Phred score to single ASCII letter
- Based on standard ASCII table

Different quality encoding scales exist, but the most commonly one used is fastqsanger, which is the scale output by Illumina since mid-2011.



0	NUL	17	C1	33	!	50	2	67	C
1	SOH	18	DC2	34	"	51	3	68	D
2	STX	19	DC3	35	#	52	4	69	E
3	ETX	20	DC4	36	$	53	5	70	F
4	EOT	21	NAK	37	%	54	6	71	G
5	ENQ	22	SYN	38	&	55	7	72	H
6	ACK	23	ETB	39	'	56	8	73	I
7	BEL	24	CAN	40	(	57	9	74	J
8	BS	25	EM	41	)	58	:	75	K
9	HT	26	SUB	42	*	59	;	76	L
10	LF	27	ESC	43	+	60	<	77	M
11	VT	28	FS	44	,	61	=	78	N
12	FF	29	GS	45	-	62	>	79	O
13	CR	30	RS	46	.	63	?	80	P
14	SO	31	US	47	/	64	@	81	Q
15	SI	32	(SPACE)	48	0	65	A	82	R
16	DLE			49	1	66	B	83	S

Babraham Bioinformatics

How QC Programmes Fits Into Processing Pipelines

QC metrics we can work with:

Phred Scores

How the Sequencer Performed

- At different cycles
- Across different locations
- For different reads

Library Composition

Adapter
Insert
Adapter

The Nature of our Sequenced Reads

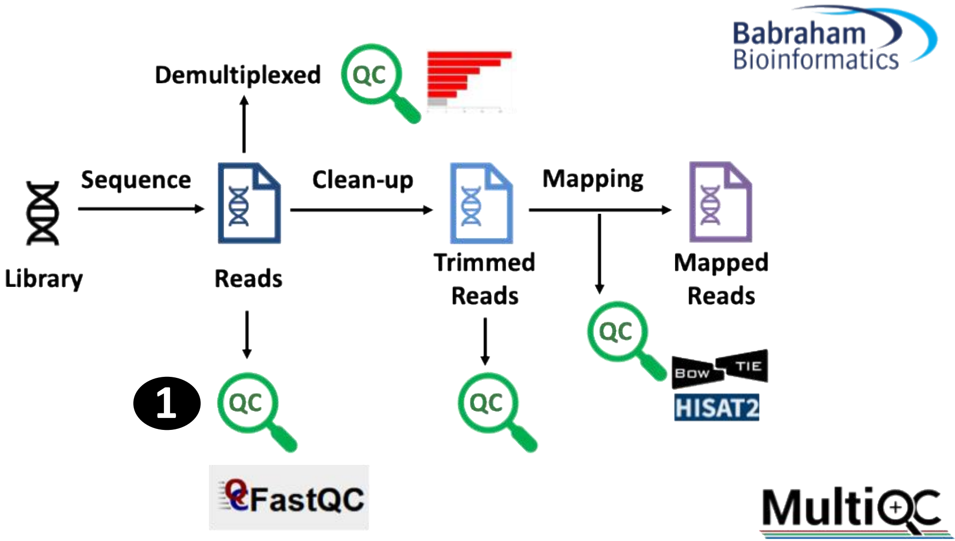
- Biases
- Contaminants
- Duplication

Mapping

Where our Sequencing Reads Come From

- Species (plural or singular!)
- Region (repetitive or unique)

QC Programs in Data Processing

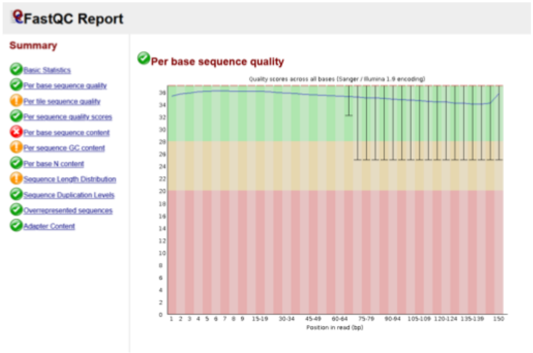


FastQC

- FastQC provides a simple way to do some quality checks on raw sequence data coming from high throughput sequencing pipelines.
- It provides a modular set of analyses, which you can use to obtain an impression of whether your data has any problems that you should be aware of before moving on to the next analysis.

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

FastQC



FastQC Report

Summary

- ✓ 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

✓ Per base sequence quality

```
fastqc seqfile1 seqfileN
fastqc *.fastq.gz
```

- Reads raw fastq file(s)
- Performs multiple checks
  - Pass/warn/fail
  - Compares to genomic library
- Generates a HTML Report

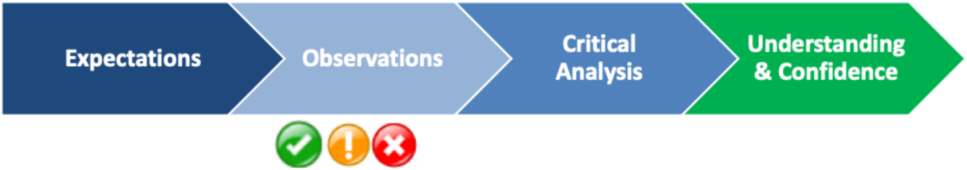
https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Babraham Bioinformatics

Interpreting the HTML report

- Within the report, a summary of all of the modules is given on the left-hand side.
- Do not take the **yellow “WARNING”s** and **red “FAIL”s as “*this sample is not usable*”**; they should be interpreted as flags!

Context is Key for QC

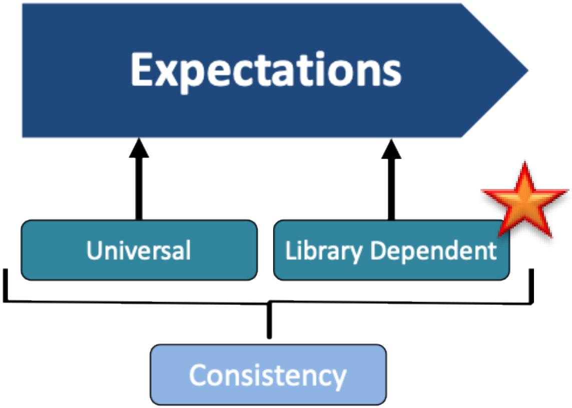


QC should be about what you expect and what you see

Context is Key for QC

Individual Library:

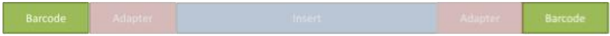
Replicate Libraries:



Babraham Bioinformatics

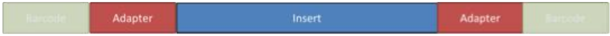
Universal QC Metrics

- Demultiplexing

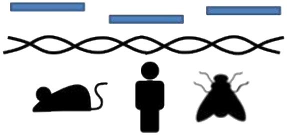


- Base Call Quality

- Adapter Content



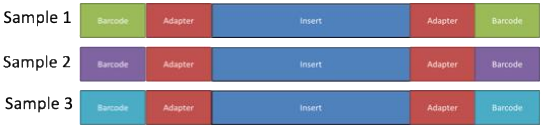
- Mapping Quality



Babraham Bioinformatics

Demultiplexing: Expectation

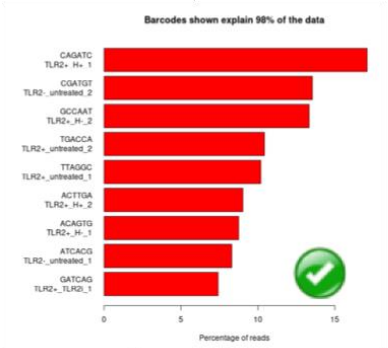
Only the barcodes we assigned to samples should be present—**no others**



ect...

What barcode sequences have we found?

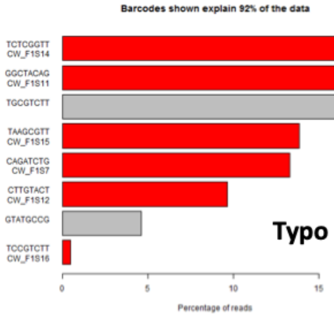
🟥 Expected Barcode ⬜ Unknown Sequence

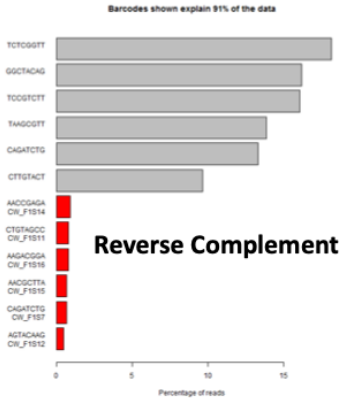


What could unknown barcode sequences mean?

Demultiplexing: Unknown

🟥 Expected Barcode ⬜ Unknown Sequence

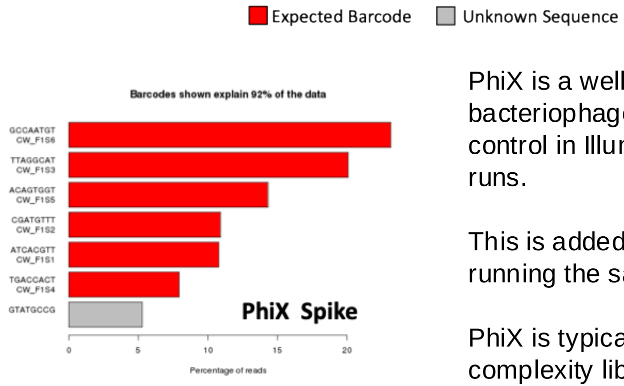




Human Error is a really common source of barcode issues

Babraham Bioinformatics

Demultiplexing: Unknown



Sometimes more technical in nature…

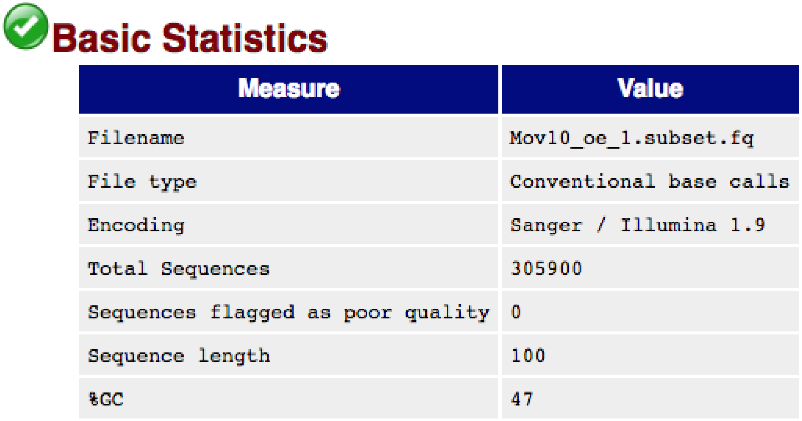
PhiX is a well characterized bacteriophage that is used as a control in Illumina sequencing runs.

This is added by the individual running the sample at the core.

PhiX is typically added for low-complexity libraries. To add sequence diversity.

Improves flow cell clustering and reduces sequencing errors.

Basic Statistics



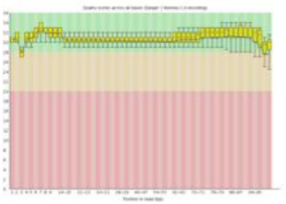
Basic Statistics

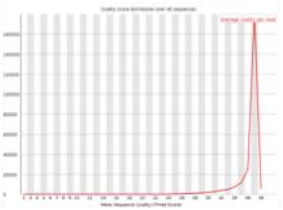
Measure	Value
Filename	Mov10_oe_1.subset.fq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	305900
Sequences flagged as poor quality	0
Sequence length	100
%GC	47

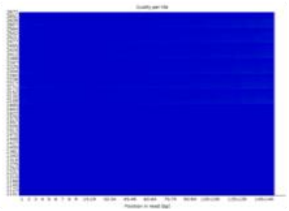
Base Call Quality: Expectations

Illumina Sequencers are technically reliable, so **we expect confident calls**

$$\begin{tikzcd}
 & \text{Phred scores} \arrow[dl] \arrow[d] \arrow[dr] & \\
\text{Per base/cycle} & \text{Per Read} & \text{Location}
\end{tikzcd}$$







FastQC Report

✓ Per base sequence quality

FastQC Report

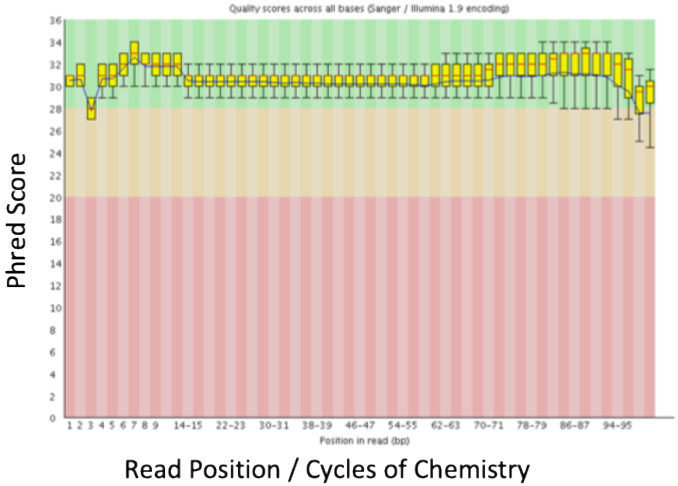
✓ Per sequence quality scores

FastQC Report

✓ Per tile sequence quality

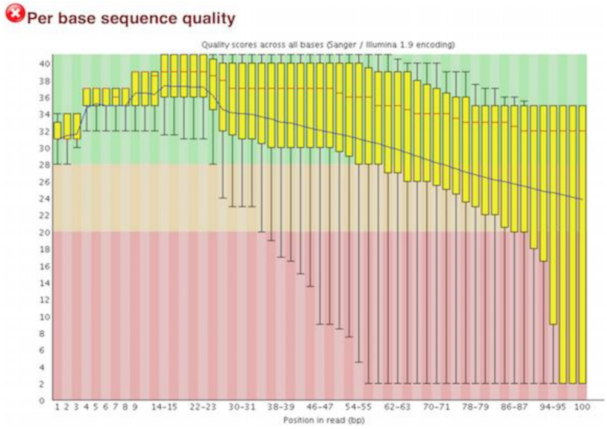
Babraham Bioinformatics

Per base sequence quality



The yellow box represents the 25th and 75th percentiles, with the red line as the median. The whiskers are the 10th and 90th percentiles. The blue line represents the average quality score for the nucleotide. Based on these metrics, the quality scores for nearly all reads have scores above 28.

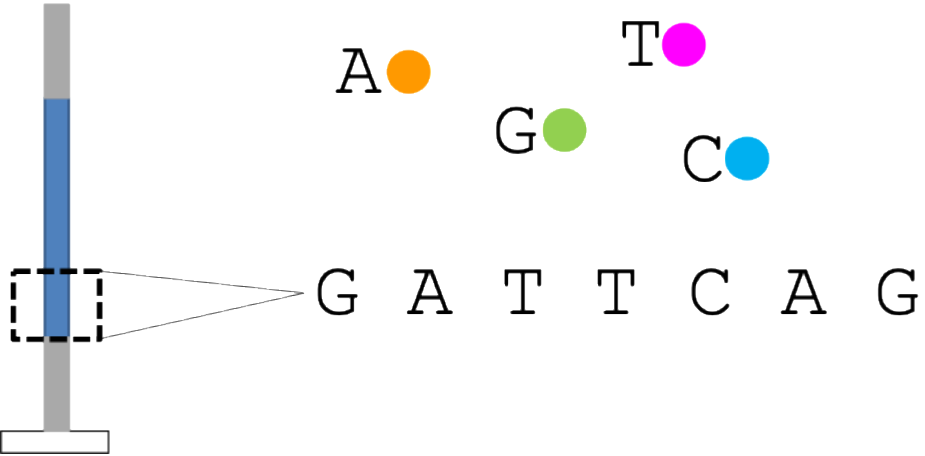
Per base sequence quality



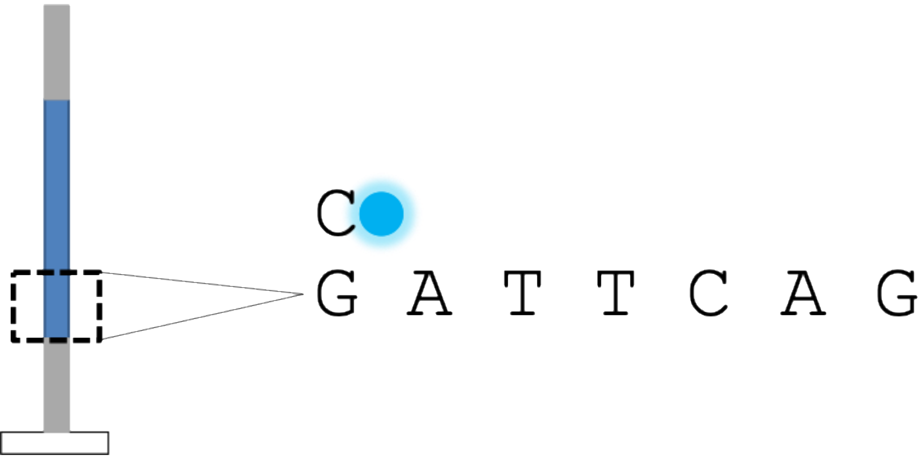
For reads generated by Illumina sequencing, this is not alarming and there are known causes for this drop in quality.

Clusters get out of sync over length of the read

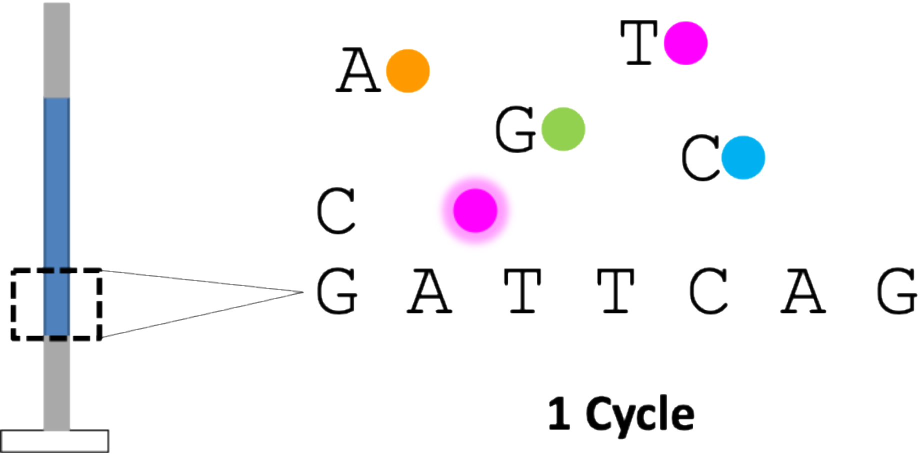
Sequence by Synthesis



Sequence by Synthesis

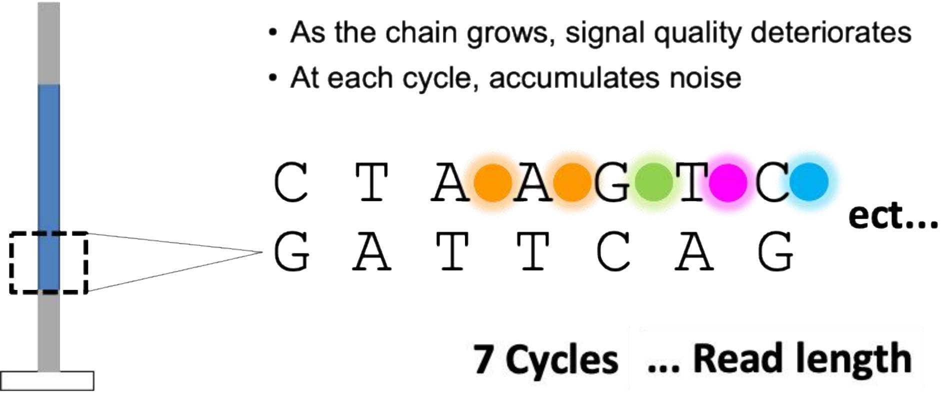


Sequence by Synthesis



A photo is taken….

Sequence by Synthesis

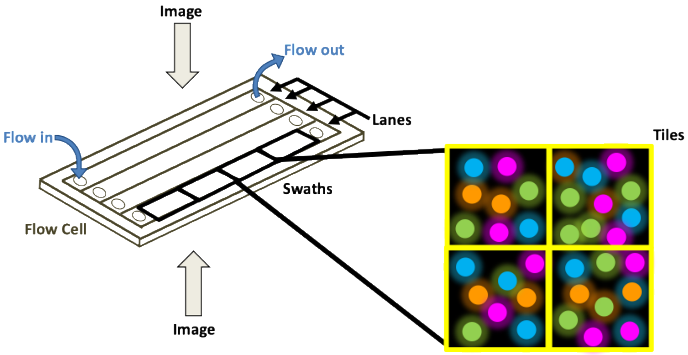


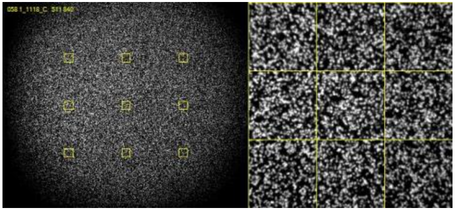
Problem

- As the chain grows, signal quality deteriorates
- At each cycle, accumulates noise

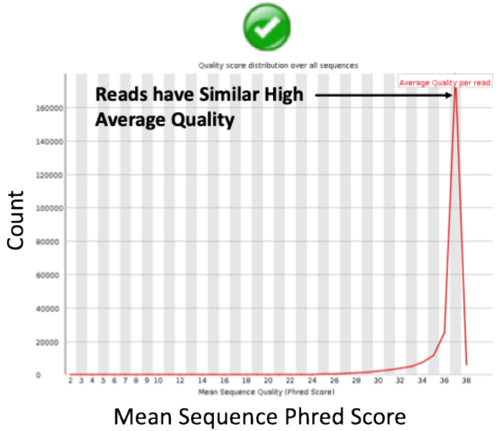
Babraham Bioinformatics

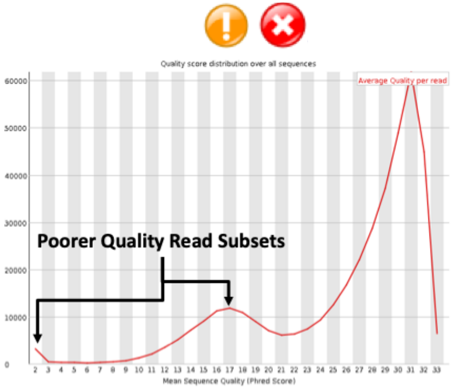
Flow Cell Imaging



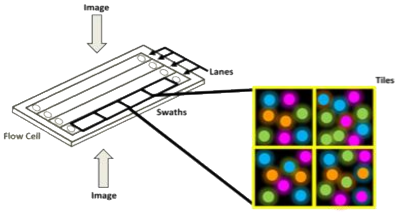


Per sequence quality scores

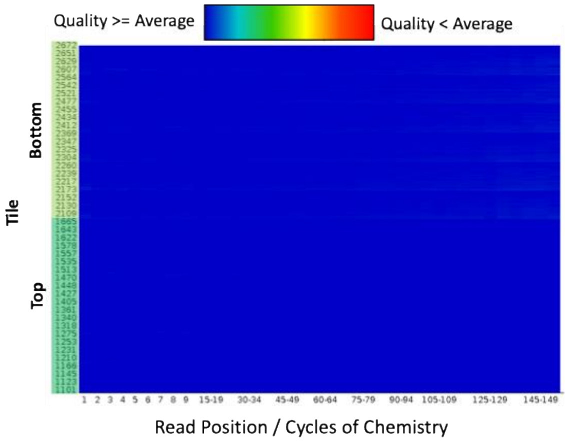




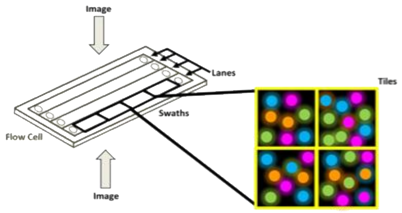
Positional Quality



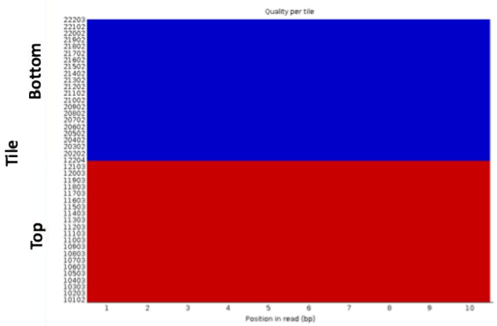
A good tile plot is a blue square



Positional Quality



A good tile plot is a blue square

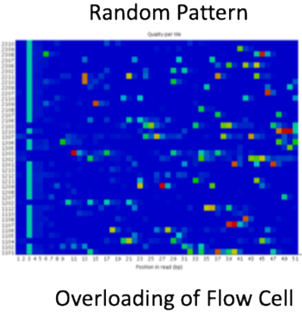


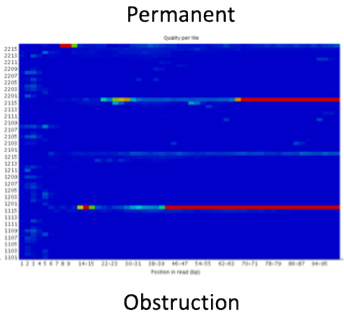
Focusing Fail!

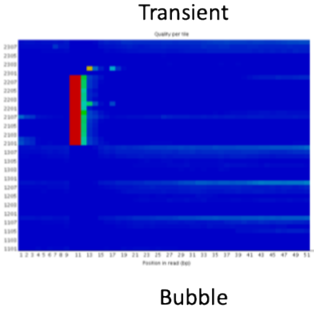
Positional Quality

More Examples of Positional Fails

Position Specific Patterning

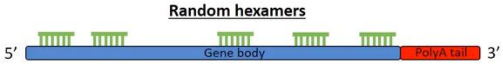


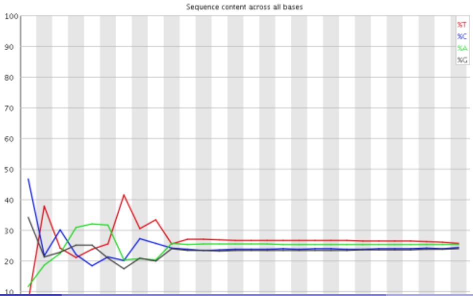




Per base sequence content

Always gives a FAIL for RNA-seq data. This is because the first 10-12 bases result from the ‘random’ hexamer priming that occurs during RNA-seq library preparation.
This priming is not as random as we might hope giving an enrichment in particular bases for these intial nucleotides.





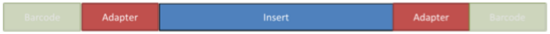
Universal QC Metrics

- Demultiplexing

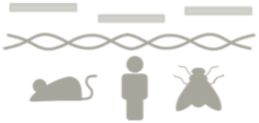
Barcode
Adapter
Insert
Adapter
Barcode

- Base Call Quality

- Adapter Content

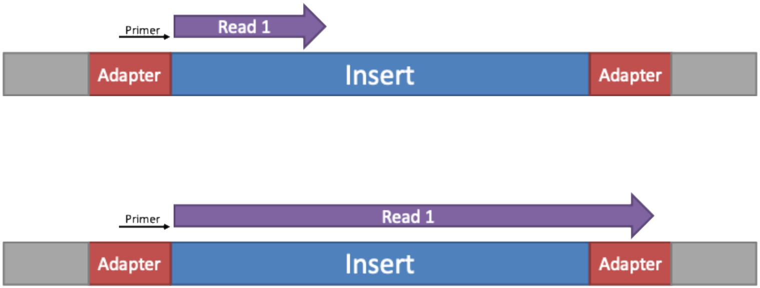


- Mapping Quality



Babraham Bioinformatics

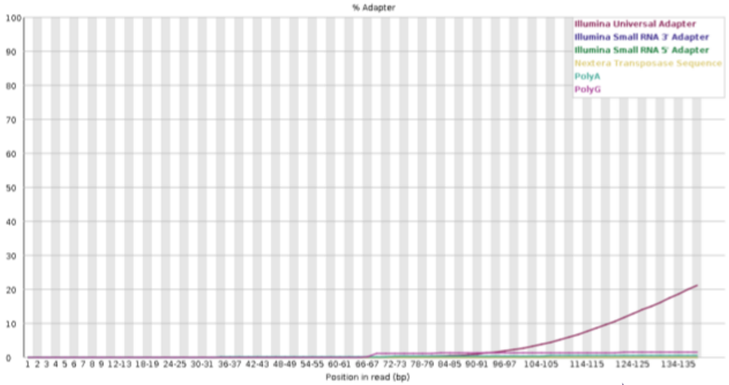
Adapter Content

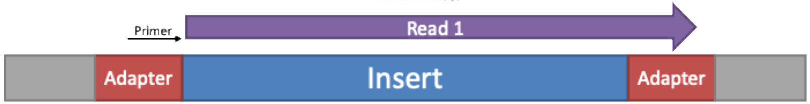


Due to variable insert and read length, **we may sequence adapter at the end of our reads**

Babraham Bioinformatics

Measuring Read-through Adapters





Babraham Bioinformatics

Library Dependent QC Metrics

From the Base Sequence:

- GC Content

- Base Composition

GATC

- Duplication

GATCTACGAGTTACGATCAGT
GATCTACGAGTTACGATCAGT
GATCTACGAGTTACGATCAGT
GATCTACGAGTTACGATCAGT

Library Dependent QC Metrics

From the Base Sequence:

- GC Content

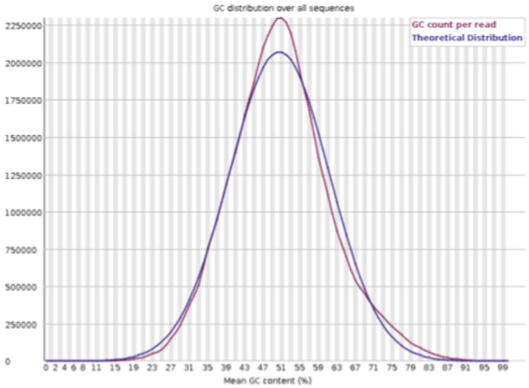
- Base Composition

GATC

- Duplication

GATCTACGAGTTACGATCAGT
GATCTACGAGTTACGATCAGT
GATCTACGAGTTACGATCAGT
GATCTACGAGTTACGATCAGT

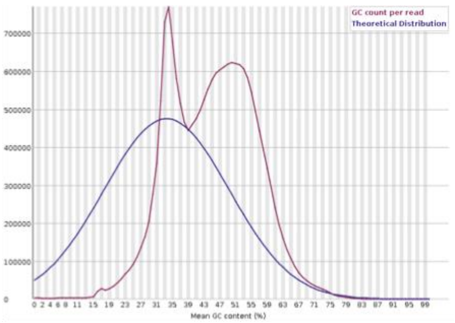
Library GC Content



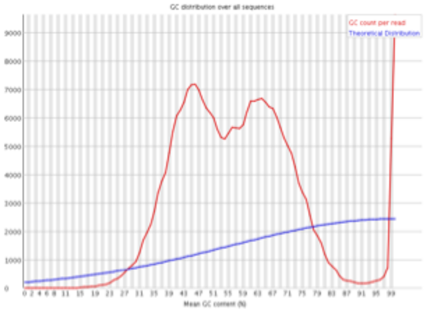
- Generic summary of library composition at a read level
- Expect a normally distributed set of values centred on the overall GC content

Per sequence GC content

This plot would indicate some type of over-represented sequence with the sharp peaks, indicating either contamination or a highly over-expressed gene.

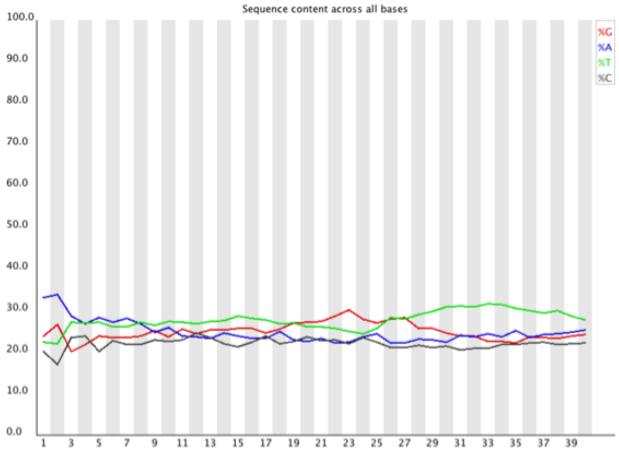


Single contamination



Broad contamination

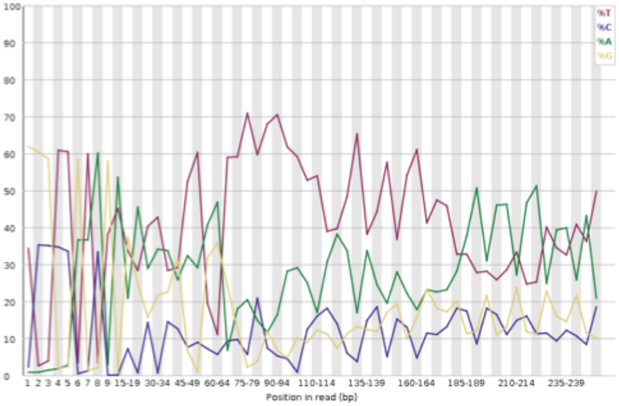
Library Base Composition



- For every chemistry cycle we can look at the number of ATGC
- The composition should be the same for all cycles

Library Base Composition: Bias

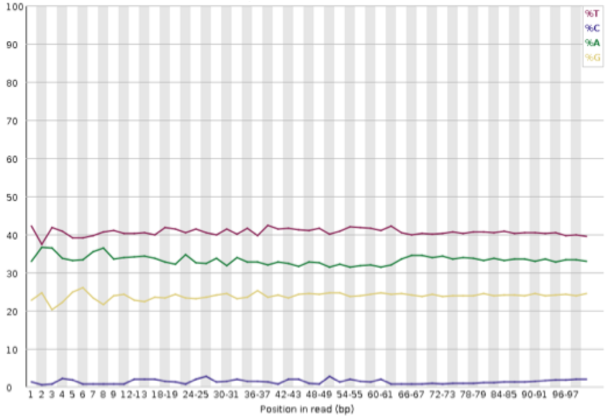
Amplicon



Very low diversity

Library Base Composition: Bias

WGBS



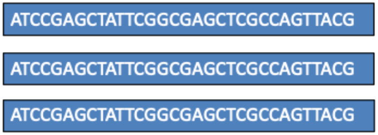
Consistent disproportional expression of bases

Bisulphite treated – C is converted to T

Duplication

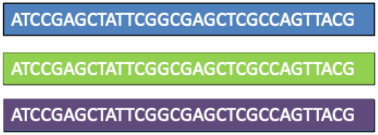
If the exact same sequence appears more than once it could be…

Technical:


ATCCGAGCTATTCGGCGAGCTCGCCAGTTACG
ATCCGAGCTATTCGGCGAGCTCGCCAGTTACG
ATCCGAGCTATTCGGCGAGCTCGCCAGTTACG

- PCR duplicates

Coincidental:


ATCCGAGCTATTCGGCGAGCTCGCCAGTTACG
ATCCGAGCTATTCGGCGAGCTCGCCAGTTACG
ATCCGAGCTATTCGGCGAGCTCGCCAGTTACG

- Deep sequencing
- Highly present sequences
- Restricted diversity libraries

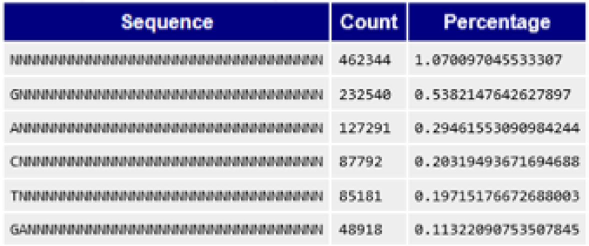
Babraham Bioinformatics

Overrepresented Sequences

- Extreme duplication
- Displays the sequences (at least 20 bp) that occur in more than 0.1% of the total number of sequences.
- The exact same sequence is a significant proportion of the whole library (which might not be duplicated overall)

Overrepresented Sequences: Poly N

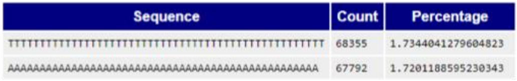
PolyN – Quality too poor to make any calls



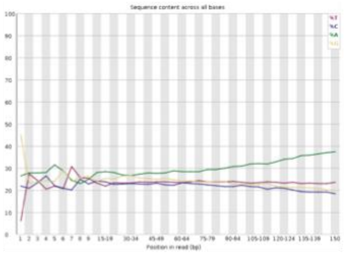
Sequence	Count	Percentage
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	462344	1.070097045533307
GNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	232540	0.5382147642627897
ANNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	127291	0.29461553090984244
CNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	87792	0.2031949367169468
TNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	85181	0.19715176672688003
GANNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	48918	0.11322090753507845

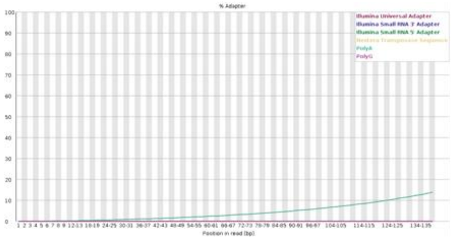
Overrepresented Sequences: Poly A

PolyA (or PolyT) – Common in RNA-Seq

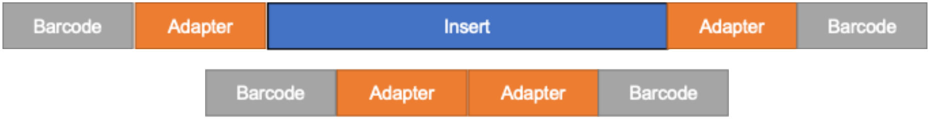


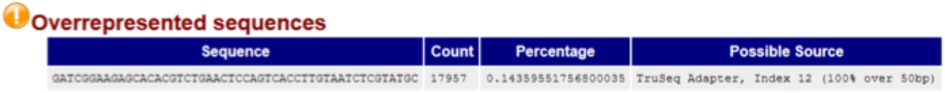
Sequence	Count	Percentage
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT	68355	1.7344041279604823
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	67792	1.7201188595230343





Overrepresented Sequences: Adapter Dimers




Overrepresented sequences

Sequence	Count	Percentage	Possible Source
GATCGGAAGAGCACACGTCTGAACTCCAGTCACCTTGTAATCTCGTATGC	17957	0.14359551756800035	TruSeq Adapter, Index 12 (100% over 50bp)

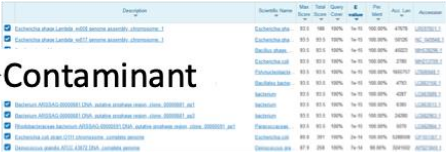
Overrepresented Sequences: Specific Sequences

- Other potential sources...

Sequence	Count	Percentage	Possible Source
GGCTTCCTCGGCCCCGGGATTCGGCGAAAGCTGCGGCCGGAGGGCTGTAA	746766	1.360148419566899	No Hit

Basic Local Alignment Search Tool

Sequence	Count	Percentage	Possible Source
CTTATACACATCTCCGAGCCCACGAGACTAAGGCGAATCTCGTATGCCGT	2767629	5.149013792611521	No Hit



Let’s Practice Running FastQC

We would like to run the FastQC tool on fastq files in the `raw_fastq` directory.

```
fastqc --help
```

Try running it now.

Run `fastqc` with the Environment Module System:

```
module load gcc/13.3.0-xp3epyt
module load fastqc/0.12.1-qxseug5
```

How to use `fastqc`?

```
fastqc --help

SYNOPSIS

    fastqc seqfile1 seqfile2 .. seqfileN

    fastqc [-o output dir]
    [--(no)extract]
    [-f fastq|bam|sam]
    [-c contaminant file]
    seqfile1 .. seqfileN
```

Quick Exercise

Run fastqc on `Mov10_oe_1.subset.fq`

```
fastqc Mov10_oe_1.subset.fq
```

FASTQC Outputs

For each individual FASTQ file that is input to FastQC, there are **two output files that are generated**.

1. The first is **an HTML file** which is a self-contained document with various graphs embedded into it. Each of the graphs evaluate different quality aspects of our data, we will discuss in more detail in this lesson.
2. Alongside the HTML file is **a zip file**. This file contains the different plots from the report as separate image files but also contains data files which are designed to be easily parsed to allow for a more detailed and automated evaluation of the raw data on which the QC report is built.

Class Exercise #1

- Run FASTQC on all FASTQ files in `raw_fastq`. FASTQC allows you to redirect your output into a specified location with the `-o` parameter. Be sure to use this parameter in your final code.

If successful, you will see the following outputs inside of the `fastqc` folder:

```
Irrel_kd_1.subset_fastqc.html  Irrel_kd_3.subset_fastqc.h
Irrel_kd_1.subset_fastqc.zip   Irrel_kd_3.subset_fastqc.z
Irrel_kd_2.subset_fastqc.html  Mov10_oe_1.subset_fastqc.h
Irrel_kd_2.subset_fastqc.zip   Mov10_oe_1.subset_fastqc.z
```

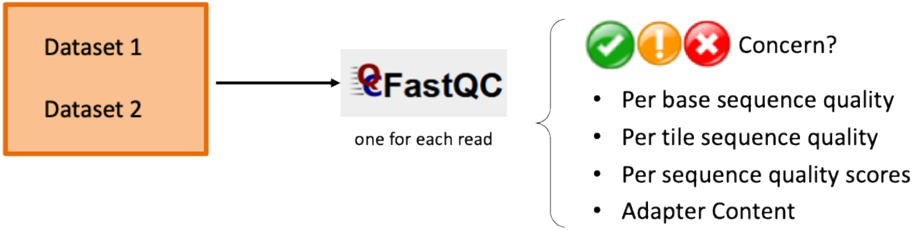
Special Note

We are running FASTQC interactively. This is running on the login node relatively quickly. This is because this alignment for these FASTQ files was only performed for a small portion of the chromosome 1. Later on, this will take a lot longer. Therefore, you will need to generate a script.

Running Parameters for FASTQC:

- 10G of memory is required
- 1 node, 2 tasks

Class Exercise #2: Assessing Universal Metrics


Dataset 1
Dataset 2
FastQC
one for each read
Concern?
• Per base sequence quality
• Per tile sequence quality
• Per sequence quality scores
• Adapter Content

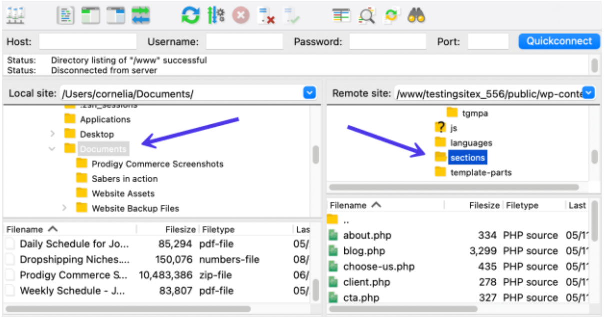
Grab the following folder from the location below.

```
/gpfs1/cl/mmg3320/course_materials/FASTQC_example
```

Viewing the HTML report from FASTQC

All of the following are solutions that allow students to transfer files between remote (i.e. VACC) and local (i.e. your laptop) servers.

An FTP application such as Filezilla



```
Status:   Directory listing of "/www" successful
Status:   Disconnected from server
```

Local site: /Users/cornelia/Documents/

- .zsh_sessions
- Applications
- Desktop
- Documents
  - Prodigy Commerce Screenshots
  - Sabers in action
  - Website Assets
  - Website Backup Files

Filename	Filesize	Filetype	Las
Daily Schedule for Jo...	85,294	pdf-file	05/
Dropshipping Niches....	150,076	numbers-file	08/
Prodigy Commerce S...	10,483,386	zip-file	06/
Weekly Schedule - J...	83,807	pdf-file	05/

Remote site: /www/testingsitex_556/public/wp-conte

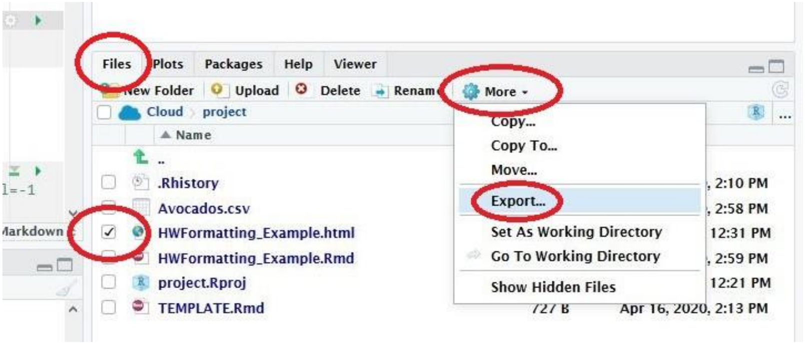
- tgmpa
- js
- languages
- sections
- template-parts

Filename	Filesize	Filetype	Last
..			
about.php	334	PHP source	05/1
blog.php	3,299	PHP source	05/1
choose-us.php	435	PHP source	05/1
client.php	278	PHP source	05/1
cta.php	327	PHP source	05/1

Viewing the HTML report from FASTQC

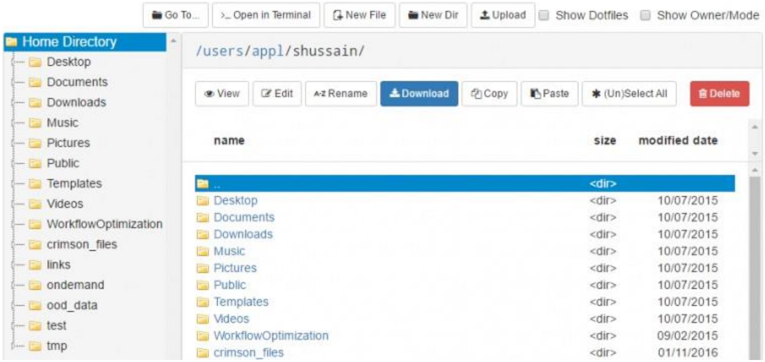
RStudio (via VACC-OOD)

You can export it or simply view it using RStudio



Viewing the HTML report from FASTQC

In File Explorer on OpenOnDemand, use the “Download” button



Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
WorkflowOptimization
crimson_files
links
ondemand
ood_data
test
tmp

/users/appl/shussain/

name	size	modified date
..	<dir>	
Desktop	<dir>	10/07/2015
Documents	<dir>	10/07/2015
Downloads	<dir>	10/07/2015
Music	<dir>	10/07/2015
Pictures	<dir>	10/07/2015
Public	<dir>	10/07/2015
Templates	<dir>	10/07/2015
Videos	<dir>	10/07/2015
WorkflowOptimization	<dir>	09/02/2015
crimson_files	<dir>	01/11/2016

Citation

This lesson has been developed by members of the teaching team at the Harvard Chan Bioinformatics Core (HBC). These are open access materials distributed under the terms of the Creative Commons Attribution license (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.