Introduction to RNA-seq for Differential Expression Analysis – 1

by Jiajia Li

Biological Data Science Institute

5 May 2025

Australian
National
University

About the trainer – Jiajia Li

Bioinformatics IT Support Officer at RSB and BDSI

- Master of Data Science (AI and Computational Modelling)
    - University of Canberra, Australia
- Bachelor of Engineering (Bioengineering)
    - Henan University, China

Past Workshops:

- Introduction to Linux and Variant Calling
- Introduction to Python
- Data Visualisation with Python
- Machine Learning with Python
- Introduction to Git and GitHub
- Introduction to NCI Gadi Supercomputer
- Introduction to Snakemake Workflow Language
- Data Management and Reproducible Research

Any workshop you are interested in so I can run it again in Semester 2?!!!

Workshop Goals

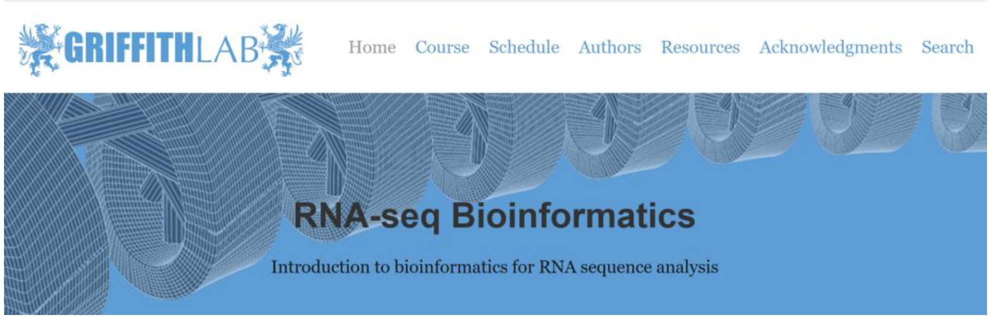
Due to the time limit and my capability, this workshop won’t go deep into the interpretation of results. But we will still do a little bit…

We will focus on understanding the **Differential Gene Expression** workflow and get the code running!! (the most exciting part…)

Bioinformatics and Biostatistics Drop-in Session

- Run **weekly** on **Tuesday** from **10-11am**, alternating locations between Robertson Building and JCSMR (shoot me an email if you don’t know where it’s on this week).
- If you have further questions about your own experiment, please welcome to join us at the drop-in session. We have an RNA-seq expert **Dr Zhi-Ping Feng** who can also provide help and advice.

Reference



RNA-seq Bioinformatics

Introduction to bioinformatics for RNA sequence analysis

https://rnabio.org/

This is a great resource from the Griffith Lab at the Washington University. I developed this workshop based on their material. Their course has a lot of useful information from theory to analysis and interpretation. My workshop is a simplified version of it.

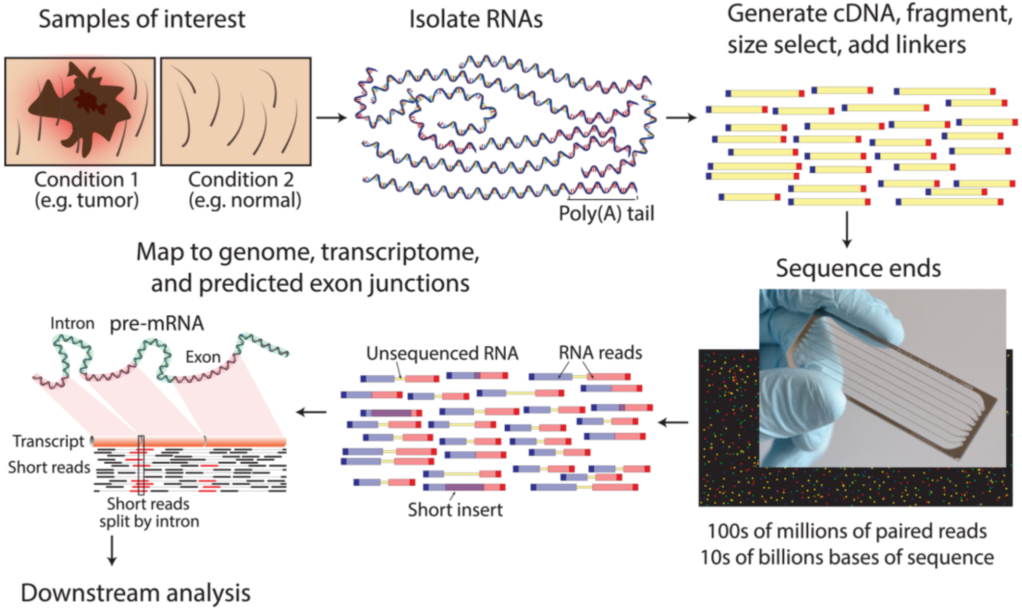
It is a good resource to look at if you want a more comprehensive understanding of RNA-sequencing techniques.

https://www.nature.com/articles/s41576-019-0150-2 This is a great paper for introductory level of RNA sequencing.

Learning Objectives of Today

- RNA-seq Analysis steps
- Different software
- Learn about our data
- Download data
- Learn about FASTQ/FASTA/GTF file format
- Learn about our reference genome
- Learn about Alignment
- Learn about Indexing
- Learn about software HISAT2
- Create an index for our reference genome
- Run FastQC and MultiQC on our data

RNA-sequencing General Steps


Samples of interest
Isolate RNAs
Generate cDNA, fragment, size select, add linkers
Condition 1 (e.g. tumor)
Condition 2 (e.g. normal)
Poly(A) tail
Map to genome, transcriptome, and predicted exon junctions
Sequence ends
Intron
pre-mRNA
Exon
Unsequenced RNA
RNA reads
Transcript
Short reads
Short reads split by intron
Short insert
100s of millions of paired reads
10s of billions bases of sequence
Downstream analysis

Step1. Experimental Design (very important)

Step2. Wet Lab

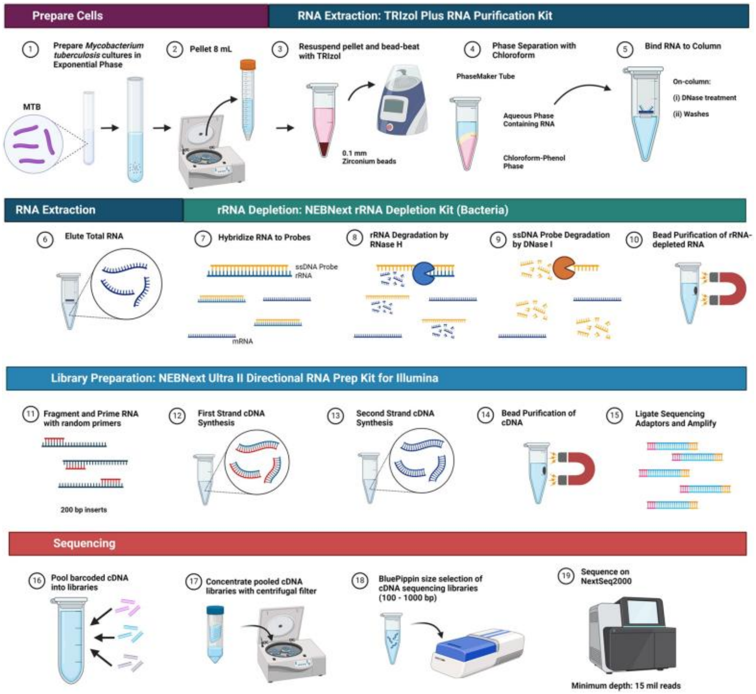
Step3. Dry Lab

Wet Lab Steps

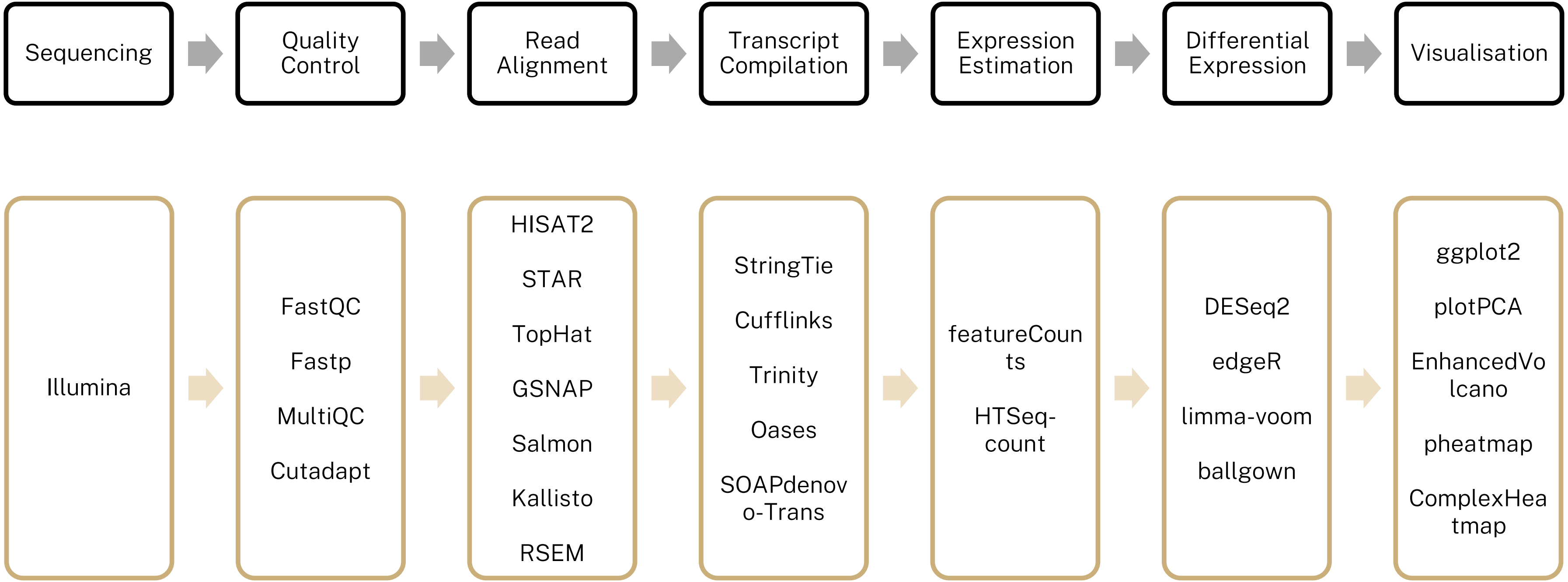
This might be done by the sequencing provider (e.g., BRF).

1) RNA extraction
2) RNA quality control
3) mRNA enrichment or rRNA depletion
4) RNA fragmentation
5) cDNA synthesis
6) Library preparation
7) Library quality control
8) Sequencing

Unfortunately, I won’t be able to guide you in these steps…

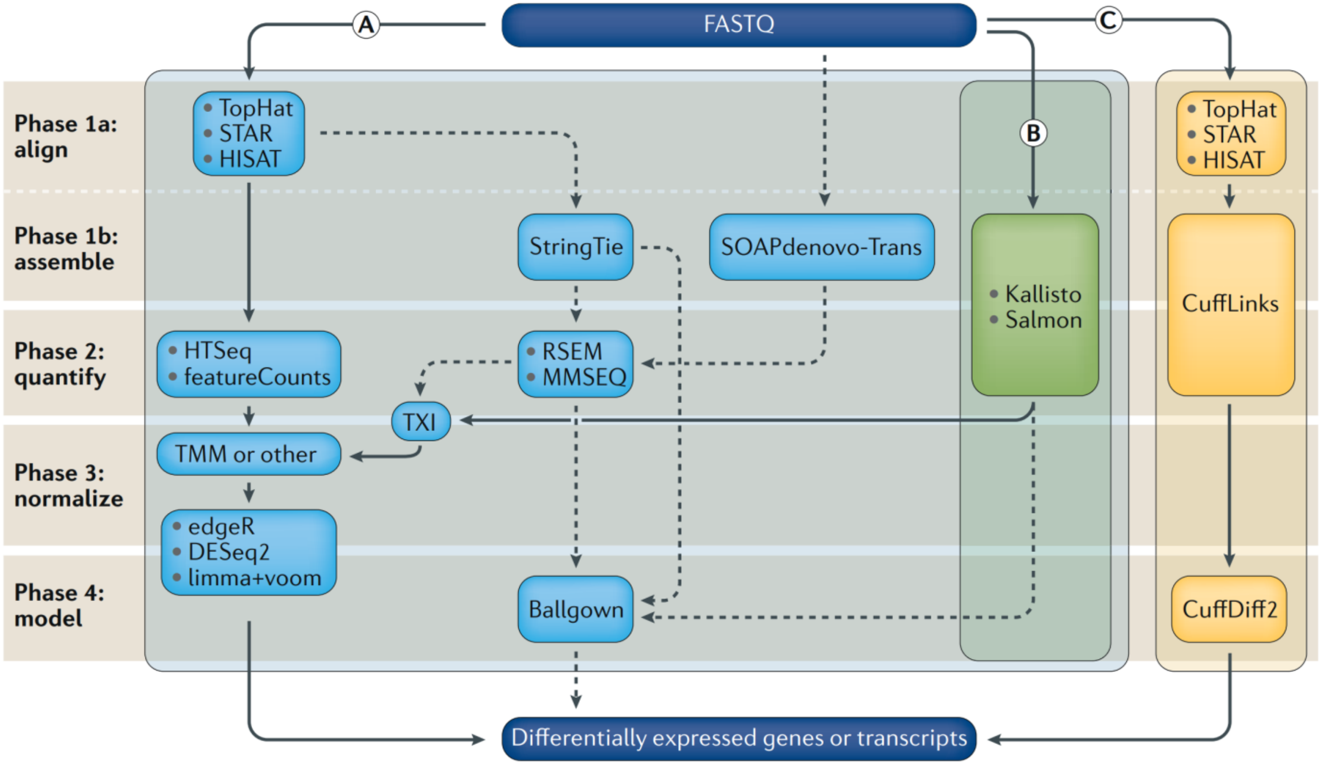

Prepare Cells
RNA Extraction: TRIzol Plus RNA Purification Kit
1 Prepare *Mycobacterium tuberculosis* cultures in Exponential Phase
MTB
2 Pellet 8 mL
3 Resuspend pellet and bead-beat with TRIzol
0.1 mm Zirconium beads
4 Phase Separation with Chloroform
PhaseMaker Tube
Aqueous Phase Containing RNA
Chloroform-Phenol Phase
5 Bind RNA to Column
On-column:
(i) DNase treatment
(ii) Washes
RNA Extraction
rRNA Depletion: NEBNext rRNA Depletion Kit (Bacteria)
6 Elute Total RNA
7 Hybridize RNA to Probes
ssDNA Probe
rRNA
mRNA
8 rRNA Degradation by RNase H
9 ssDNA Probe Degradation by DNase I
10 Bead Purification of rRNA-depleted RNA
Library Preparation: NEBNext Ultra II Directional RNA Prep Kit for Illumina
11 Fragment and Prime RNA with random primers
200 bp inserts
12 First Strand cDNA Synthesis
13 Second Strand cDNA Synthesis
14 Bead Purification of cDNA
15 Ligate Sequencing Adaptors and Amplify
Sequencing
16 Pool barcoded cDNA into libraries
17 Concentrate pooled cDNA libraries with centrifugal filter
18 BluePippin size selection of cDNA sequencing libraries (100 - 1000 bp)
19 Sequence on NextSeq2000
Minimum depth: 15 mil reads

Analysis Steps – Differential Expression



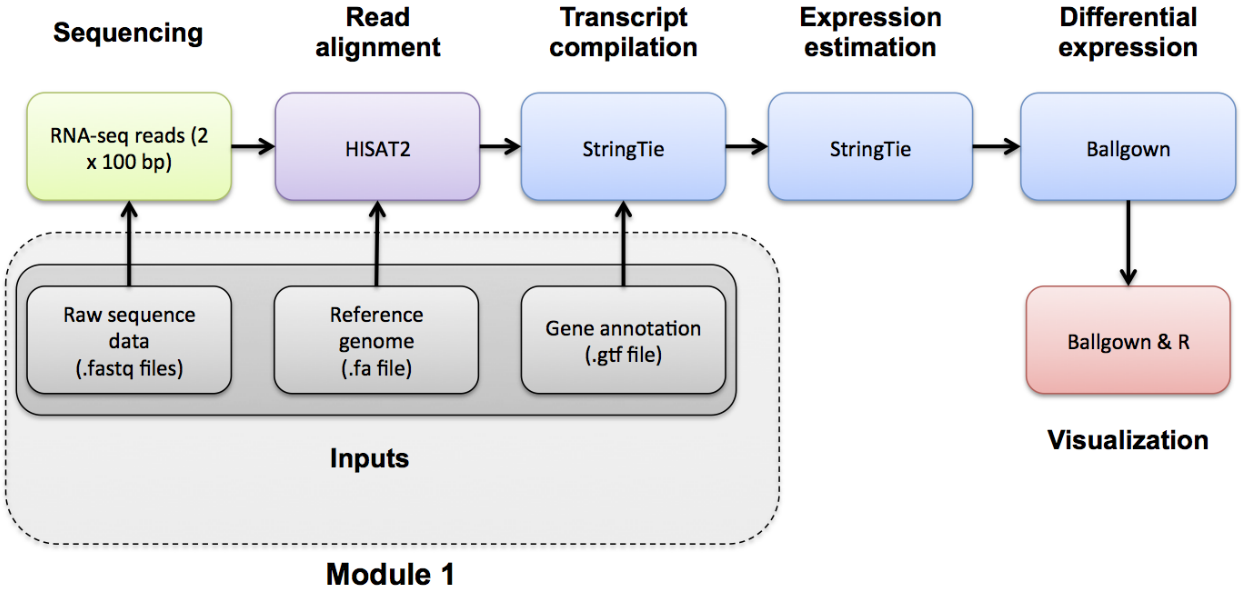
Sequencing	Quality Control	Read Alignment	Transcript Compilation	Expression Estimation	Differential Expression	Visualisation
Illumina	FastQC Fastp MultiQC Cutadapt	HISAT2 STAR TopHat GSNAP Salmon Kallisto RSEM	StringTie Cufflinks Trinity Oases SOAPdenovo-Trans	featureCounts HTSeq-count	DESeq2 edgeR limma-voom ballgown	ggplot2 plotPCA EnhancedVolcano pheatmap ComplexHeatmap

Analysis Steps – Differential Expression



Possible combinations of pipelines.

Pipeline 1 – HISAT2/StringTie/Ballgown



Sequencing

Read alignment

Transcript compilation

Expression estimation

Differential expression

RNA-seq reads (2 x 100 bp)

HISAT2

StringTie

StringTie

Ballgown

Raw sequence data (.fastq files)

Reference genome (.fa file)

Gene annotation (.gtf file)

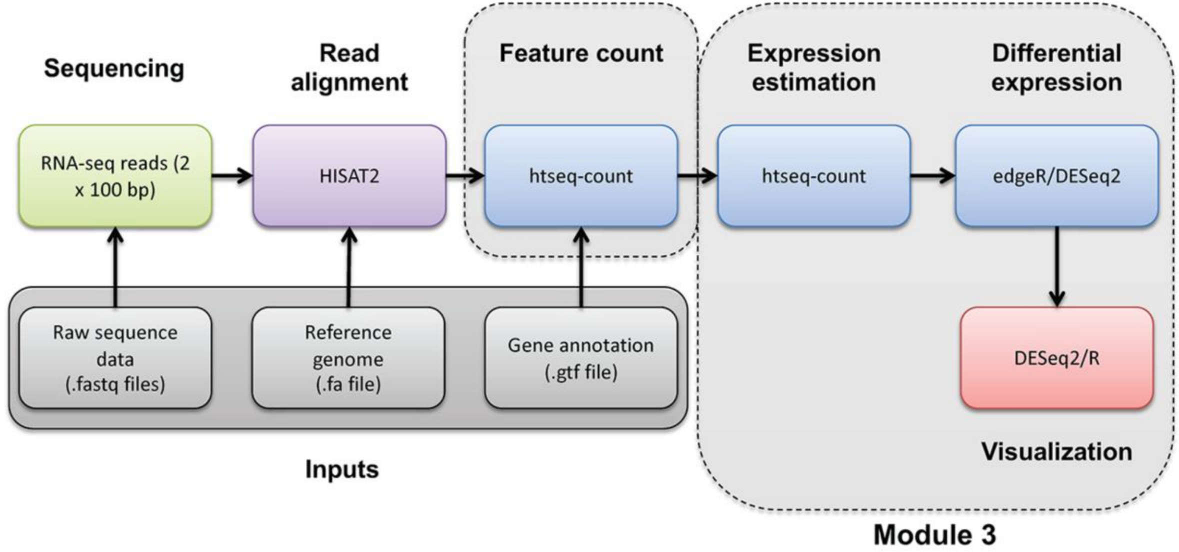
Inputs

Module 1

Ballgown & R

Visualization

Pipeline 2 – HISAT2/htseq-count/edgeR/DESeq2



The data

The test data consists of two commercially available RNA samples:

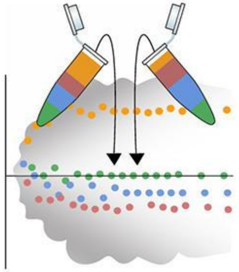
- Universal Human Reference (UHR)
- Human Brain Reference (HBR)

The UHR is total RNA isolated from a diverse set of **10 cancel cell lines** (breast, liver, cervix, testis, brain, skin, fatty tissue, histocyte, macrophage, T cell, B cell).

The HBR is total RNA isolated from the **brain of 23 Caucasians**, male and female, of varying age but mostly 60-80 years old.

In addition, a spike-in control was used. An aliquot of the ERCC ExFold RNA Spike-In Control Mixes was added to each sample.

ERCC ExFold RNA Spike-In Control Mixes



The spike-in consists of **92 transcripts** that are presented in known concentrations across a wide abundance range (from very few copies to many copies).

This range allows us to test the degree to which the RNA-seq assay (including all laboratory and analysis steps) accurately reflects the relative abundance of transcript species within a sample.

Because the concentration of transcripts in the spike-in control is known, if your analysis result couldn’t correctly reflect the concentration of the spike-in control, it means your experiment is not reliable.

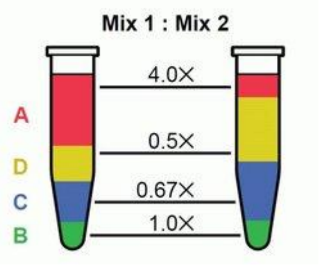
ERCC ExFold RNA Spike-In Control Mixes

There are 2 “mixes” of these transcripts to allow an assessment of differential expression output between samples if you put one mix in each of your two comparisons.

In our case, Mix1 was added to the UHR sample, and Mix2 was added to the HBR sample.

For example, Mix1 has more transcript A than Mix2. If the result shows the UHR sample has less transcript A than the HBR sample, it means your experiment is not reliable.

We also have 3 complete experiment replicates for each sample. This allows us to assess the technical variability of our overall process of producing RNA-seq data in the lab.



Library Preparation

This might be done by the service provider but it’s great if you understand the theory.

All libraries were prepared using low-throughput **TruSeq Stranded Total RNA Sample Prep Kit** libraries with Ribo-Zero Gold to **remove both cytoplasmic** and **mitochondrial rRNA**.

- Why low-throughput?
  - because we only have 6 samples

- TruSeq Stranded Total RNA Sample Prep Kit
  - Except total RNA, you can also use mRNA prep kit, then your library with only have mRNAs
  - Total RNA will contain mRNA, tRNA, rRNA, miRNA, lncRNA, etc.

- Why remove rRNA?
  - In cells, rRNA accounts for about 80%-90% of the total RNA
  - If we don’t remove rRNA, a large portion of our sequencing reads will be wasted on sequencing rRNA sequences

Library Preparation

Triplicate, indexed libraries were made starting with 100ng Agilent/Strategene Universal Human Reference total RNA and 100ng Ambion Human Brain Reference total RNA.

- Indexed – means that each library (or each sample) has a unique “barcode” attached to it.
- This allows you to pool multiple libraries together in a single sequencing run and then separate or “demultiplex” the reads back into their respective samples based on their indexing during analysis.

The UHR replicates received 2ul of 1:1000 ERCC Mix 1.
The HBR replicates received 2ul of 1:1000 ERCC Mix 2.

The libraries were quantified with KAPA Library Quantification qPCR and adjusted to the appropriate concentration for sequencing.

- Too concentrated or too diluted could negatively affect the sequencing result.

The triplicate, indexed libraries were then pooled prior to sequencing. Each pool of three replicate libraries were sequenced across 2 lanes of a HiSeq 2000 using paired-end sequence chemistry with **100bp** read lengths.

Samples

To summarise we have:

- UHR + ERCC Spike-In Mix1, Replicate 1
- UHR + ERCC Spike-In Mix1, Replicate 2
- UHR + ERCC Spike-In Mix1, Replicate 3

- HBR + ERCC Spike-In Mix2, Replicate 1
- HBR + ERCC Spike-In Mix2, Replicate 2
- HBR + ERCC Spike-In Mix2, Replicate 3

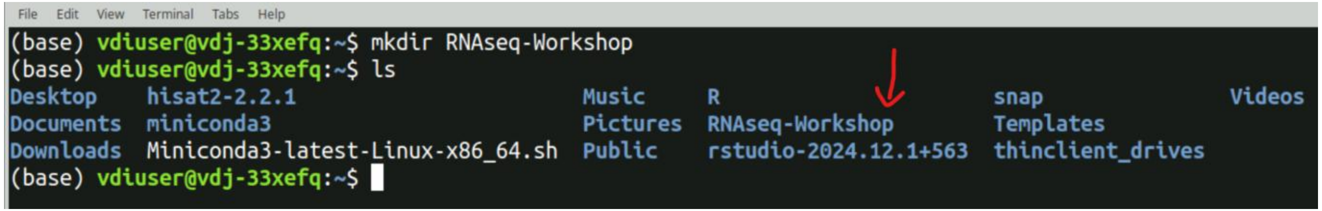
Each data set has a corresponding pair of FASTQ file (read1 and read2 of paired-end reads).

Download Data

Now, let's go to https://desktop.rc.nectar.org.au/ and open our ARDC Ubuntu machine.

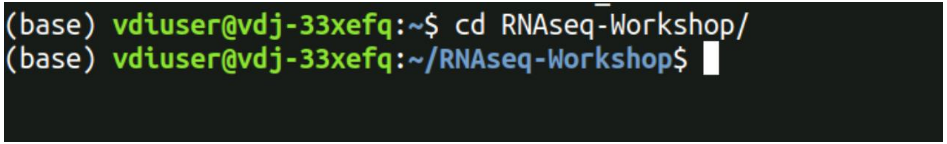
Open Terminal and create a new directory "RNAseq-Workshop".

To create a new folder in command line, we use command `mkdir RNAseq-Workshop`.



```
(base) vdiuser@vdj-33xefq:~$ mkdir RNAseq-Workshop
(base) vdiuser@vdj-33xefq:~$ ls
Desktop    hisat2-2.2.1                     Music     R                     snap              Videos
Documents  miniconda3                       Pictures  RNAseq-Workshop       Templates
Downloads  Miniconda3-latest-Linux-x86_64.sh  Public    rstudio-2024.12.1+563  thinclient_drives
(base) vdiuser@vdj-33xefq:~$
```

Go into this folder by running `cd RNAseq-Workshop`.

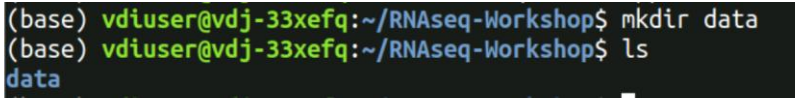


```
(base) vdiuser@vdj-33xefq:~$ cd RNAseq-Workshop/
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop$
```

Download Data

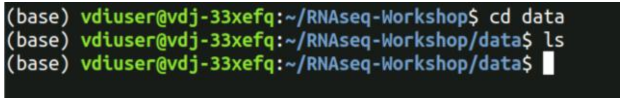
Then, we will create another folder “data” inside the “RNAseq-Workshop” folder.

`mkdir data`



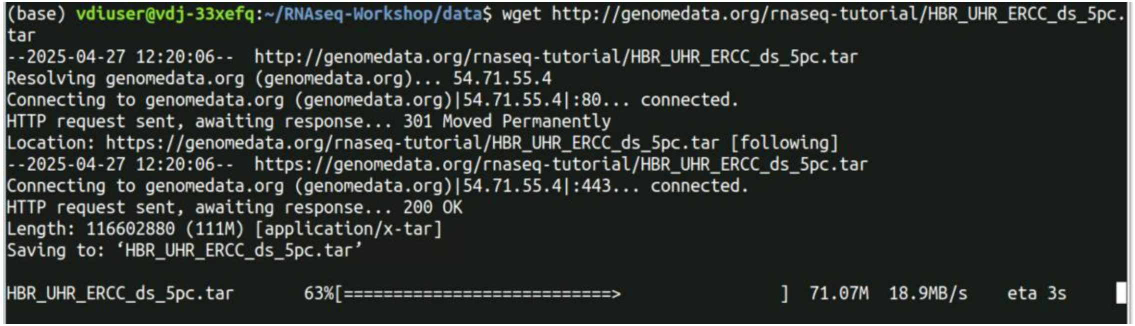
```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop$ mkdir data
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop$ ls
data
```

Go into this “data” folder, `cd data`.



```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop$ cd data
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ ls
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$
```

Run `wget http://genomedata.org/rnaseq-tutorial/HBR_UHR_ERCC_ds_5pc.tar` to download the .tar file, .tar is a type of compressed file.



```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ wget http://genomedata.org/rnaseq-tutorial/HBR_UHR_ERCC_ds_5pc.
tar
--2025-04-27 12:20:06--  http://genomedata.org/rnaseq-tutorial/HBR_UHR_ERCC_ds_5pc.tar
Resolving genomedata.org (genomedata.org)... 54.71.55.4
Connecting to genomedata.org (genomedata.org)|54.71.55.4|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://genomedata.org/rnaseq-tutorial/HBR_UHR_ERCC_ds_5pc.tar [following]
--2025-04-27 12:20:06--  https://genomedata.org/rnaseq-tutorial/HBR_UHR_ERCC_ds_5pc.tar
Connecting to genomedata.org (genomedata.org)|54.71.55.4|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 116602880 (111M) [application/x-tar]
Saving to: ‘HBR_UHR_ERCC_ds_5pc.tar’

HBR_UHR_ERCC_ds_5pc.tar       63%[===========================>                ]  71.07M  18.9MB/s    eta 3s
```

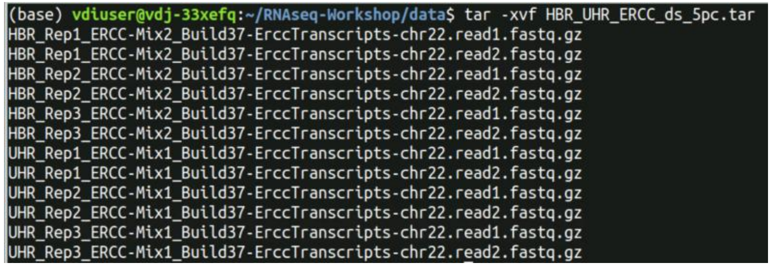
Download Data

Decompress the data by running
`tar -xvf HBR_UHR_ERCC_ds_5pc.tar`

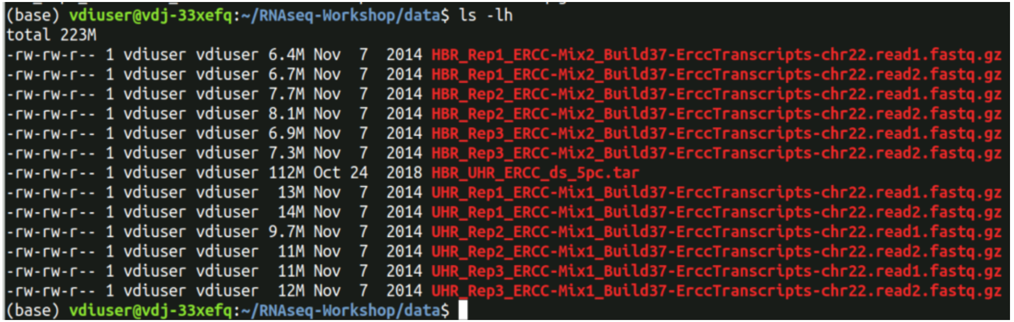
You should have total 12 files.

Check the size of your file using
`ls -lh`

They seem to be small and around 6-13MB in size, that is because these reads have been pre-filtered to those only mapped to chromosome 22.



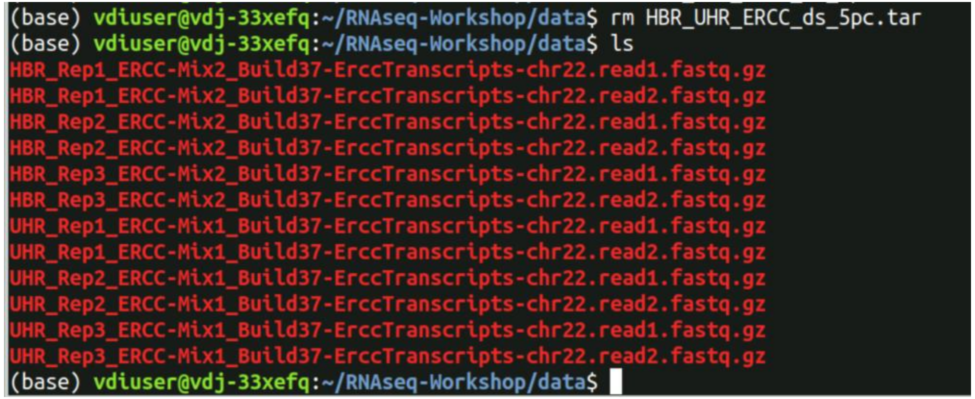
```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ tar -xvf HBR_UHR_ERCC_ds_5pc.tar
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
HBR_Rep3_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep3_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
UHR_Rep1_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep1_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
UHR_Rep2_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep2_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
UHR_Rep3_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep3_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
```



```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ ls -lh
total 223M
-rw-rw-r-- 1 vdiuser vdiuser 6.4M Nov  7  2014 HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
-rw-rw-r-- 1 vdiuser vdiuser 6.7M Nov  7  2014 HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
-rw-rw-r-- 1 vdiuser vdiuser 7.7M Nov  7  2014 HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
-rw-rw-r-- 1 vdiuser vdiuser 8.1M Nov  7  2014 HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
-rw-rw-r-- 1 vdiuser vdiuser 6.9M Nov  7  2014 HBR_Rep3_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
-rw-rw-r-- 1 vdiuser vdiuser 7.3M Nov  7  2014 HBR_Rep3_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
-rw-rw-r-- 1 vdiuser vdiuser 112M Oct 24  2018 HBR_UHR_ERCC_ds_5pc.tar
-rw-rw-r-- 1 vdiuser vdiuser  13M Nov  7  2014 UHR_Rep1_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
-rw-rw-r-- 1 vdiuser vdiuser  14M Nov  7  2014 UHR_Rep1_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
-rw-rw-r-- 1 vdiuser vdiuser 9.7M Nov  7  2014 UHR_Rep2_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
-rw-rw-r-- 1 vdiuser vdiuser  11M Nov  7  2014 UHR_Rep2_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
-rw-rw-r-- 1 vdiuser vdiuser  11M Nov  7  2014 UHR_Rep3_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
-rw-rw-r-- 1 vdiuser vdiuser  12M Nov  7  2014 UHR_Rep3_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$
```

Download Data

After we unzip the tar file, we don’t need it anymore and can delete it by running `rm HBR_UHR_ERCC_ds_5pc.tar`

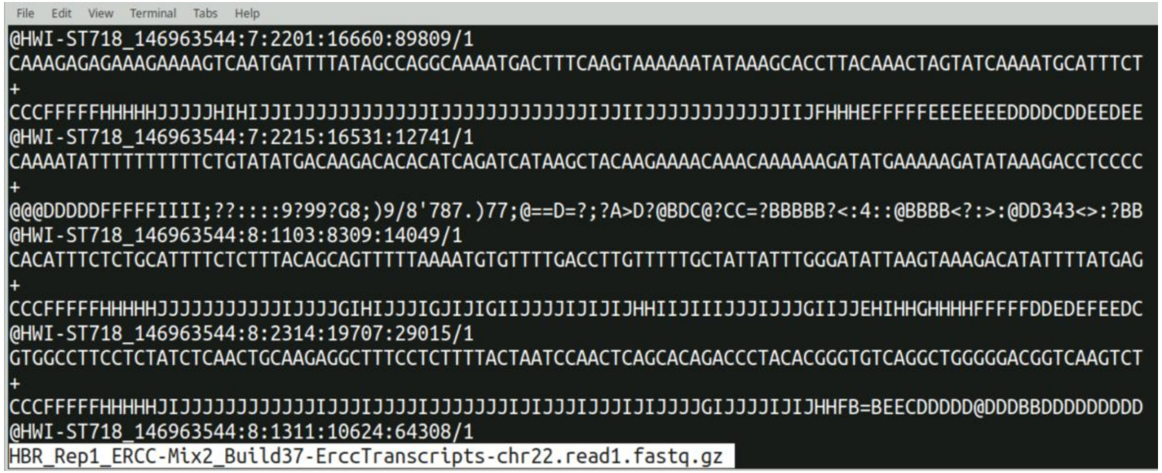


```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ rm HBR_UHR_ERCC_ds_5pc.tar
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ ls
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
HBR_Rep3_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep3_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
UHR_Rep1_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep1_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
UHR_Rep2_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep2_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
UHR_Rep3_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep3_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$
```

FASTQ File

The FASTQ file is widely used to store both **sequence data** and its **quality scores**. A FASTQ file consists of **multiple entries**, with each entry representing **a single sequence read**.

Let’s use our first file as an example to show you what a FASTQ file looks like.
Run `less HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz`



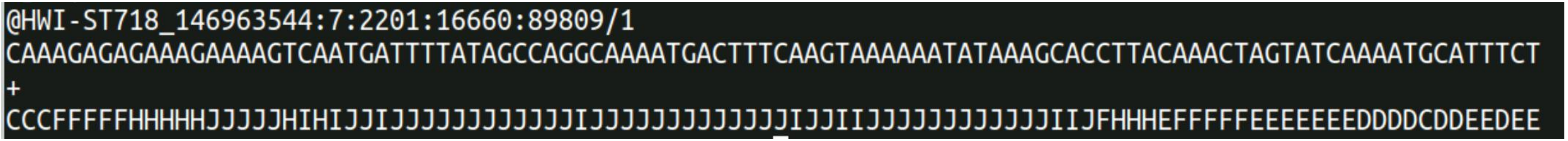
```
@HWI-ST718_146963544:7:2201:16660:89809/1
CAAAGAGAGAAAGAAAAGTCAATGATTTTATAGCCAGGCAAAATGACTTTCAAGTAAAAAATATAAAGCACCTTACAAACTAGTATCAAAATGCATTTCT
+
CCCFFFFFHHHHHJJJJJHIHIJJIJJJJJJJJJJJJJIJJJJJJJJJJJJJJIJJIIJJJJJJJJJJJJJIIJFHHHEFFFFFEEEEEEEDDDDCDDEEDEE
@HWI-ST718_146963544:7:2215:16531:12741/1
CAAAATATTTTTTTTTTCTGTATATGACAAGACACACATCAGATCATAAGCTACAAGAAAACAAACAAAAAAGATATGAAAAAGATATAAAGACCTCCCC
+
@@@DDDDDFFFFFIIII;??::::9?99?G8;)9/8'787.)77;@==D=?;?A>D?@BDC@?CC=?BBBBB?<:4::@BBBB<?:>:@DD343<>:?BB
@HWI-ST718_146963544:8:1103:8309:14049/1
CACATTTCTCTGCATTTTCTCTTTACAGCAGTTTTTAAAATGTGTTTTGACCTTGTTTTTGCTATTATTTGGGATATTAAGTAAAGACATATTTTATGAG
+
CCCFFFFFHHHHHJJJJJJJJJJJJIJJJJGIHIJJJIGJIJIGIIJJJJIJIJIJHHIIJIIIJJJIJJJGIIJJEHIHHGHHHHFFFFFDDEDEFEEDC
@HWI-ST718_146963544:8:2314:19707:29015/1
GTGGCCTTCCTCTATCTCAACTGCAAGAGGCTTTCCTCTTTTACTAATCCAACTCAGCACAGACCCTACACGGGTGTCAGGCTGGGGGACGGTCAAGTCT
+
CCCFFFFFHHHHHJIJJJJJJJJJJJJJIJJJIJJJJIJJJJJJJIJIJJJIJJJIJIJJJJGIJJJJIJIJHHFB=BEECDDDDD@DDDBBDDDDDDDDD
@HWI-ST718_146963544:8:1311:10624:64308/1
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
```

FASTQ File

The command `less` is a file pager that allows you to **view the contents of a file** in a terminal, and it works with .gz files which is another format of compressed files. You can use the "↑" and "↓" arrow keys to navigate, or the "space" bar to go to next page.

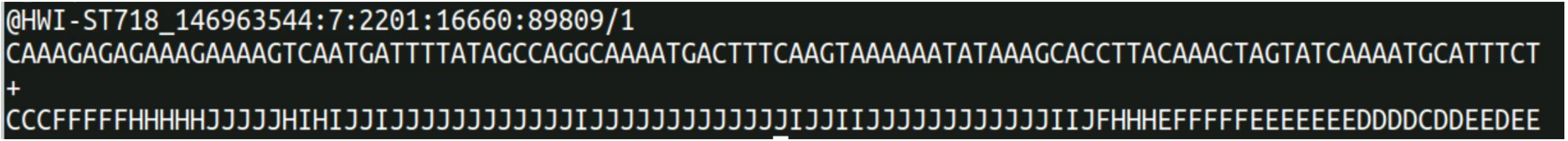
We can also run `zcat HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz | head –n 4` to only print out the first 4 lines of the file on the terminal.

`cat` is a command can print out all the content of a file to the screen, and `zcat` can print out the content of compressed files.



```
@HWI-ST718_146963544:7:2201:16660:89809/1
CAAAGAGAGAAAGAAAAGTCAATGATTTTATAGCCAGGCAAAATGACTTTCAAGTAAAAAATATAAAGCACCTTACAAACTAGTATCAAAATGCATTTCT
+
CCCFFFFFHHHHHJJJJJHIHIJJIJJJJJJJJJJJJIJJJJJJJJJJJJJIJJIIJJJJJJJJJJJJIIJFHHHEFFFFFEEEEEEEDDDDCDDEEDEE
```

FASTQ File



```
@HWI-ST718_146963544:7:2201:16660:89809/1
CAAAGAGAGAAAGAAAAGTCAATGATTTTATAGCCAGGCAAAATGACTTTCAAGTAAAAAATATAAAGCACCTTACAAACTAGTATCAAAATGCATTTCT
+
CCCFFFFFHHHHHJJJJJHIHIJJIJJJJJJJJJJJJIJJJJJJJJJJJJJIJJIIJJJJJJJJJJJJIIJFHHHEFFFFFEEEEEEEDDDDCDDEEDEE
```

Line 1 – Sequence Identifier

- Always start with an “@”.
- Metadata about the read, such as read name, sample info, or a unique identifier.

Line 2 – Sequence Line

- The actual sequence, A, T, G, C

Line 3 – Plus Sign “+”

- This line serves as a separator.

Line 4 – Quality Scores

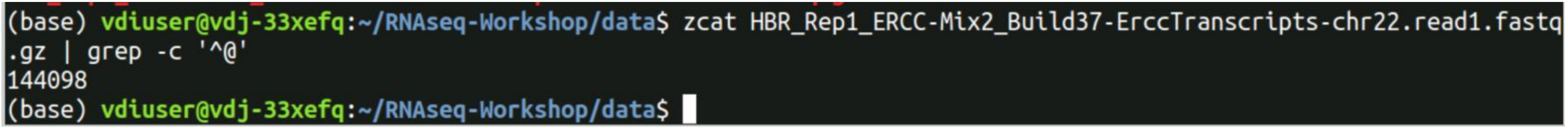
- A string of **ASCII characters** that represent the quality of each base in the sequence.
- Typically encoded using the **Phred quality score system**.

FASTQ File

Count how many reads in your FASTQ file. Since we know the first line starts with @ and it represents a sequence read. If we count the number of lines that start with @ we would know how many reads in the file.

To do so, we can run
`zcat HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz | grep –c ‘^@’`

- grep – a command search for patterns
- “^@” – search for lines start with @
- -c – count the number of lines that match the pattern



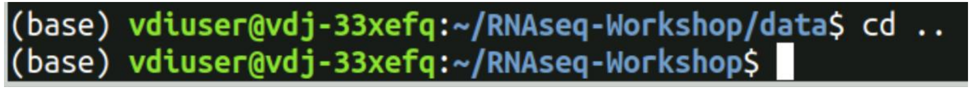
```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ zcat HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq
.gz | grep -c '^@'
144098
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ 
```

Reference Genome

In this analysis we will use the **GRCh38** version of the human genome from Ensemble (https://ftp.ensembl.org/pub/release-86/fasta/homo_sapiens/dna/). We are going to perform the analysis using only a single chromosome (**chr22**) and the ERCC spike-in to make it run faster.

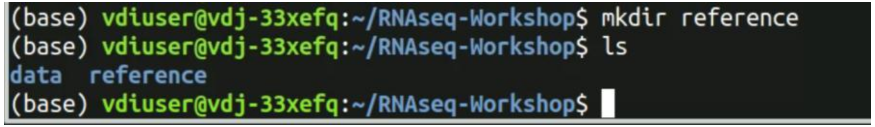
Let's go back to our "RNAseq-Workshop" folder and create a new folder called "reference".

Use `cd ..` to move one level up.



```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ cd ..
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop$
```

Then run `mkdir reference`.



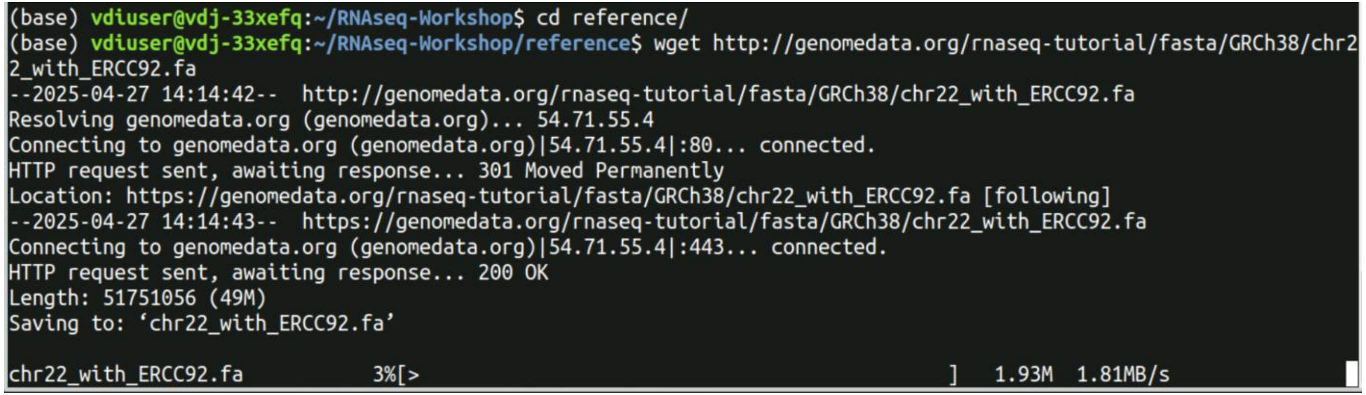
```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop$ mkdir reference
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop$ ls
data  reference
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop$
```

Reference Genome

Go into the newly created “reference” folder - `cd reference`

Run `wget http://genomedata.org/rnaseq-tutorial/fasta/GRCh38/chr22_with_ERCC92.fa` to download the references.

This is a simplified reference, contains only chr22 and ERCC transcripts.



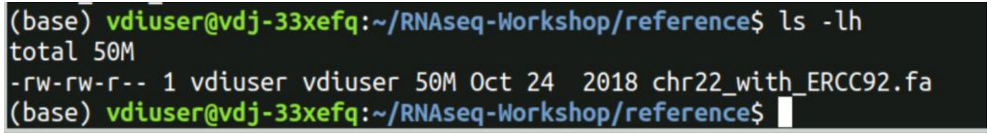
```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop$ cd reference/
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ wget http://genomedata.org/rnaseq-tutorial/fasta/GRCh38/chr2
2_with_ERCC92.fa
--2025-04-27 14:14:42--  http://genomedata.org/rnaseq-tutorial/fasta/GRCh38/chr22_with_ERCC92.fa
Resolving genomedata.org (genomedata.org)... 54.71.55.4
Connecting to genomedata.org (genomedata.org)|54.71.55.4|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://genomedata.org/rnaseq-tutorial/fasta/GRCh38/chr22_with_ERCC92.fa [following]
--2025-04-27 14:14:43--  https://genomedata.org/rnaseq-tutorial/fasta/GRCh38/chr22_with_ERCC92.fa
Connecting to genomedata.org (genomedata.org)|54.71.55.4|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 51751056 (49M)
Saving to: ‘chr22_with_ERCC92.fa’

chr22_with_ERCC92.fa          3%[>                                            ]   1.93M  1.81MB/s
```

Reference Genome

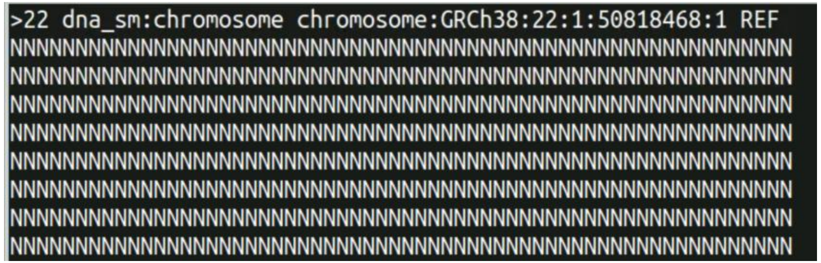
Use `ls -lh` to check the size of the downloaded reference genome.

".fa" is FASTA file format.



```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ ls -lh
total 50M
-rw-rw-r-- 1 vdiuser vdiuser 50M Oct 24  2018 chr22_with_ERCC92.fa
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ 
```

Use `less chr22_with_ERCC92.fa` to have a look of the file.



```
>22 dna_sm:chromosome chromosome:GRCh38:22:1:50818468:1 REF
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

"N" means undetermined nucleotides, there is a gap on chromosome 22, at least from this reference.

FASTA File Format

FASTA is a widely used file format for representing **nucleotide sequences** (DNA or RNA) or protein sequences.

A FASTA file can contain multiple entries, each entry has the following structure:

Header Line:

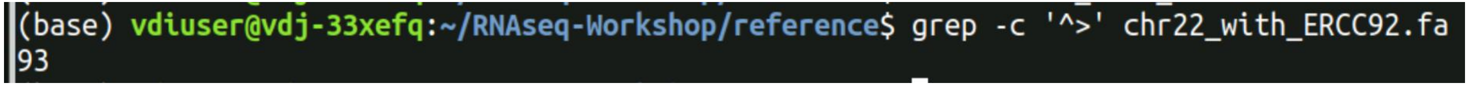
- Starts with an “>”
- Then description or identifier about the sequence.

Sequence Lines:

- A, T, G, C/A, U, G, C/amino acids
- The sequence can be broken into **multiple lines** to avoid very long lines, but the sequence itself should be continuous without spaces, gaps or any other characters.

FASTA File Format

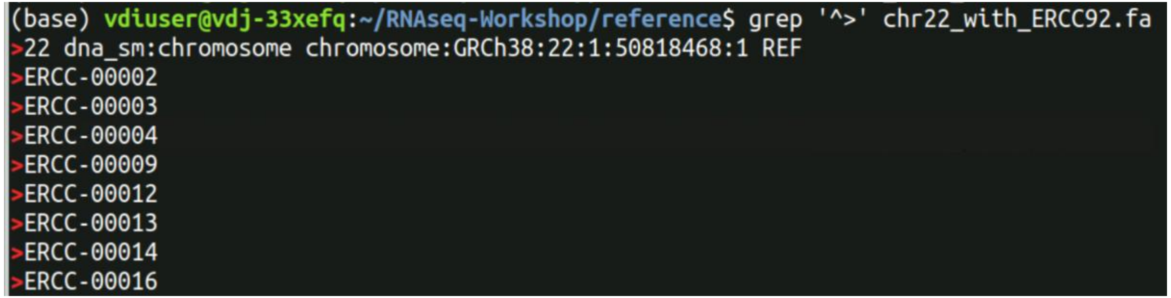
Check how many sequences in our reference genome.
`grep –c ‘^>’ chr22_with_ERCC92.fa`



```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ grep -c '^>' chr22_with_ERCC92.fa
93
```

93 sequences!! why?? usually for a chromosome it should be only one single sequence.

We can search all the lines start with “>” and print them out to see.
`grep ‘^>’ chr22_withERCC92.fa`



```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ grep '^>' chr22_with_ERCC92.fa
>22 dna_sm:chromosome chromosome:GRCh38:22:1:50818468:1 REF
>ERCC-00002
>ERCC-00003
>ERCC-00004
>ERCC-00009
>ERCC-00012
>ERCC-00013
>ERCC-00014
>ERCC-00016
```

Reference Genome

If you download the same chromosome 22 for GRCh38 from other resources, such as UCSC, the name shows in the “>” line could be different.

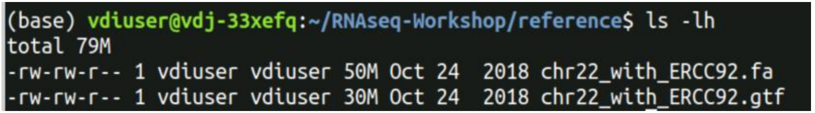
Remember that the names of your reference sequences (chromosomes) must match those in your annotation GTF files.

Known Gene/Transcript Annotations

In this tutorial we will use annotations from Ensembl (https://ftp.ensembl.org/pub/release-86/gtf/homo_sapiens/) for chromosome 22 only.

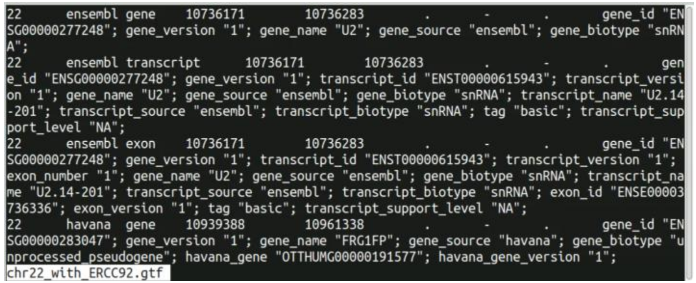
For time reasons, these are prepared for you.

Make sure you are in the “reference” folder and run:
`wget http://genomedata.org/rnaseq-tutorial/annotations/GRCh38/chr22_with_ERCC92.gtf`



```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ ls -lh
total 79M
-rw-rw-r-- 1 vdiuser vdiuser 50M Oct 24  2018 chr22_with_ERCC92.fa
-rw-rw-r-- 1 vdiuser vdiuser 30M Oct 24  2018 chr22_with_ERCC92.gtf
```

Use `less chr22_with_ERCC92.gtf` to have a look of the file content.

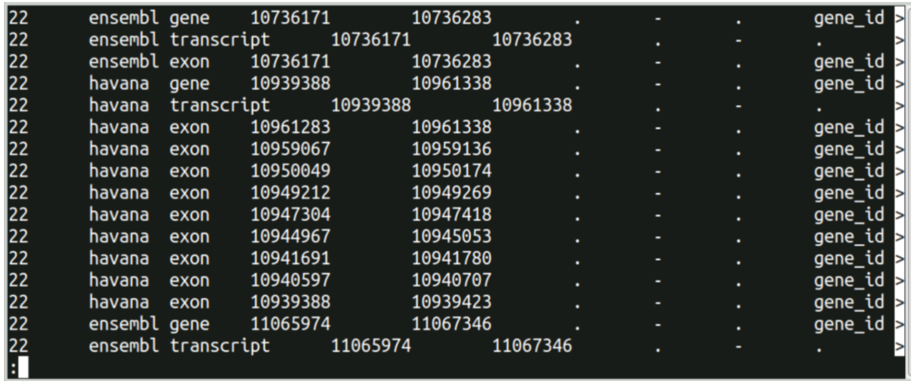


```
22      ensembl gene    10736171        10736283        .       -       .       gene_id "EN
SG00000277248"; gene_version "1"; gene_name "U2"; gene_source "ensembl"; gene_biotype "snRN
A";
22      ensembl transcript      10736171        10736283        .       -       .       gen
e_id "ENSG00000277248"; gene_version "1"; transcript_id "ENST00000615943"; transcript_versi
on "1"; gene_name "U2"; gene_source "ensembl"; gene_biotype "snRNA"; transcript_name "U2.14
-201"; transcript_source "ensembl"; transcript_biotype "snRNA"; tag "basic"; transcript_sup
port_level "NA";
22      ensembl exon    10736171        10736283        .       -       .       gene_id "EN
SG00000277248"; gene_version "1"; transcript_id "ENST00000615943"; transcript_version "1";
exon_number "1"; gene_name "U2"; gene_source "ensembl"; gene_biotype "snRNA"; transcript_na
me "U2.14-201"; transcript_source "ensembl"; transcript_biotype "snRNA"; exon_id "ENSE00003
736336"; exon_version "1"; tag "basic"; transcript_support_level "NA";
22      havana  gene    10939388        10961338        .       -       .       gene_id "EN
SG00000283047"; gene_version "1"; gene_name "FRG1FP"; gene_source "havana"; gene_biotype "u
nprocessed_pseudogene"; havana_gene "OTTHUMG00000191577"; havana_gene_version "1";
chr22_with_ERCC92.gtf
```

Known Gene/Transcript Annotations

It looks a bit messy because each line is too long, and it returns to the second line.

Use `less –S chr22_with_ERCC92.gtf` to view it in one line and in a left-right scrollable way.



```
22      ensembl gene    10736171        10736283        .       -       .       gene_id >
22      ensembl transcript      10736171        10736283        .       -       .       >
22      ensembl exon    10736171        10736283        .       -       .       gene_id >
22      havana  gene    10939388        10961338        .       -       .       gene_id >
22      havana  transcript      10939388        10961338        .       -       .       >
22      havana  exon    10961283        10961338        .       -       .       gene_id >
22      havana  exon    10959067        10959136        .       -       .       gene_id >
22      havana  exon    10950049        10950174        .       -       .       gene_id >
22      havana  exon    10949212        10949269        .       -       .       gene_id >
22      havana  exon    10947304        10947418        .       -       .       gene_id >
22      havana  exon    10944967        10945053        .       -       .       gene_id >
22      havana  exon    10941691        10941780        .       -       .       gene_id >
22      havana  exon    10940597        10940707        .       -       .       gene_id >
22      havana  exon    10939388        10939423        .       -       .       gene_id >
22      ensembl gene    11065974        11067346        .       -       .       gene_id >
22      ensembl transcript      11065974        11067346        .       -       .       >
:
```

But still not very good…

Try `cat chr22_with_ERCC92.gtf | column -t | less -p exon -S`, it looks better.

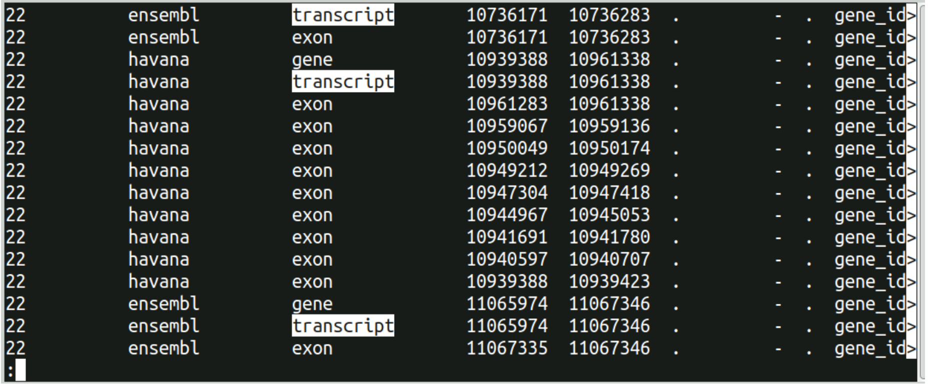
GTF File Format

GTF stands for **Gene Transfer Format** (sometimes called GFF version 2.5).

It’s a text file used in genomics to describe **gene structures** – like where genes, exons, transcripts are located on a genome.

Each line in a GTF file describes one feature (such as an exon, a transcript, a CDS, etc.).

Fields are tab-separated – **9 fields per line**.



```
22          ensembl     transcript      10736171  10736283  .         -  .  gene_id>
22          ensembl     exon            10736171  10736283  .         -  .  gene_id>
22          havana      gene            10939388  10961338  .         -  .  gene_id>
22          havana      transcript      10939388  10961338  .         -  .  gene_id>
22          havana      exon            10961283  10961338  .         -  .  gene_id>
22          havana      exon            10959067  10959136  .         -  .  gene_id>
22          havana      exon            10950049  10950174  .         -  .  gene_id>
22          havana      exon            10949212  10949269  .         -  .  gene_id>
22          havana      exon            10947304  10947418  .         -  .  gene_id>
22          havana      exon            10944967  10945053  .         -  .  gene_id>
22          havana      exon            10941691  10941780  .         -  .  gene_id>
22          havana      exon            10940597  10940707  .         -  .  gene_id>
22          havana      exon            10939388  10939423  .         -  .  gene_id>
22          ensembl     gene            11065974  11067346  .         -  .  gene_id>
22          ensembl     transcript      11065974  11067346  .         -  .  gene_id>
22          ensembl     exon            11067335  11067346  .         -  .  gene_id>
:
```

GTF File Format

Column	What it means	Example
1. Seqname	Chromosome or scaffold (chr1, chr22, MT, …)	chr22
2. Source	Where the annotation comes from (Ensembl, HAVANA, …)	ensembl
3. Feature	Types of feature (gene, transcript, exon, CDS, …)	exon
4. Start	Start position	10736171
5. End	End position (inclusive)	10736283
6. Score	A score (like confidence), can be . if not used	.
7. Strand	+ or – (which DNA strand)	-
8. Frame	Reading frame (0, 1, 2), or . is not applicable	.

9. Attributes	Extra information in key-value pairs, separated by semicolon “;”	gene_id "ENSG00000277248";
		gene_version "1";
		transcript_id "ENST00000615943";
		…

Indexing Reference Genome

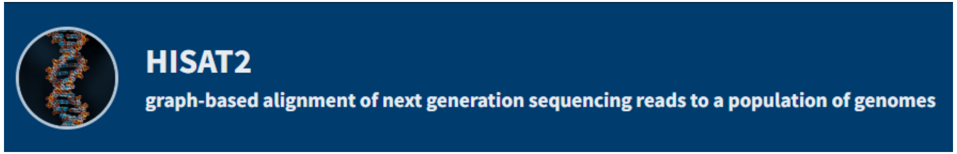
Indexing a reference genome is to prepare it for alignment tools can **find matching regions quickly**.

Think of it like:

- Without an index – searching for a sequence in the whole genome would be like reading through an entire book page by page.
- With an index – you have a table of contents and page numbers; you can jump directly to the right place.

Because genomes are normally huge, and it is very slow search the whole thing every time.

HISAT2



HISAT2

graph-based alignment of next generation sequencing reads to a population of genomes

HISAT2 stands for Hierarchical Indexing for Spliced Alignment of Transcripts (version 2).
It is a very fast and efficient aligner for **mapping short sequencing reads (like Illumina) to a reference genome.**

It’s especially good at:

- Handling **RNA-seq data** (reads that span **splice junctions**)
- Aligning to **large genomes** like human or mouse

It’s kind of the next generation tools after older aligners like **TopHat** and **Bowtie**.

Alignment

Alignment means – “Mapping sequencing reads back to a reference genome (or transcriptome) to find out **where they came from**”.

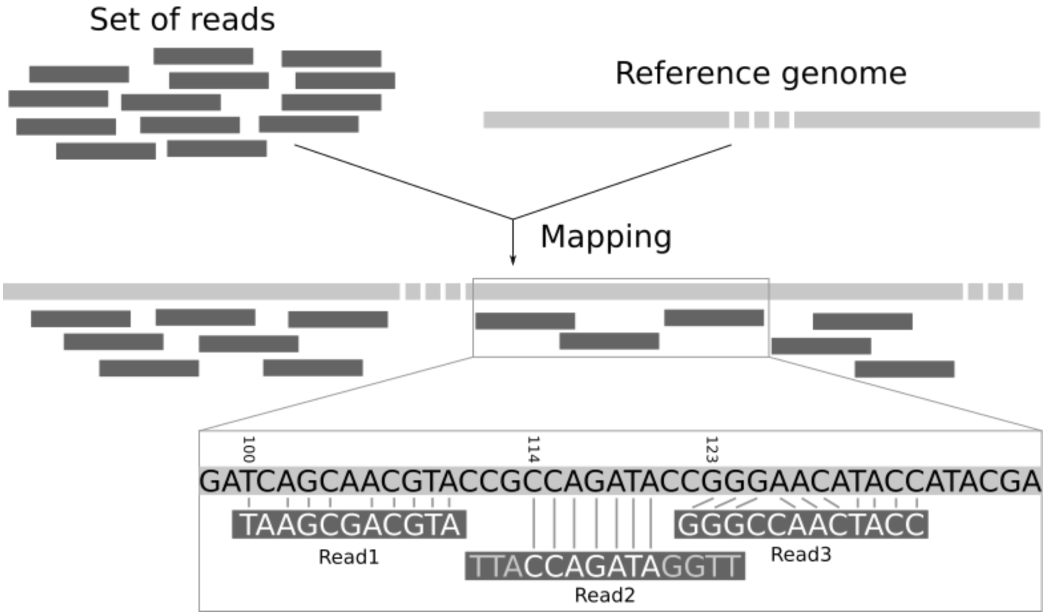
- In DNA-seq: reads are mapped **directly** to the genomic DNA. Straightforward and no introns to skip.
- In RNA-seq: reads are mapped to the genome, but they might **skip introns**, because RNA comes from **spliced transcripts**. Complex because reads may span exon-exon junctions.

You could map RNA-seq to a transcriptome directly as well, if you have a known transcriptome. Tools like **Salmon** and **Kallisto** can do it.

You could also perform **de novo transcriptome assembly** which assemble transcripts directly from the RNA-seq reads. Tools such as Trinity, rnaSPAdes, etc. can do it.

Alignment

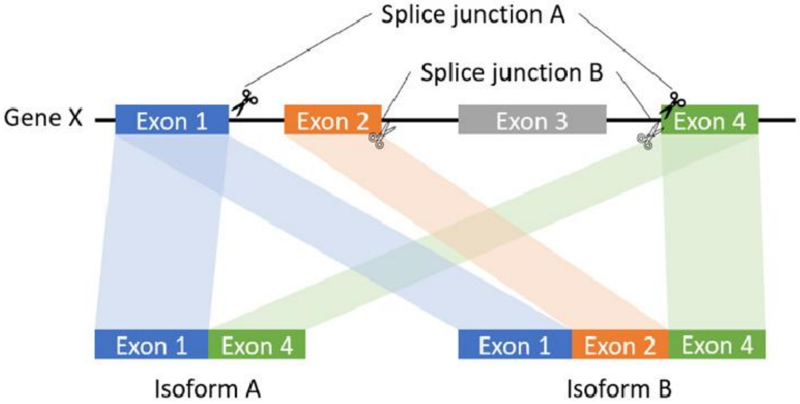
Illustration for DNA-seq mapping. For each read it normally fully matching to the reference genome.

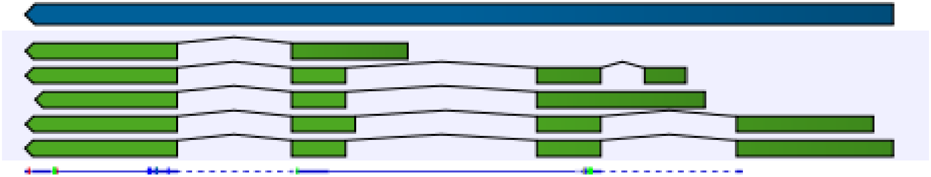


Alignment

For an RNA sequencing read, it could possibly be cut by introns and when it maps to genome, there will be a gap.

It’s not necessarily that each of your read could be cut by introns, if the exons are longer than your read length, it can stay intact.





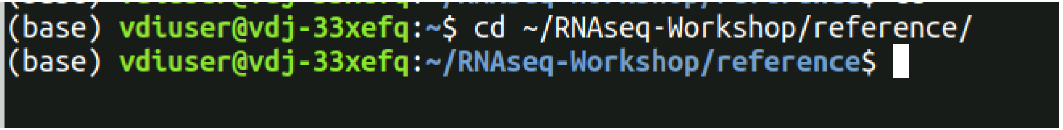
Create a HISAT2 index

Now let’s create an index for our reference, here we need to use a few python scripts from the HISAT2 software.

There are 3 steps:

- Extract splice sites
- Extract exons
- Build the index

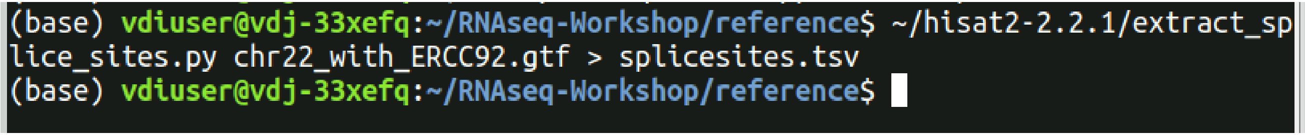
First, in your ARDC ubuntu machine, open Terminal. And navigate to where your reference located.
`cd ~/RNAseq-Workshop/reference`



```
(base) vdiuser@vdj-33xefq:~$ cd ~/RNAseq-Workshop/reference/
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ 
```

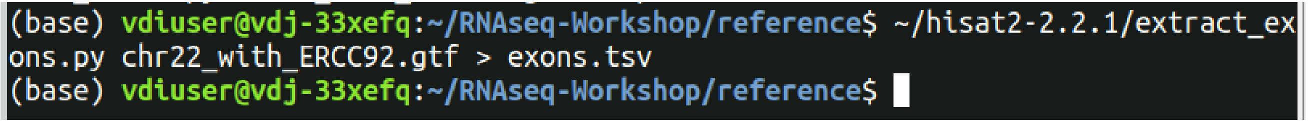
Create a HISAT2 index

Then run `~/hisat2-2.2.1/extract_splice_sites.py chr22_with_ERCC92.gtf > splicesites.tsv` to extract the splice sites from genome annotation file.



```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ ~/hisat2-2.2.1/extract_sp
lice_sites.py chr22_with_ERCC92.gtf > splicesites.tsv
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ 
```

Then run `~/hisat2-2.2.1/extract_exons.py chr22_with_ERCC92.gtf > exons.tsv` to extract exons.

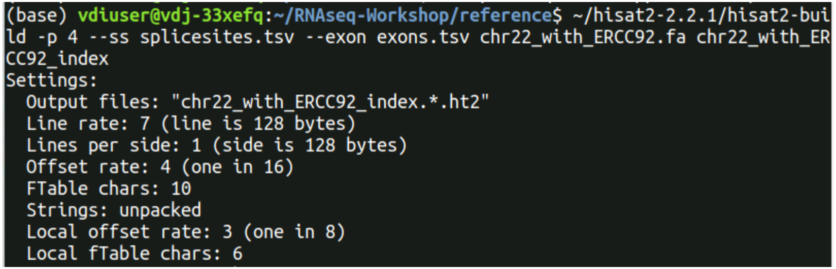


```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ ~/hisat2-2.2.1/extract_ex
ons.py chr22_with_ERCC92.gtf > exons.tsv
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ 
```

Create a HISAT2 index

Finally, we use the splice sites and exons information to build an index on the reference genome (FASTA file).

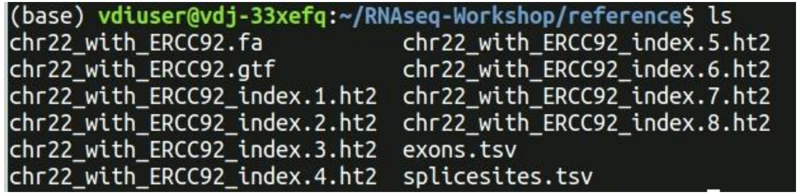
Run `~/hisat2-2.2.1/hisat2-build -p 4 --ss splicesites.tsv --exon exons.tsv chr22_with_ERCC92.fa chr22_with_ERCC92_index`



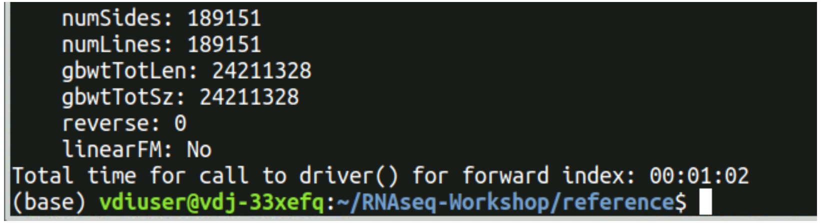
```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ ~/hisat2-2.2.1/hisat2-bui
ld -p 4 --ss splicesites.tsv --exon exons.tsv chr22_with_ERCC92.fa chr22_with_ER
CC92_index
Settings:
  Output files: "chr22_with_ERCC92_index.*.ht2"
  Line rate: 7 (line is 128 bytes)
  Lines per side: 1 (side is 128 bytes)
  Offset rate: 4 (one in 16)
  FTable chars: 10
  Strings: unpacked
  Local offset rate: 3 (one in 8)
  Local fTable chars: 6
```

-p to specify how many threads to use.
--ss to specify splice sites file.
--exon to specify exons file.

Use `ls` to check your result files!!



```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ ls
chr22_with_ERCC92.fa              chr22_with_ERCC92_index.5.ht2
chr22_with_ERCC92.gtf             chr22_with_ERCC92_index.6.ht2
chr22_with_ERCC92_index.1.ht2  chr22_with_ERCC92_index.7.ht2
chr22_with_ERCC92_index.2.ht2  chr22_with_ERCC92_index.8.ht2
chr22_with_ERCC92_index.3.ht2  exons.tsv
chr22_with_ERCC92_index.4.ht2  splicesites.tsv
```

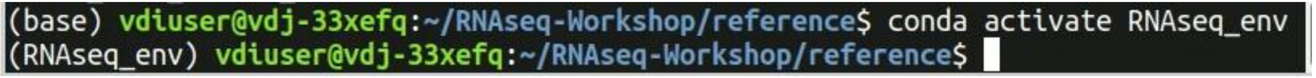


```
    numSides: 189151
    numLines: 189151
    gbwtTotLen: 24211328
    gbwtTotSz: 24211328
    reverse: 0
    linearFM: No
Total time for call to driver() for forward index: 00:01:02
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ 
```

Pre-alignment Quality Control

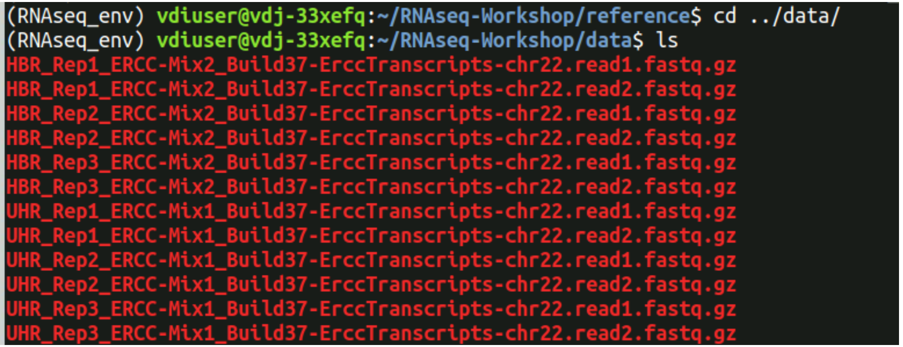
We can use **FastQC** to get a sense of your data quality before alignment. FastQC can generate a comprehensive report, showing several key metrics about the sequencing quality.

We have installed FastQC through Conda, so let’s activate our Conda environment first.
`conda activate RNAseq_env`



```
(base) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ conda activate RNAseq_env
(RNAseq_env) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$
```

Then we go to the “data” folder where our sequencing files locates.
`cd ../data`

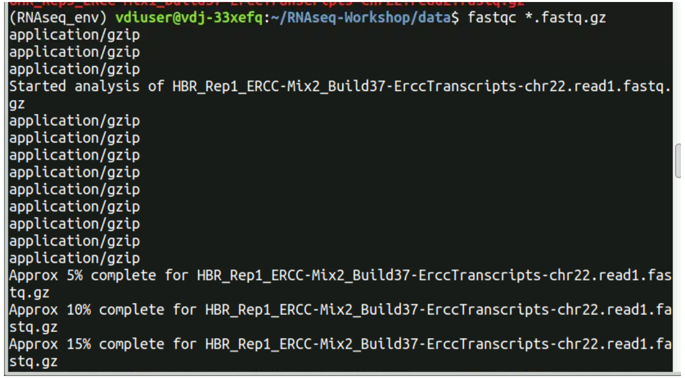


```
(RNAseq_env) vdiuser@vdj-33xefq:~/RNAseq-Workshop/reference$ cd ../data/
(RNAseq_env) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ ls
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
HBR_Rep3_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep3_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
UHR_Rep1_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep1_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
UHR_Rep2_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep2_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
UHR_Rep3_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep3_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
```

Pre-alignment Quality Control

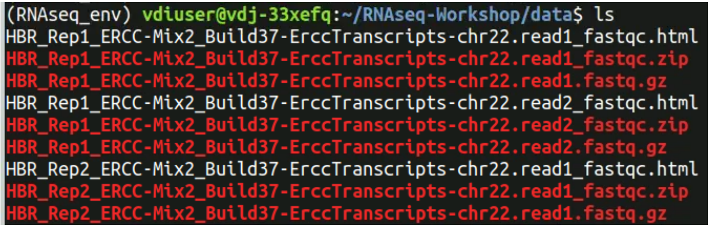
Then we can run `fastqc *.fastq.gz`

"*" is a **wildcard** in Linux, it will match all the files that end with .fastq.gz under your current directory.



```
(RNAseq_env) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ fastqc *.fastq.gz
application/gzip
application/gzip
application/gzip
Started analysis of HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.
gz
application/gzip
application/gzip
application/gzip
application/gzip
application/gzip
application/gzip
application/gzip
application/gzip
application/gzip
Approx 5% complete for HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fas
tq.gz
Approx 10% complete for HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fa
stq.gz
Approx 15% complete for HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fa
stq.gz
```

Use `ls` to check the result.



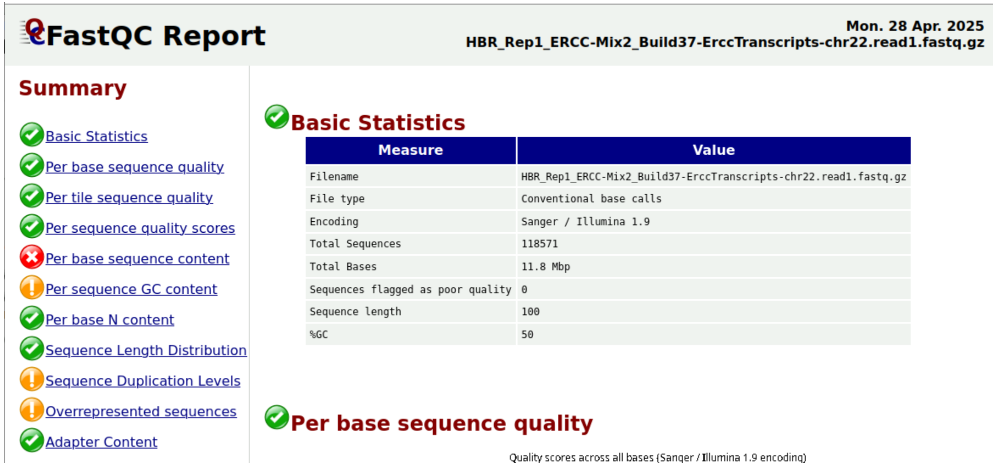
```
(RNAseq_env) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ ls
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1_fastqc.html
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1_fastqc.zip
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2_fastqc.html
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2_fastqc.zip
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1_fastqc.html
HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1_fastqc.zip
HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
```

Pre-alignment Quality Control

For each file, there is a HTML, and a ZIP file generated. The HTML can be opened by a web browser and will contain a report for this FASTQ file.

On your Desktop, open the Home folder, and navigate to where these HTML reports located. It should be under “**RNAseq-Workshop/data**”, then you can double click to open.

Let’s open the first one “HBR_Rep1_read1.HTML”



FastQC Report

Mon. 28 Apr. 2025
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz

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

✓ Basic Statistics

Measure	Value
Filename	HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	118571
Total Bases	11.8 Mbp
Sequences flagged as poor quality	0
Sequence length	100
%GC	50

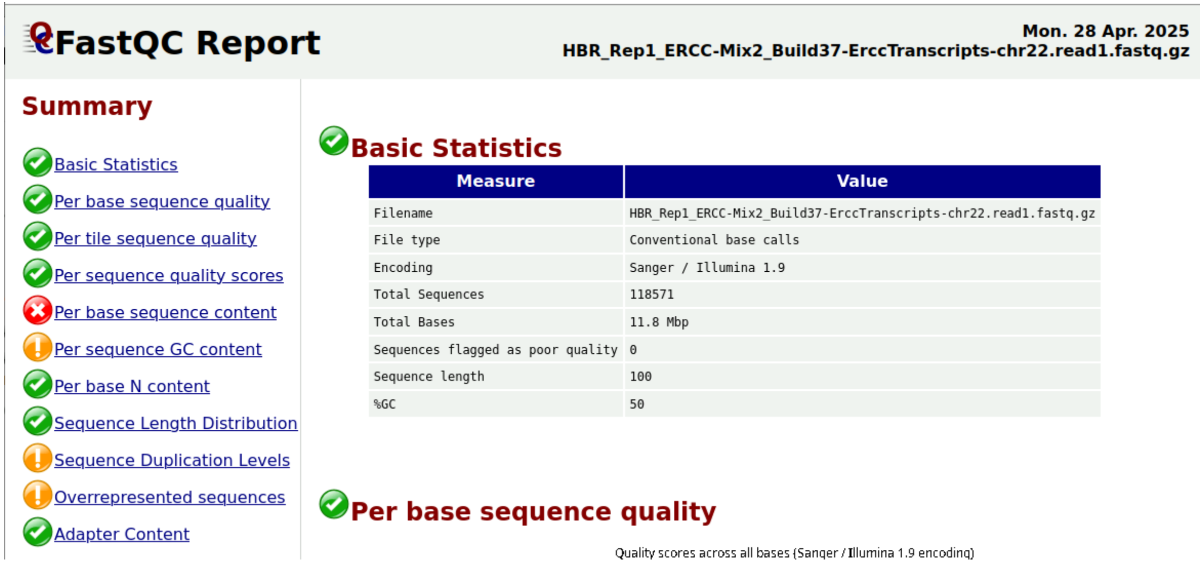
✓ Per base sequence quality

Quality scores across all bases (Sanger / Illumina 1.9 encoding)

Pre-alignment Quality Control

On the left pane, there are a few metrices. It can give you an overall idea of how your sample data looks like.

Green – Pass. Red – Fail. Yellow – Warning.



FastQC Report

Mon. 28 Apr. 2025
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz

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

✓ Basic Statistics

Measure	Value
Filename	HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	118571
Total Bases	11.8 Mbp
Sequences flagged as poor quality	0
Sequence length	100
%GC	50

✓ Per base sequence quality

Quality scores across all bases (Sanger / Illumina 1.9 encoding)

Our sample looks alright; we have 1 failing and 3 warnings.

Pre-alignment Quality Control

Here is a link of what a good Illumina short-read sequence should look like.

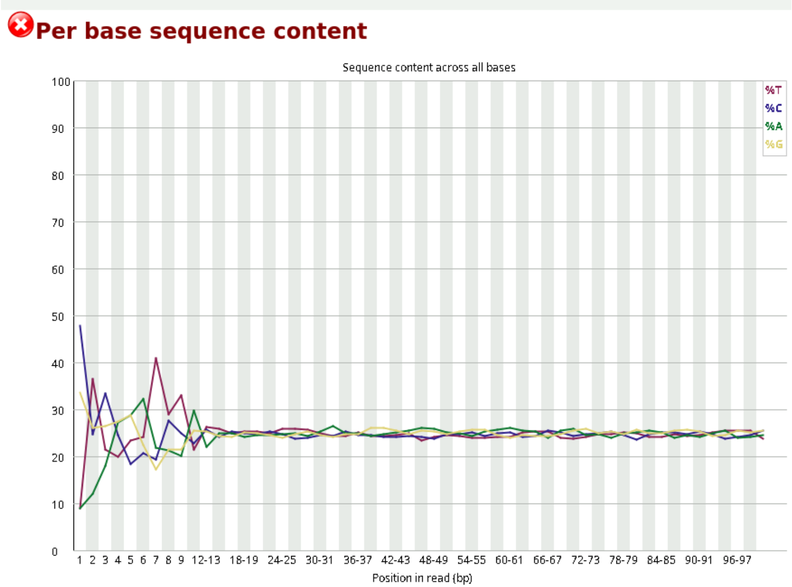
https://www.bioinformatics.babraham.ac.uk/projects/fastqc/good_sequence_short_fastqc.html

We are not going to go through all the matrices, but we will look at those with issues.

Per base sequence content

For each position across all your reads, it plots the **percentage** of A, T, G, and C bases at that position.

For example, on position 1 (base 1), there are 10% A, 10% T, 35% G, and 50% C.



Per base sequence content

In an ideal case, if your library is random, you expect the proportions of A, T, G, C, to be **pretty flat** and **similar** across positions.

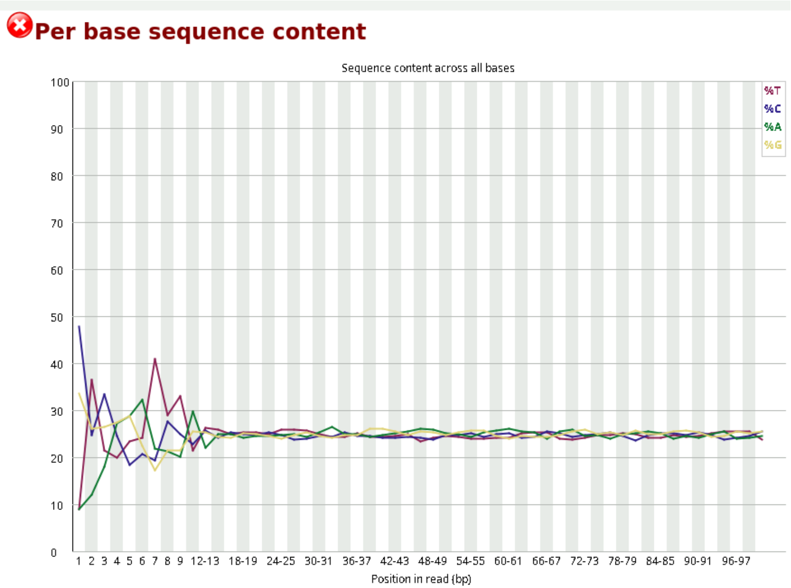
They should all be around 25% each.

This sample seems to have **small wobbles** at the beginning but becomes stable after **13** bases which is good.

Per base sequence content

Should we trim it?

Some biases are normal at the very start (1-10 bases) due to primers or **technical effects**, especially in RNA-seq or amplicon sequencing.



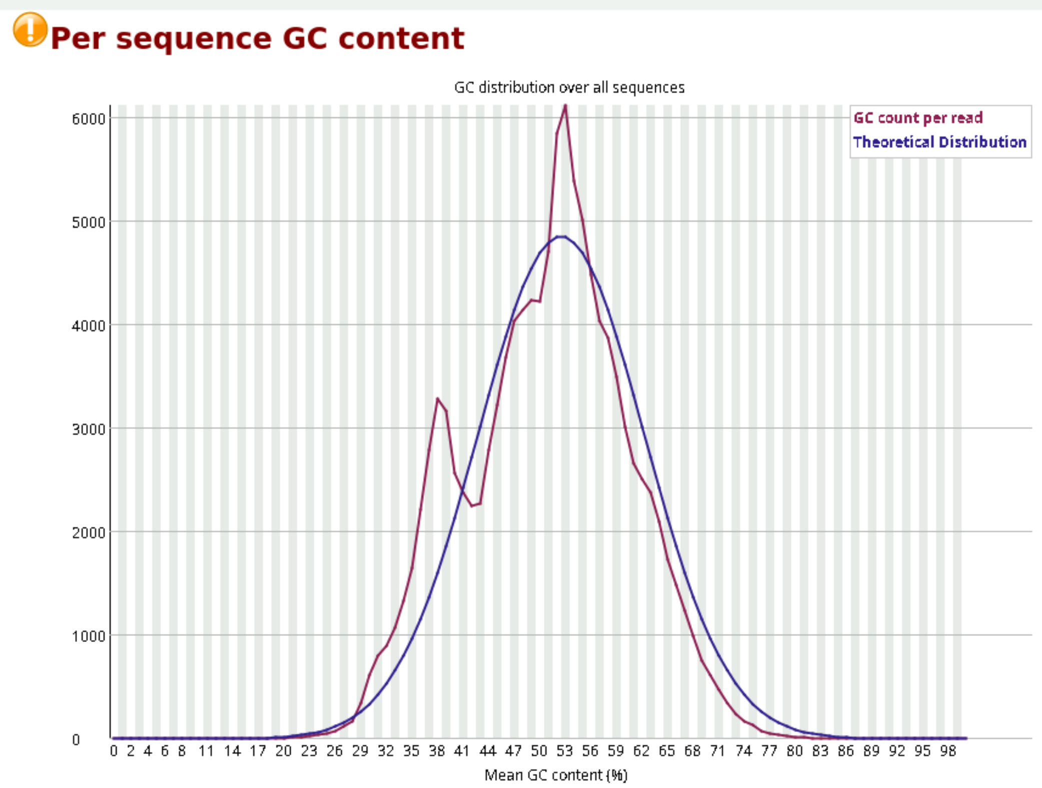
Per base sequence content

If the wobbles are small and settle down after, you probably don't have to trim it.

Many people just leave it because modern aligners (like HISAT2) **handle small biases pretty well**.

But if you want to be super clean or your analysis is sensitive (e.g. variant calling, assembly). You can trim the first 10-13 bases using **Cutadapt** or **Fastp**.

Per sequence GC content



A good plot of per sequence GC content should be:

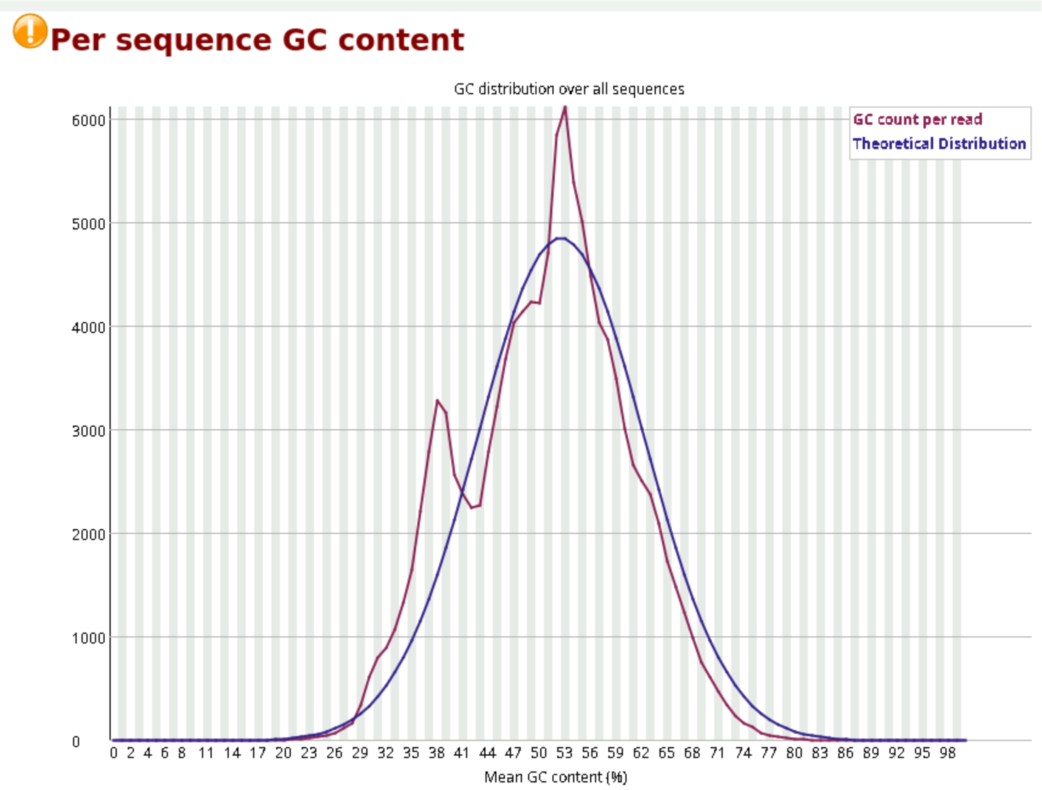
- smooth, bell-shaped curve (normal distribution)
- centred roughly around your species' expected GC content (Human 40-41%)
- the peak should match what you expect based on your organism

If your sample has:

- two peaks
- the peak is much higher or lower than you expected
- is very broad and flat.

Something might be wrong with your sample.

Per sequence GC content



Per sequence GC content

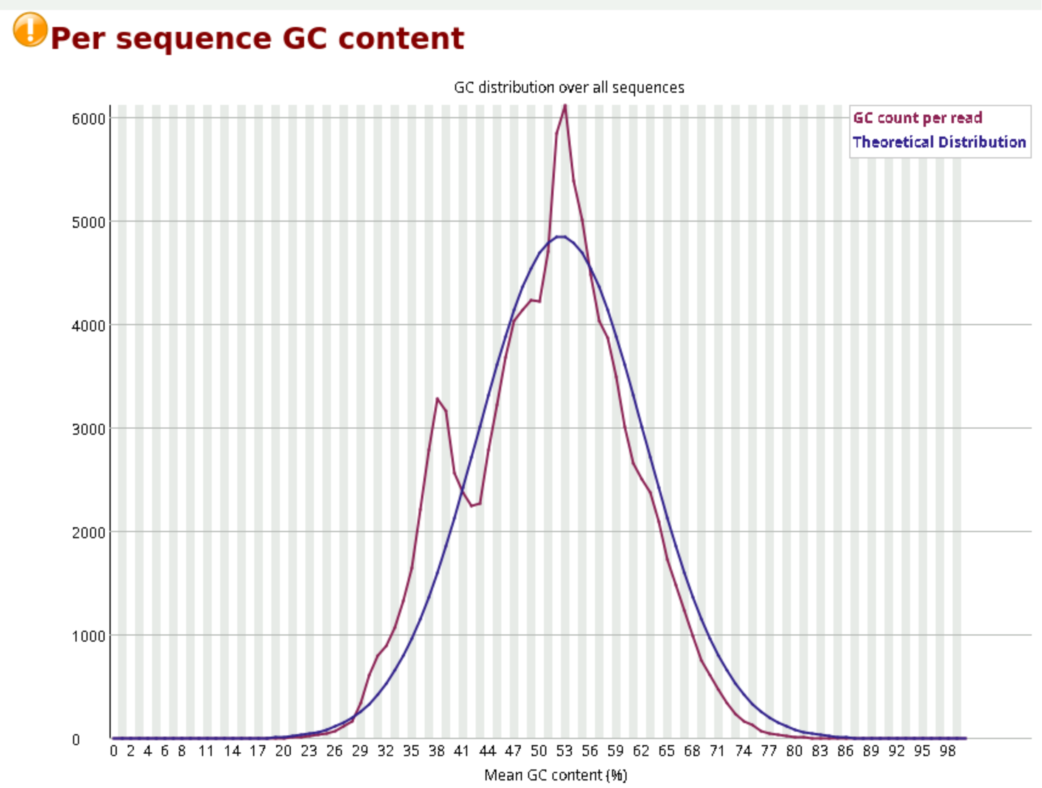
Our sample has two peaks:

1. peak at ~38% and has ~3200 sequences
2. peak at ~53% and has ~6000 sequences

Two peaks usually suggests two different populations of sequences in the data.

Any ideas why???

Per sequence GC content



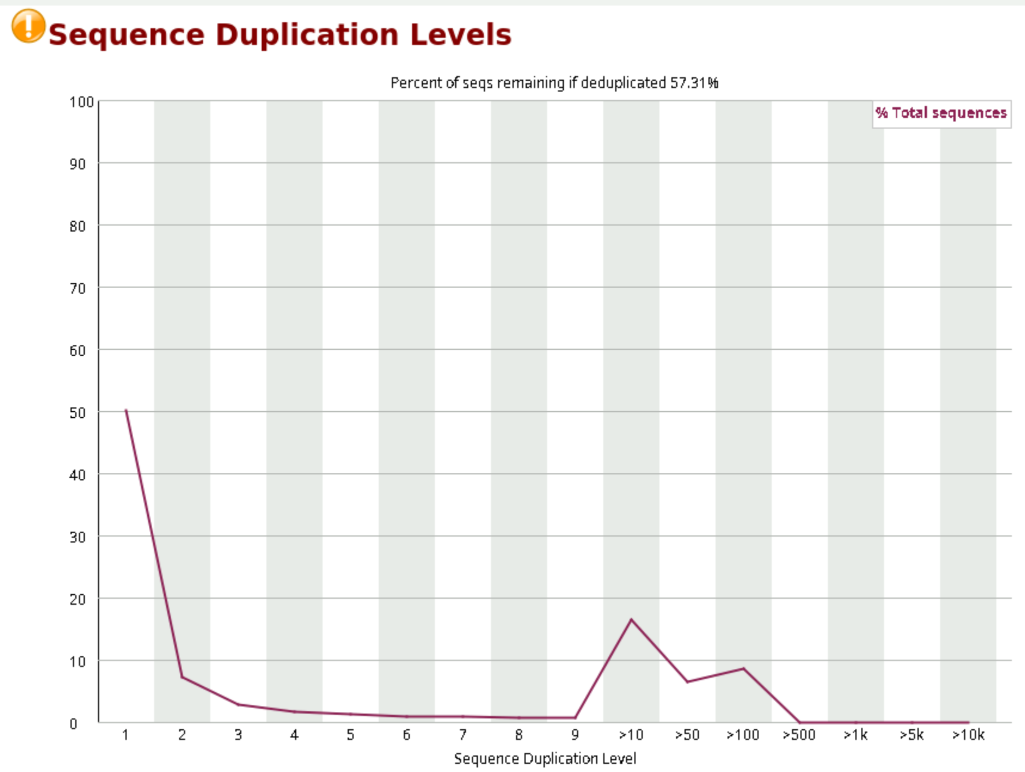
The first peak at 38% likely represents our **human brain cells**.

The second peak at 53% might represents:

- contamination from another organism - some bacteria or species have higher GC content
- adapters or technical artifacts - but adapters are usually small and wouldn’t cause a whole peak
- mixed library - two different samples got mixed during prep

It is our added ERCC spike-in control…!!! No need to worry.

Sequence duplication levels



What a good plot looks like?

For genomic DNA sequencing:

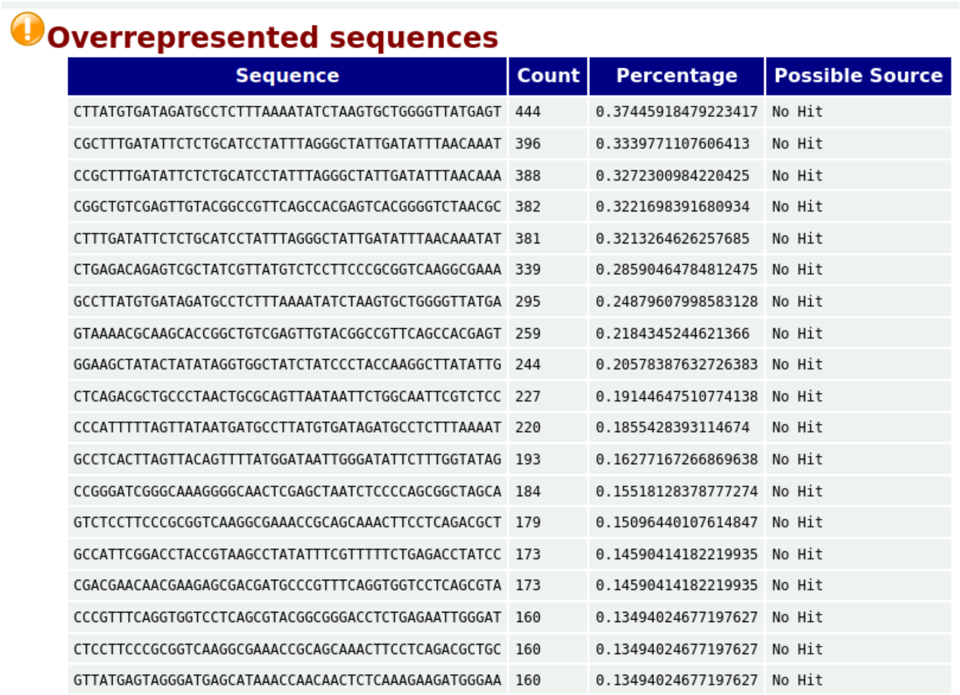
- you want most reads to be **unique** (seen only once)
- A small tail of duplicates is acceptable

For RNA-seq:

- **Higher duplication is normal** because highly expressed genes produce many identical reads
- especially true for small transcriptomes like brain-specific libraries

So, no worries for this warning as well!!

Overrepresented sequences



Overrepresented sequences

Sequence	Count	Percentage	Possible Source
CTTATGTGATAGATGCCTCTTTAAAATATCTAAGTGCTGGGGTTATGAGT	444	0.37445918479223417	No Hit
CGCTTTGATATTCTCTGCATCCTATTTAGGGCTATTGATATTTAACAAAT	396	0.3339771107606413	No Hit
CCGCTTTGATATTCTCTGCATCCTATTTAGGGCTATTGATATTTAACAAA	388	0.3272300984220425	No Hit
CGGCTGTCGAGTTGTACGGCCGTTCAGCCACGAGTCACGGGGTCTAACGC	382	0.3221698391680934	No Hit
CTTTGATATTCTCTGCATCCTATTTAGGGCTATTGATATTTAACAAATAT	381	0.3213264626257685	No Hit
CTGAGACAGAGTCGCTATCGTTATGTCTCCTTCCCGCGGTCAAGGCGAAA	339	0.28590464784812475	No Hit
GCCTTATGTGATAGATGCCTCTTTAAAATATCTAAGTGCTGGGGTTATGA	295	0.24879607998583128	No Hit
GTAAAACGCAAGCACCGGCTGTCGAGTTGTACGGCCGTTCAGCCACGAGT	259	0.2184345244621366	No Hit
GGAAGCTATACTATATAGGTGGCTATCTATCCCTACCAAGGCTTATATTG	244	0.20578387632726383	No Hit
CTCAGACGCTGCCCTAACTGCGCAGTTAATAATTCTGGCAATTCGTCTCC	227	0.19144647510774138	No Hit
CCCATTTTTAGTTATAATGATGCCTTATGTGATAGATGCCTCTTTAAAAT	220	0.1855428393114674	No Hit
GCCTCACTTAGTTACAGTTTTATGGATAATTGGGATATTCTTTGGTATAG	193	0.16277167266869638	No Hit
CCGGGATCGGGCAAAGGGGCAACTCGAGCTAATCTCCCCAGCGGCTAGCA	184	0.15518128378777274	No Hit
GTCTCCTTCCCGCGGTCAAGGCGAAACCGCAGCAAACTTCCTCAGACGCT	179	0.15096440107614847	No Hit
GCCATTCGGACCTACCGTAAGCCTATATTTCGTTTTTCTGAGACCTATCC	173	0.14590414182219935	No Hit
CGACGAACAACGAAGAGCGACGATGCCCGTTTCAGGTGGTCCTCAGCGTA	173	0.14590414182219935	No Hit
CCCGTTTCAGGTGGTCCTCAGCGTACGGCGGGACCTCTGAGAATTGGGAT	160	0.13494024677197627	No Hit
CTCCTTCCCGCGGTCAAGGCGAAACCGCAGCAAACTTCCTCAGACGCTGC	160	0.13494024677197627	No Hit
GTTATGAGTAGGGATGAGCATAAACCAACAACTCTCAAAGAAGATGGGAA	160	0.13494024677197627	No Hit

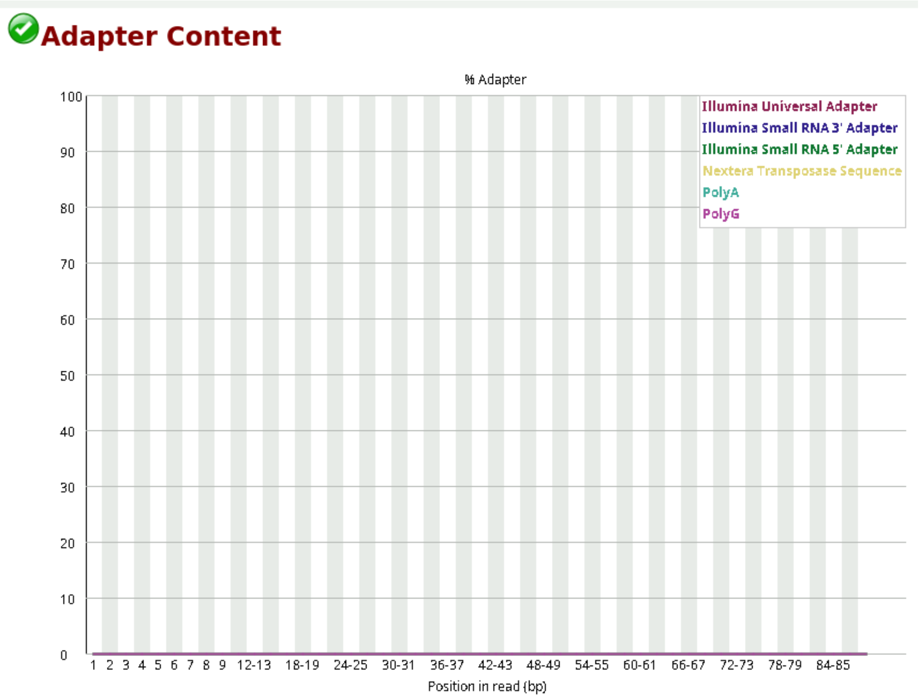
It lists any sequences that make up more than 0.1% of your total reads.

It also tries to BLAST them against a small database to guess what they are (e.g., adapters, ribosomal RNA, specific genes).

So, what do you think they could be??

We can also BLAST them to a bigger database like using the NCBI BLAST.

Adapter content



You should expect there is **no adapter** exists in your data.

Our sample doesn't have adapter contamination which is great.

So, for our sample, we don't have to trim the adapter. But in a lot of scenarios, you need to trim the adapter.

Sometimes, the RNA-seq service provider perform the adapter trimming and filtering for you so you receive a clean data from them.

When the adapter appears, it usually appears at the end of your sequence.

MultiQC

multiqc

Open-source tool to aggregate bioinformatic analyses results.

Now we have finished looking at one result from our data, but in total we have 12 FASTQ files we need to look at!!! A lot of work to do, isn't it??

Don't worry, MultiQC can help.

MultiQC is a program that gathers summary statistics from many bioinformatics tools (like FastQC). It combines all the reports from multiple samples into one big, beautiful, interactive HTML report.

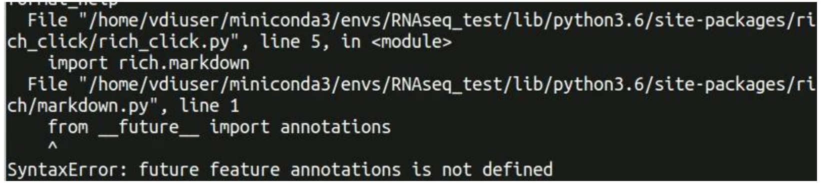
Go to directory `~/RNAseq-Workshop/data`.
Run `multiqc ./`

MultiQC

multiqc

Open-source tool to aggregate bioinformatic analyses results.

Unfortunately, we will encounter an error here because the python version we have here is not compatible to MultiQC.



```
  File "/home/vdiuser/miniconda3/envs/RNAseq_test/lib/python3.6/site-packages/ri
ch_click/rich_click.py", line 5, in <module>
    import rich.markdown
  File "/home/vdiuser/miniconda3/envs/RNAseq_test/lib/python3.6/site-packages/ri
ch/markdown.py", line 1
    from __future__ import annotations
    ^
SyntaxError: future feature annotations is not defined
```

MultiQC requires python >= 3.8

But we have python 3.6.10 when we get Conda to solve the environment for us when we install all the software together.

It’s okay, let’s install a newer version of python and try again.
`conda install -c conda-forge python=3.8`

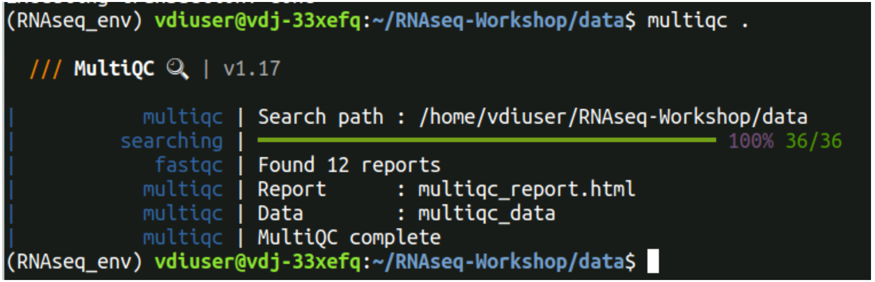
MultiQC

multiqc

Open-source tool to aggregate bioinformatic analyses results.

When it’s done, let’s try run MultiQC again.

`multiqc .`



```
(RNAseq_env) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ multiqc .

  /// MultiQC 🔍 | v1.17

|           multiqc | Search path : /home/vdiuser/RNAseq-Workshop/data
|         searching | ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 36/36
|            fastqc | Found 12 reports
|           multiqc | Report      : multiqc_report.html
|           multiqc | Data        : multiqc_data
|           multiqc | MultiQC complete
(RNAseq_env) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ █
```

Seems working… great!!

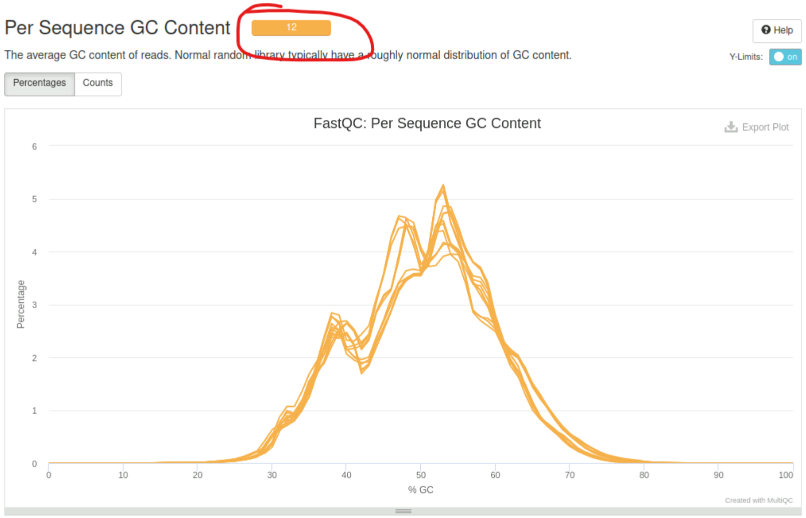
Then let’s use the file explorer to find the “multiqc_report.html” file and open it.

You can view all 12 results together, and if they are similar, we then can batch processing a standard trimming and cleaning process on them.

MultiQC

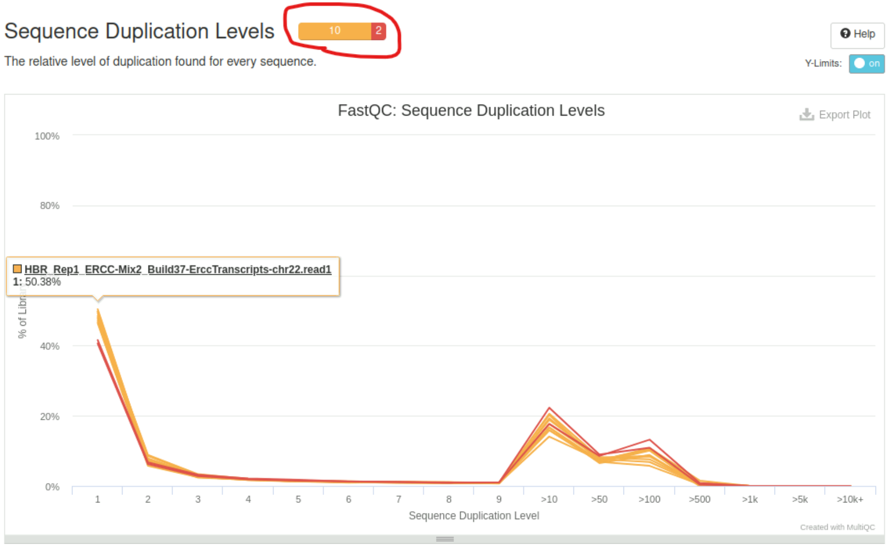
multiqc

Open-source tool to aggregate bioinformatic analyses results.



Per Sequence GC Content 12

The average GC content of reads. Normal random library typically have a roughly normal distribution of GC content.



Sequence Duplication Levels 10 2

The relative level of duplication found for every sequence.

Help
Y-Limits: on
FastQC: Sequence Duplication Levels
Export Plot
100%
80%
40%
20%
0%
% of Libr
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1
1: 50.38%
1
2
3
4
5
6
7
8
9
>10
>50
>100
>500
>1k
>5k
>10k+
Sequence Duplication Level
Created with MultiQC

They seem to have similar result which is great!

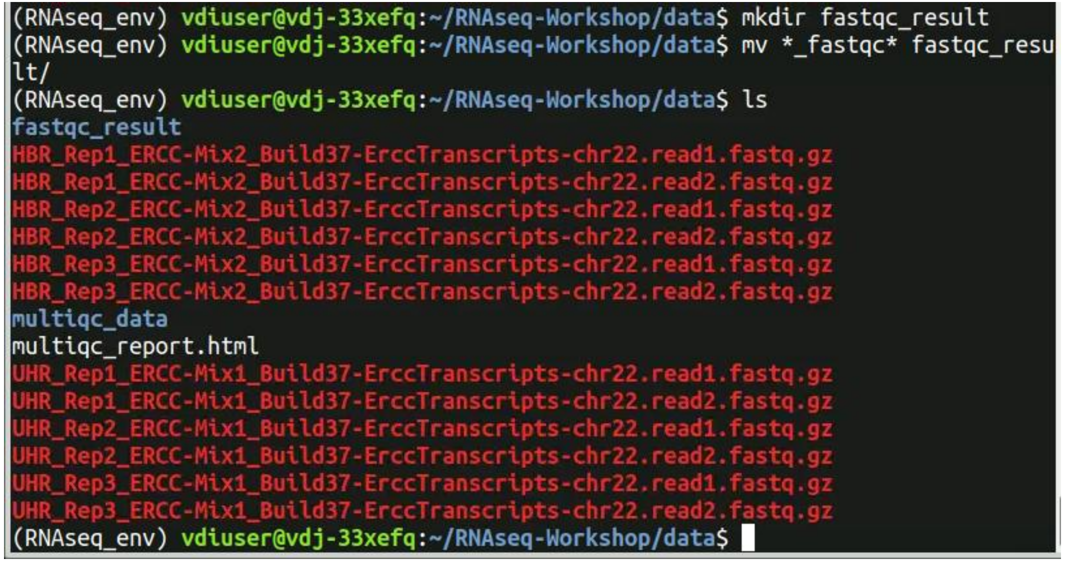
And if we ever want to trim them or clean them, we can do batch processing.

Clean up

We don’t want our data folder look too messy. So, let’s move all the QC result to a new folder.

Let’s create a folder “fastqc-result” under “data”:
`mkdir fastqc_result`

And move all the fastqc result into it:
`mv *_fastqc* fastqc_result`



```
(RNAseq_env) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ mkdir fastqc_result
(RNAseq_env) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ mv *_fastqc* fastqc_resu
lt/
(RNAseq_env) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$ ls
fastqc_result
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep1_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep2_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
HBR_Rep3_ERCC-Mix2_Build37-ErccTranscripts-chr22.read1.fastq.gz
HBR_Rep3_ERCC-Mix2_Build37-ErccTranscripts-chr22.read2.fastq.gz
multiqc_data
multiqc_report.html
UHR_Rep1_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep1_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
UHR_Rep2_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep2_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
UHR_Rep3_ERCC-Mix1_Build37-ErccTranscripts-chr22.read1.fastq.gz
UHR_Rep3_ERCC-Mix1_Build37-ErccTranscripts-chr22.read2.fastq.gz
(RNAseq_env) vdiuser@vdj-33xefq:~/RNAseq-Workshop/data$
```

Trimming and filtering FASTQ data

Normally, it is a standard step to trim any adapters or primers that exist in your data and filter out low quality reads.

But our data looks clean so we will skip this step.

Tools for trimming and filtering:

- Cutadapt
- Fastp
- Trimmomatic

That’s all for today.. questions??

Thank you

Contact us

Jiajia Li
Biological Data Science Institute

RN Robertson Building, 46 Sullivan’s Creek Rd
The Australian National University
Canberra ACT 2600

E jiajia.li1@anu.edu.au
W https://bdsi.anu.edu.au/

Australian
National
University