Stage 03

أكاديمية كاوست
KAUST ACADEMY

Transcriptomics Analysis Pipeline

Day 02 – Session 01

The KAUST Academy & The Bioinformatics Platform

4-7 Feb 2026

Agenda – Day 02

أكاديمية كاوست
KAUST ACADEMY

Time	Session	Topics & Activities
09:00-09:15	S1: Recap of Day 1	Review of previous day's content
09:15-10:00	S2: Quality Control	Sequencing quality metrics, Phred scores, technical variation, trimming, preprocessing
10:00-11:00	L3: Quality Control Hands-on	FastQC, fastp; inspect data quality metrics, trimming, adapter removal, pre/post-QC comparison View Lab Instructions →
11:00-12:00	S4: Reference Genome Alignment	Reference genomes, annotations, splice-aware alignment
14:00-15:00	L4: Genome Alignment Hands-on	Align reads, inspect BAM files, evaluate mapping metrics (samtools, STAR) View Lab Instructions →
15:00-15:45	S5: Read Counting & Quantification	Gene vs transcript quantification, expression matrices
15:45-16:30	L5: Quantification & Expression Matrix	Salmon, expression matrix inspection View Lab Instructions →
16:30-17:00	Q&A & Project Review	Questions and project progress support

+ Experimental design

Send your questions here

Recap

Answer:

- Compute environments
  - Hardware: PCs, Workstations, Clusters; onsite vs cloud
  - Software: OS choice and command-line

- An overview of transcriptomics

From the lab we learned:

- Created an organized project directory structure
- Learned how to navigate NCBI GEO and SRA to find RNA-seq datasets
- Learned how to browse Ensembl to find reference genome and annotation files
- Downloaded reference genome and annotation files for chromosome 11
- Explored FASTA, GTF and FASTQ file formats

Agenda – Day 02

أكاديمية كاوست
KAUST ACADEMY

Time	Session	Topics & Activities
09:00-09:15	S1: Recap of Day 1	Review of previous day's content
09:15-10:00	S2: Quality Control	Sequencing quality metrics, Phred scores, technical variation, trimming, preprocessing
10:00-11:00	L3: Quality Control Hands-on	FastQC, fastp; inspect data quality metrics, trimming, adapter removal, pre/post-QC comparison View Lab Instructions →
11:00-12:00	S4: Reference Genome Alignment	Reference genomes, annotations, splice-aware alignment
14:00-15:00	L4: Genome Alignment Hands-on	Align reads, inspect BAM files, evaluate mapping metrics (samtools, STAR) View Lab Instructions →
15:00-15:45	S5: Read Counting & Quantification	Gene vs transcript quantification, expression matrices
15:45-16:30	L5: Quantification & Expression Matrix	Salmon, expression matrix inspection View Lab Instructions →
16:30-17:00	Q&A & Project Review	Questions and project progress support

+ Experimental design

أكاديمية كاوست
KAUST ACADEMY

Sequencing Quality Evaluation

Metrics, and Best Practices for RNA-seq Data

Day 02 – Session 02

Why Quality Control Matters

Poor Quality Bases

Low-quality base calls at read ends can lead to incorrect alignments and false variant calls.

Adapter Contamination

Residual adapter sequences cause mis-alignments or complete removal of reads during mapping.

Too-Short Reads

Very short reads after trimming cannot be reliably aligned and may crash alignment software.

Method: *Trim Galore* a wrapper for *FastQC* and *Cutadapt* or *fastp* that has many features

Dataset inspection

Before getting into the data ensure:

- Review the metadata or sample sheet
- You have the right dataset (files from the correct project and study)
- Match number of samples and replicates with number of FastQ files
- Check the size of all files and look for any big difference (**du -sh**)
- Use **md5** checksum if it is copied from another source
- Print the last few lines of each file to make sure they end correctly
- Count number of lines and they should be divisible by 4, why ? (**wc -l**)
- Make sure the files are named properly (don't start with numbers, no spaces or special characters in the file name, etc.
- If data is paired-end, the FastQ files should usually end with as *_1.fastq* and *_2.fastq*

Start with raw reads (FastQ Format)

@K00235:171:H2VKHBBXY:3:1101:19380:2475 1:N:0:CGATGT
CAGAGCATTCNGCTACTCCATTACAACGCGTGTGGAATGGGAGACACTTTTTATCGGCACTACATT
+
AAAAAEEEEE#EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAEEEEEEEEEEEEEE

the header line represents

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

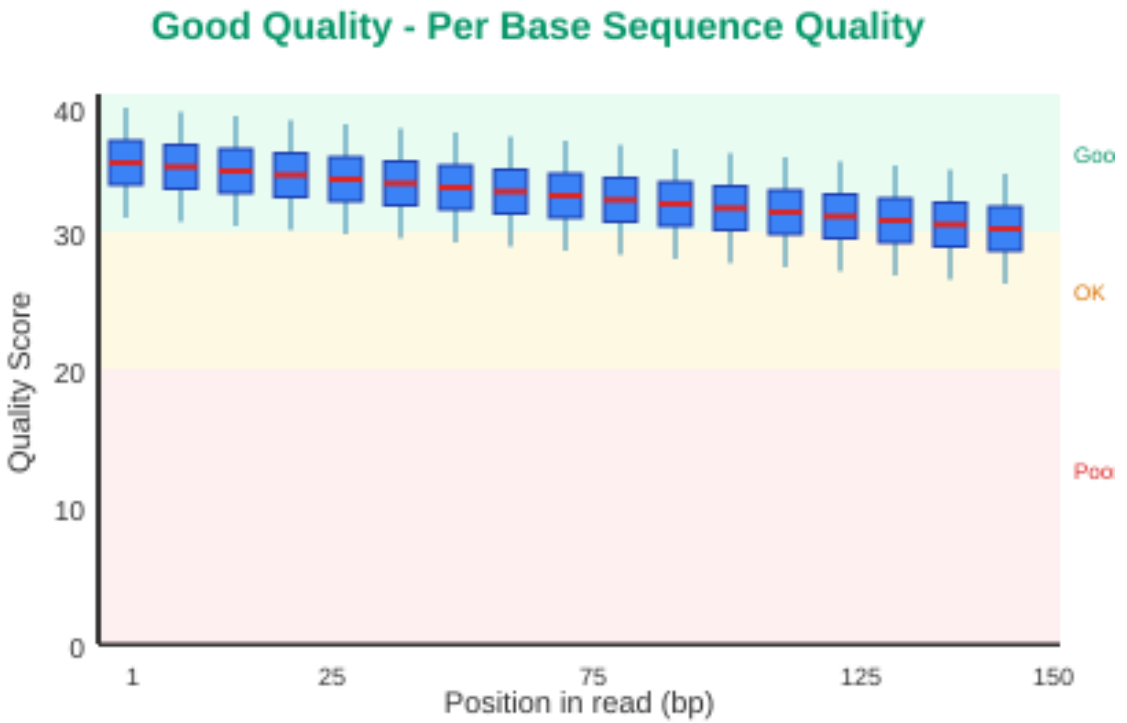
- Starts with @ (required by fastq spec)
- Instrument ID (HWUSI-EAS611)
- Run number (34)
- Flowcell ID (6669YAAXX)
- Lane (5)
- Tile (1)
- X-position (5069)
- Y-position (1159)
- [space]
- Read number (1)
- Was filtered (Y/N) (N) - You wouldn't normally see the Ys
- Control number (0 = no control)
-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

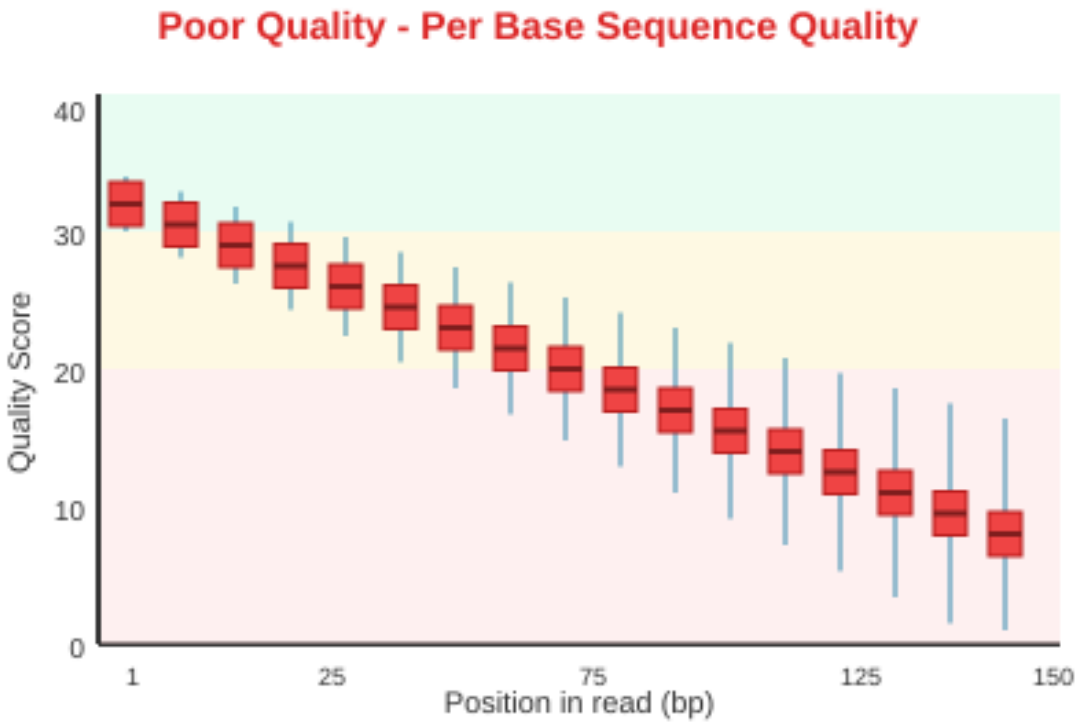
Any FastQ file from illumina usually contains millions of such reads

Therefore, we use speciallized tools to handle FastQ files

1) Per-Base Sequence Quality

Phred quality scores at each position along the read





Interpretation:

• Green zone (Q30+): Excellent quality, 99.9% accuracy • Yellow zone (Q20-30): Acceptable • Red zone (<Q20): Poor quality
• Quality drop at 3' end is normal — trim if severe • Box plots show median (red line), IQR (box), and range (whiskers)

2) Understanding Phred Quality Scores

Phred quality scores (Q) indicate the probability of an incorrect base call:

$Q = -10 \times \log_{10}(P)$ where P = probability of error

Phred Score	Error Probability	Base Call Accuracy	Interpretation
Q10	1 in 10 (10%)	**90%**	Poor - not usable
Q20	1 in 100 (1%)	**99%**	Acceptable minimum
Q30	1 in 1,000 (0.1%)	**99.9%**	Good - standard target
Q40	1 in 10,000 (0.01%)	**99.99%**	Excellent

Key takeaway: Aim for >80% of bases at Q30 or above.
Modern Illumina sequencers typically achieve >85% Q30

```
@K00235:171:H2VKHBBXY:3:1101:19380:2475 1:N:0:CGATGT
CAGAGCATTCNGCTACTCCATTACAACGCGTGTGGAATGGG
+
AAAAAEEEEE#EEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```

Sanger, Illumina v1.3 to 1.7 (ASCII_BASE=64)

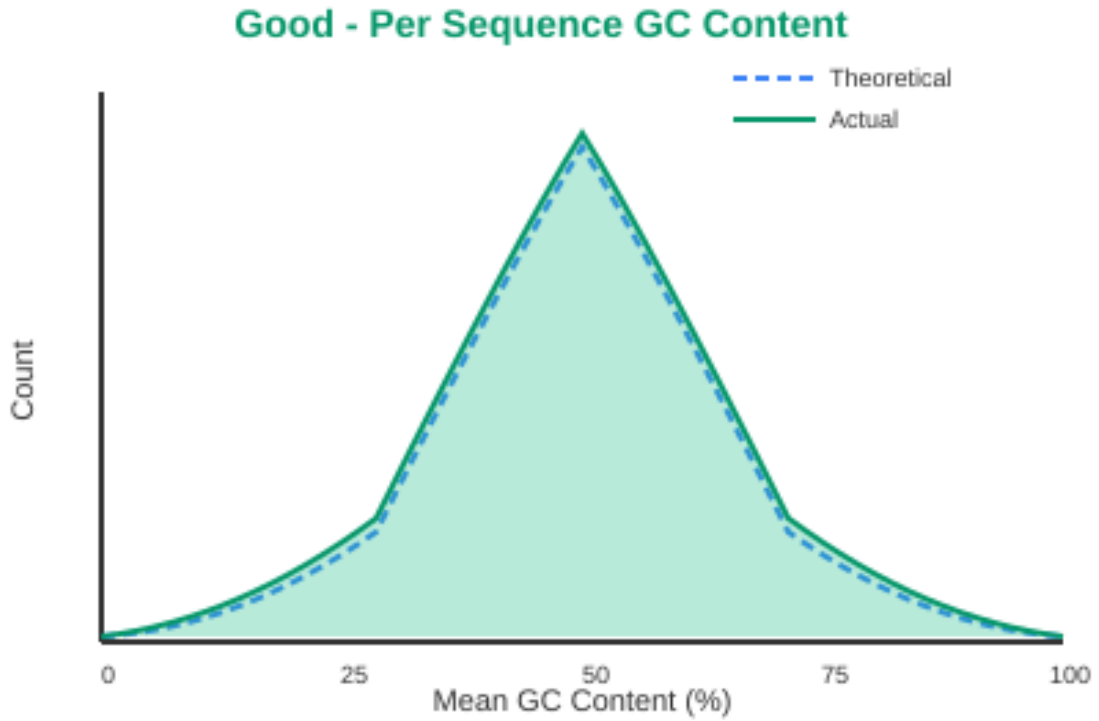
Q	ASCII	P	Q	ASCII	P	Q	ASCII	P	Q	ASCII	P
1	A	0.79433	12	L	0.06310	23	W	0.00501	34	b	0.00040
2	B	0.63096	13	M	0.05012	24	X	0.00398	35	c	0.00032
3	C	0.50119	14	N	0.03981	25	Y	0.00316	36	d	0.00025
4	D	0.39811	15	O	0.03162	26	Z	0.00251	37	e	0.00020
5	E	0.31623	16	P	0.02512	27	[	0.00200	38	f	0.00016
6	F	0.25119	17	Q	0.01995	28	\	0.00158	39	g	0.00013
7	G	0.19953	18	R	0.01585	29	]	0.00126	40	h	0.00010
8	H	0.15849	19	S	0.01259	30	^	0.00100			
9	I	0.12589	20	T	0.01000	31	_	0.00079			
10	J	0.10000	21	U	0.00794	32	`	0.00063			
11	K	0.07943	22	V	0.00631	33	a	0.00050			

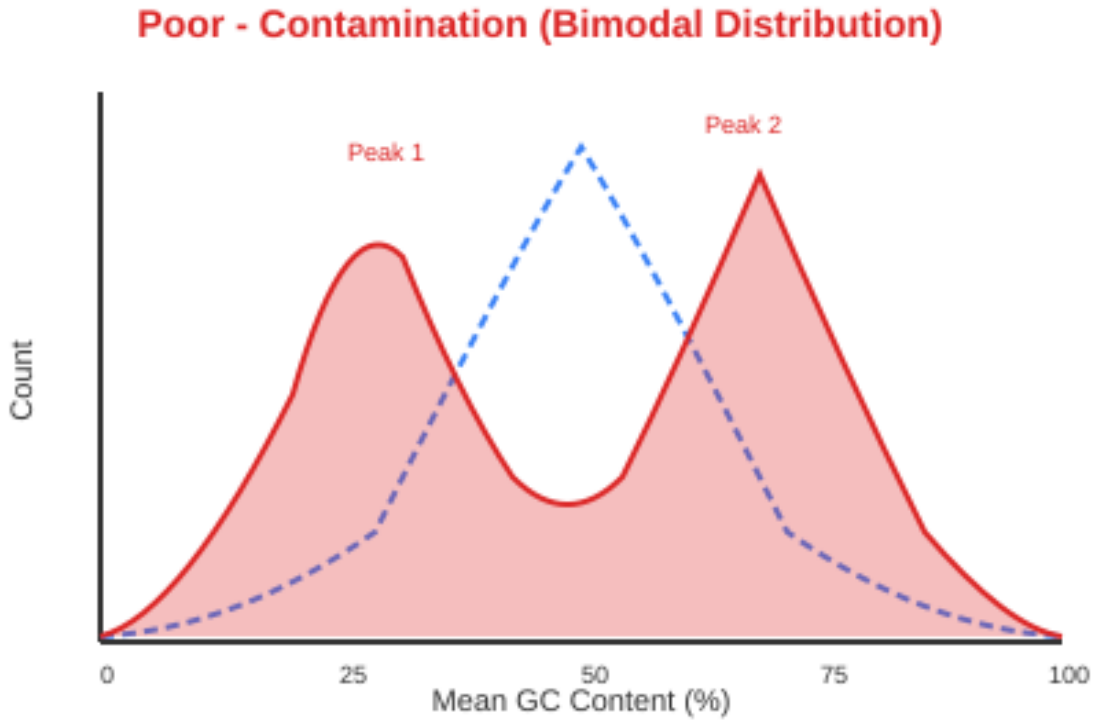
Illumina v1.8 and later (ASCII_BASE=33)

Q	ASCII	P	Q	ASCII	P	Q	ASCII	P	Q	ASCII	P
1	"	0.79433	12	-	0.06310	23	8	0.00501	34	C	0.00040
2	#	0.63096	13	.	0.05012	24	9	0.00398	35	D	0.00032
3	$	0.50119	14	/	0.03981	25	:	0.00316	36	E	0.00025
4	%	0.39811	15	0	0.03162	26	;	0.00251	37	F	0.00020
5	&	0.31623	16	1	0.02512	27	<	0.00200	38	G	0.00016
6	'	0.25119	17	2	0.01995	28	=	0.00158	39	H	0.00013
7	(	0.19953	18	3	0.01585	29	>	0.00126	40	I	0.00010
8	)	0.15849	19	4	0.01259	30	?	0.00100	41	J	0.00008
9	*	0.12589	20	5	0.01000	31	@	0.00079			
10	+	0.10000	21	6	0.00794	32	A	0.00063			
11	,	0.07943	22	7	0.00631	33	B	0.00050			

3) GC Content Distribution

Distribution of GC content across all reads — should match expected genome GC%



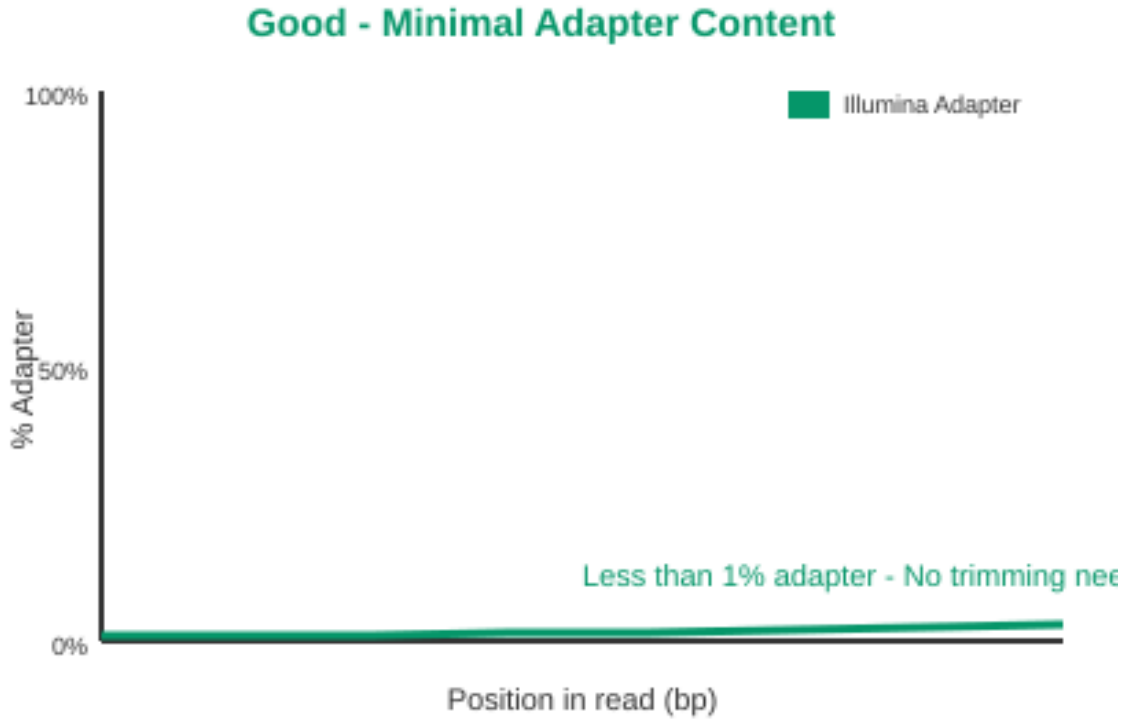


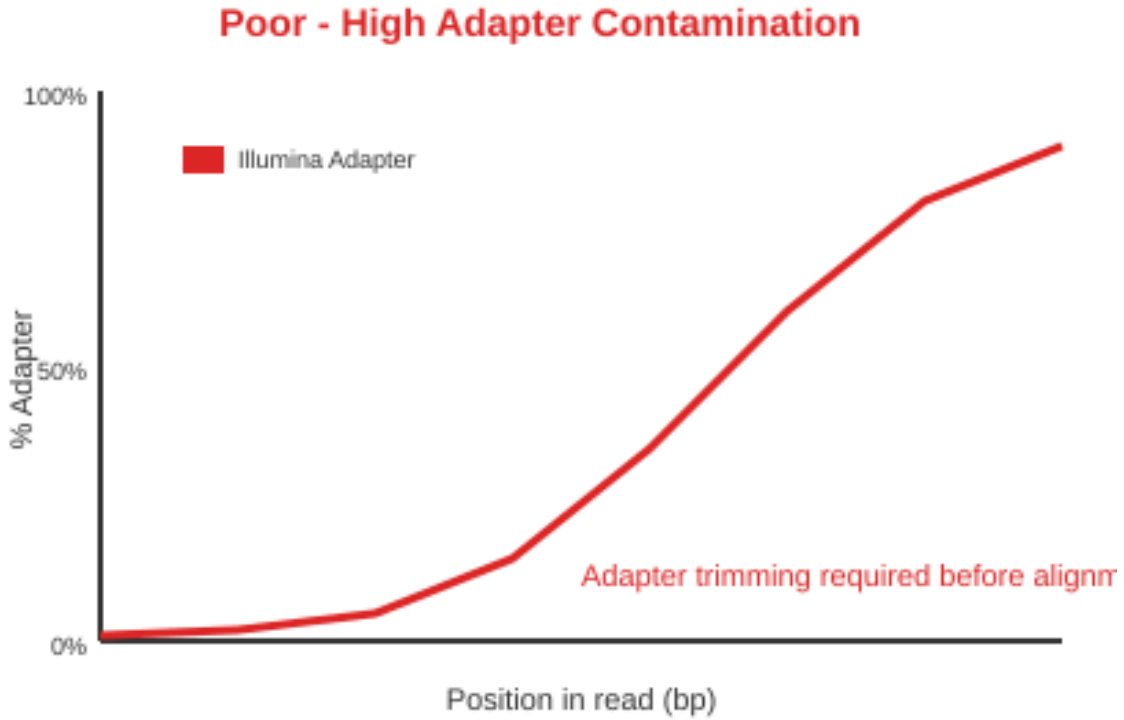
Interpretation:

- Good: Single peak matching theoretical distribution (blue dashed line) — indicates clean, uncontaminated sample
- Bad: Multiple peaks suggest contamination (bacteria, adapter dimers) or significant library bias — investigate source

4) Adapter Conten

Cumulative percentage of reads with adapter sequence at each position



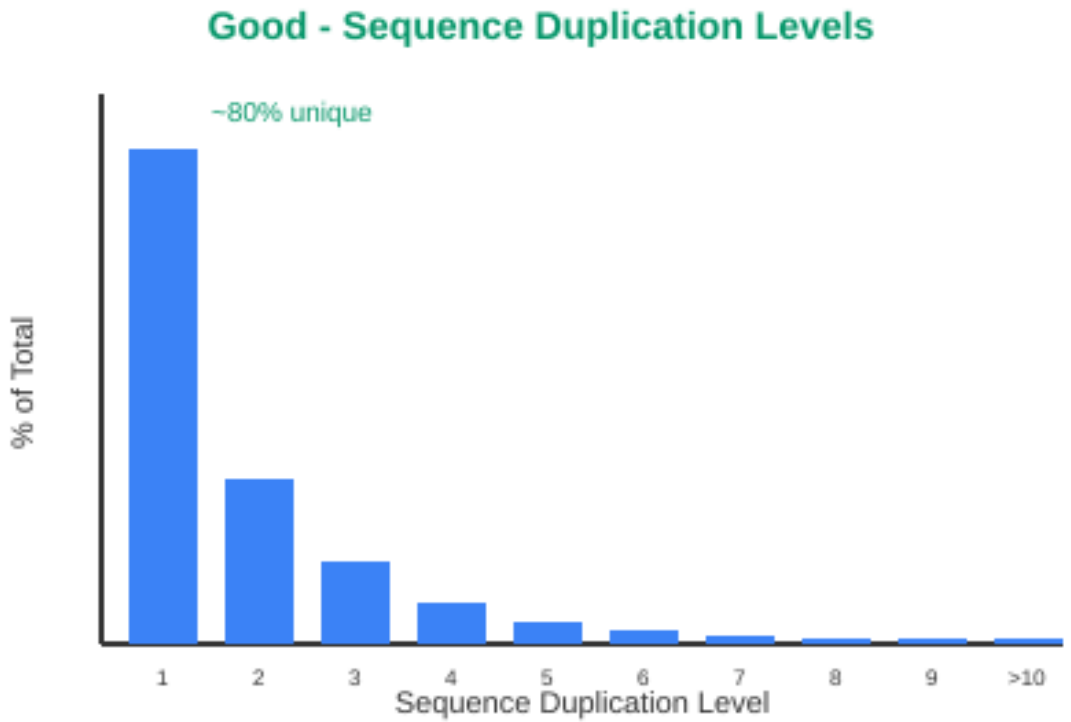


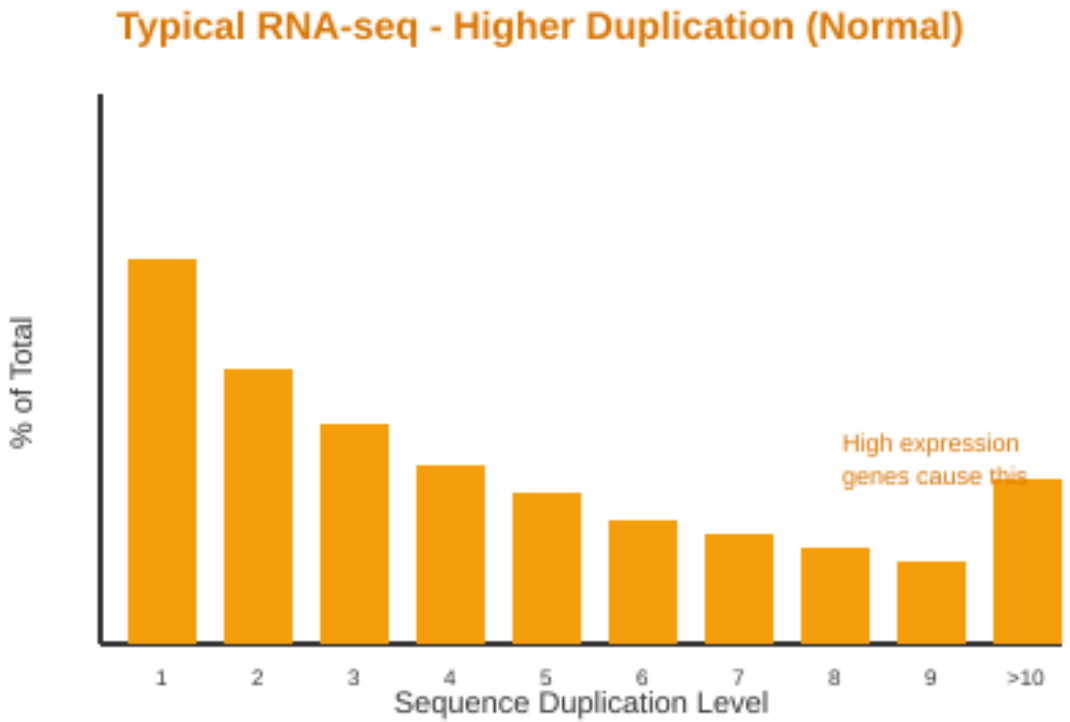
Interpretation:

- Good: Flat line near 0% — adapters were properly removed or insert size is appropriate
- Bad: Rising curve at 3' end — short inserts or read-through into adapter; use Cutadapt or Trimmomatic to remove

5) Sequence Duplication Levels

Percentage of sequences appearing at different duplication levels





Interpretation:

- DNA-seq: Most sequences should be unique (left plot) — high duplication indicates low library complexity or PCR artifacts
- RNA-seq: Higher duplication is NORMAL (right plot) — highly expressed genes naturally produce many identical reads

RNA-seq Quality Control

Accounting for sequencing and library issues in RNA-seq Data

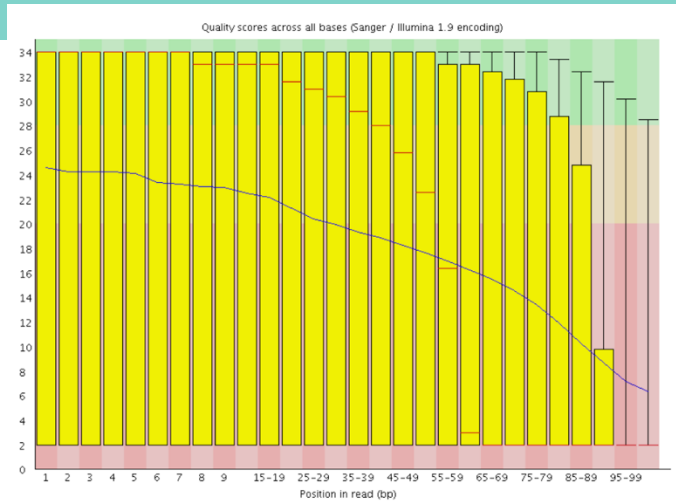
Day 02 – Session 02

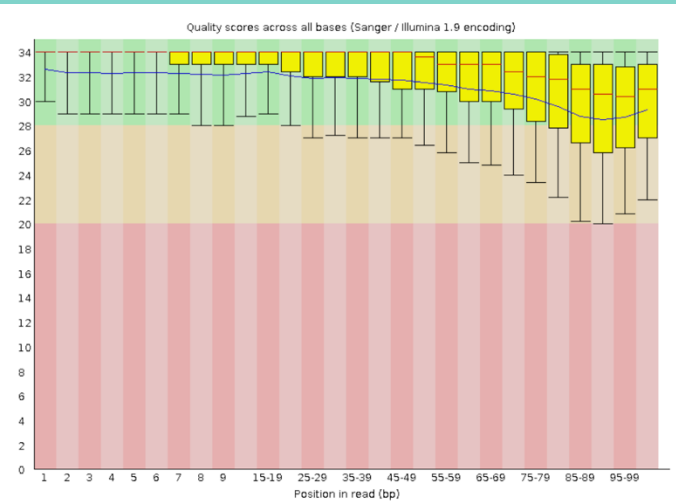
cont.

Quality Trimming Results

Trimmed with a Phred score threshold of 20

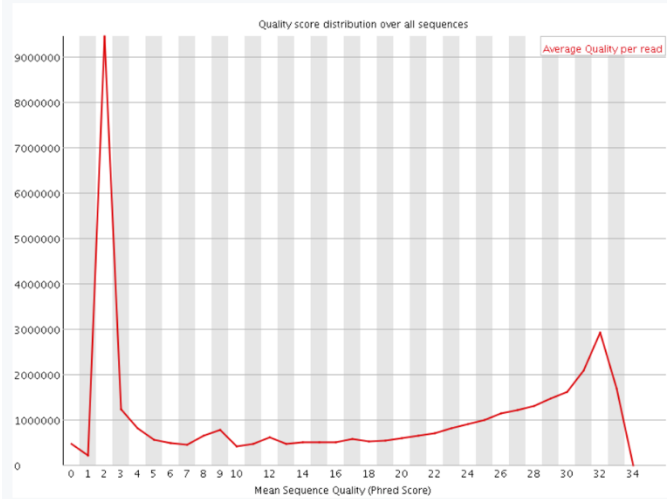
What differences do you observe?

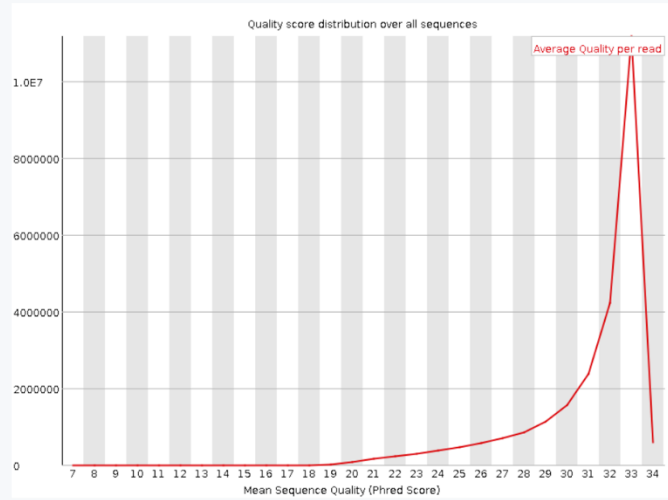




Before Quality Trimming

After Quality Trimming





Adapter Trimming

Detecting and removing sequencing adapter sequences

What It Does

- Removes adapter sequences from 3' end of reads
- Auto-detects adapter type (Illumina, Nextera, sRNA) (fastp uses PE read overlap analysis)
- Uses stringent overlap detection (even 1 bp match)
- Searches first 1m reads to find adapters

Why It's Critical

- Adapter sequences are not part of the biological sample
- Contamination causes incorrect alignments
- In bisulfite-seq: leads to wrong methylation calls

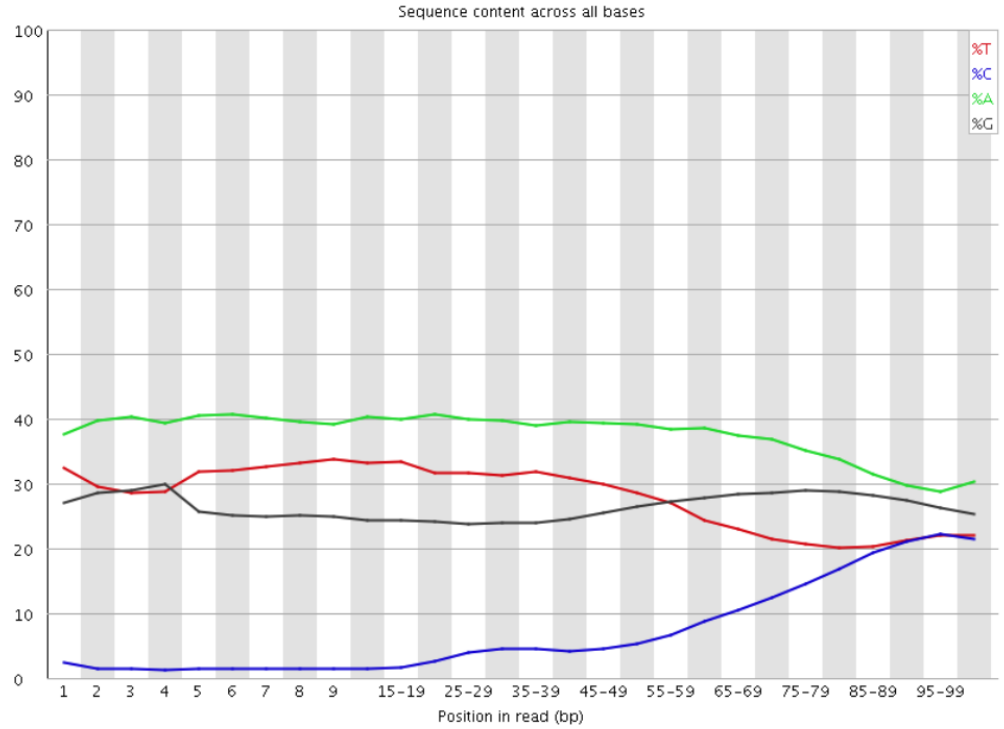
Example:

- Illumina paired-end adapters ('AGATCGGAAGAGC') are automatically recognized and removed. For reads shorter than the insert size, adapter read-through is detected and trimmed.
- For paired-end data, fastp analyzes where read pairs overlap. If Read 1 ends with adapter sequence and Read 2 begins with it (palindrome pattern), this confirms adapter contamination, and both are trimmed.

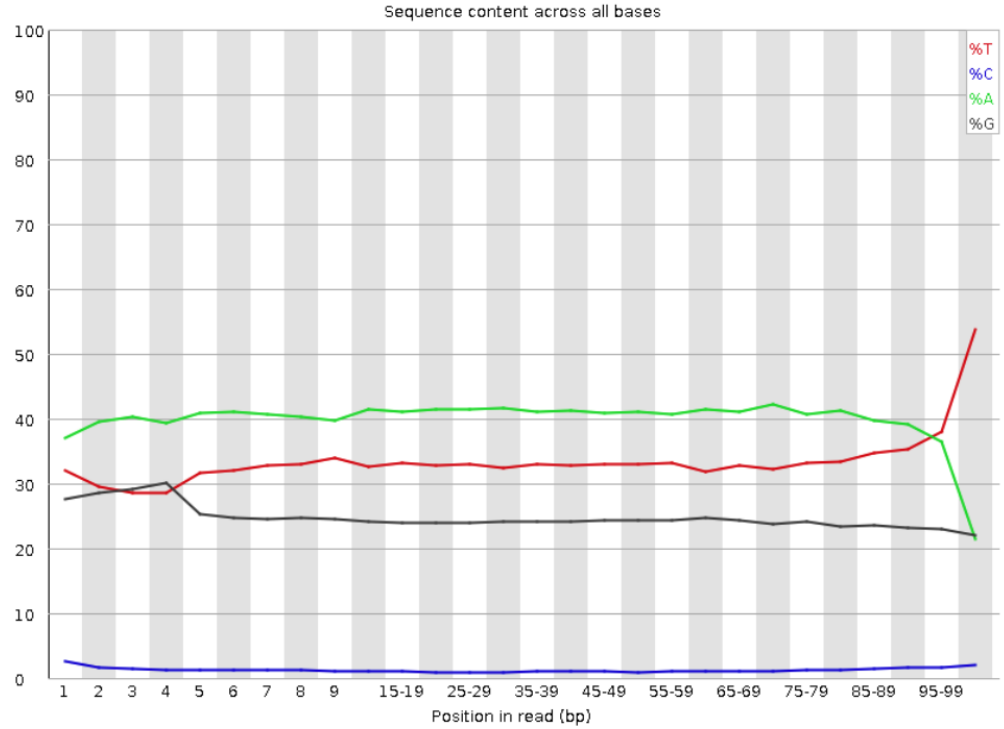
Adapter Trimming

Adapter trimming with Cutadapt gets rid of most signs of adapter contamination

Before Adapter Trimming



After Adapter Trimming



What differences do you observe?

Read Length Filtering

Removing reads that are too short after trimming

What It Does

- Filters reads below a minimum length (default: 20 bp)
- Reduces output file size by removing unusable reads
- Prevents alignment software crashes

Why It's Needed

- Very short reads cannot align uniquely to genomes
- Many aligners require minimum sequence length
- Short reads increase multi-mapping ambiguity

Example:

A 100 bp read with 85 bp of adapter contamination would be trimmed to 15 bp. With the default 20 bp threshold, this read is removed entirely as it cannot contribute meaningful alignment information.

Overrepresented Sequences

Detecting sequences that appear too frequently (tool: fastp)

What It Does

- Samples reads evenly across entire file
- Identifies sequences at 10, 20, 40, 100 bp lengths
- Reports position distribution (head vs tail)

Why It's Essential

- Reveals adapter contamination patterns
- Detects polyG/polyX artifacts
- Identifies PCR over-duplication signatures

Example:
If 'AGATCGGAAGAGC' appears in 5% of reads concentrated at read tails, this indicates adapter contamination. fastp reports both frequency and position, helping diagnose the specific issue.

Duplication Analysis

Evaluating and optionally removing PCR duplicates (tool: fastp)

What It Does

- Calculates duplication rate across the dataset
- Optional deduplication to remove identical reads

Source:

- Very deep sequencing?
- Specific genes?
- PCR issues?
- Ribosomal RNAs sequenced?

Example:
fastp reports duplication levels with GC content correlation. High-duplication reads often have biased GC, indicating PCR artifacts. This helps diagnose library prep problems early.

Subsampling / Read Limiting

Processing only a subset of reads

What It Does

- --reads_to_process: Limit total reads processed
- Default 0 means process all reads
- Creates filtered subset of original data

Use Cases

- Quick quality preview of large datasets
- Create test subsets for pipeline development
- Downsample for resource-limited analysis

Example:
Using --reads_to_process 1000000 processes only the first 1M reads. Ideal for quickly checking data quality before committing to full processing.

Decontaminating reads

Why is Decontamination Critical?

Contaminated reads lead to unreliable results and wasted resources

Common Contamination Sources

- **Ribosomal RNA (rRNA)**

Can comprise 40-80% of total RNA-seq reads

- **Host genomic DNA**

Human/mouse contamination in samples

- **Microbial contamination**

Bacteria, fungi, or viral sequences

- **Technical artifacts**

Adapters, primers, PhiX spike-in

Impact on Analysis

- **Reduced mapping rates**

Fewer reads align to target genome

- **Skewed gene expression**

False differential expression results

- **Wasted sequencing depth**

Paying to sequence contaminants

- **Assembly artifacts**

Chimeric contigs, misassemblies

Up to 80% of RNA-seq reads can be rRNA without depletion

Tools for Decontamination

Tool	Target	Best For
Kraken2 + KrakenTools	Bacteria, viral, human, fungi	General-purpose decontamination
BBSplit (BBTools)	Host genomes, multiple refs	Host removal, PDX experiments
SortMeRNA	rRNA only	RNA-seq rRNA depletion
Bowtie2 + SAMtools	Any reference genome	Precise removal, custom refs

Tip: Combine tools for comprehensive decontamination

Use SortMeRNA for rRNA, then Kraken2/BBSplit for remaining contaminants

Best Practices & Quality Control

Best Practices

- **Run FastQC before AND after**

Compare read counts, quality distributions

- **Use appropriate databases**

Match to your expected contaminants

- **Keep logs and statistics**

Track % reads removed at each step

- **Validate with known samples**

Test pipeline on positive controls

- **Document tool versions**

Ensure reproducibility

Warning Signs

- **>30% reads removed**

May indicate sample or library issue

- **Very low mapping after cleaning**

Check if correct reference was used

- **Unexpected species in Kraken**

Cross-contamination or mislabeling

- **Inconsistent results across replicates**

Batch effects or technical issues

- **Quality drop after filtering**

Review filtering parameters

Hands-on: Quality Control

Lab 3: Quality Control Hands-on

Learning Objectives

- Run FastQC to assess raw read quality
- Interpret FastQC reports and identify quality issues
- Use fastp for quality trimming and adapter removal
- Compare pre- and post-trimming quality with MultiQC

Step 1: Initial QC with FastQC

```
fastqc *.fastq.gz -o fastqc_raw/ -t 8
```

Step 2: Trim adapters and low-quality bases

```
fastp -i R1.fq.gz -I R2.fq.gz -o R1_trim.fq.gz -O
R2_trim.fq.gz  --detect_adapter_for_pe -q 20 -l 36 --
html fastp.html
```

Step 3: Post-trim QC

```
fastqc *_trim.fq.gz -o fastqc_trimmed/ -t 8
```

Step 6: Aggregate all QC reports

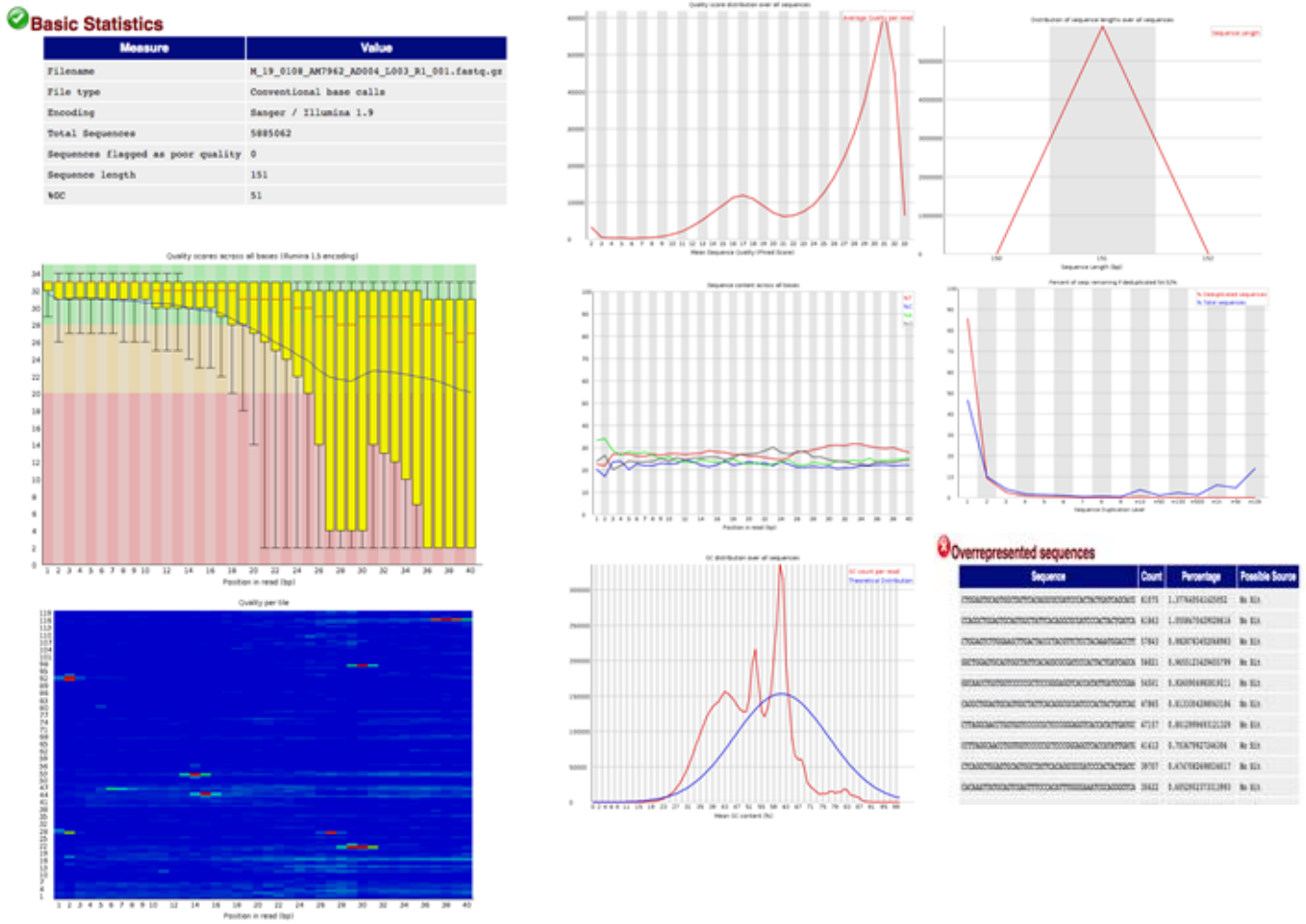
```
multiqc . -o multiqc_report/
```

FastQC tool for genomics sequencing data evaluation

Open the HTML file generated by fastQC to find the QC plots shown on the next slides

Fastp also generates an HTML file that contains many useful QC plots, inspect them

Reports generated by FastQC



FastQC modules - Basic Statistics

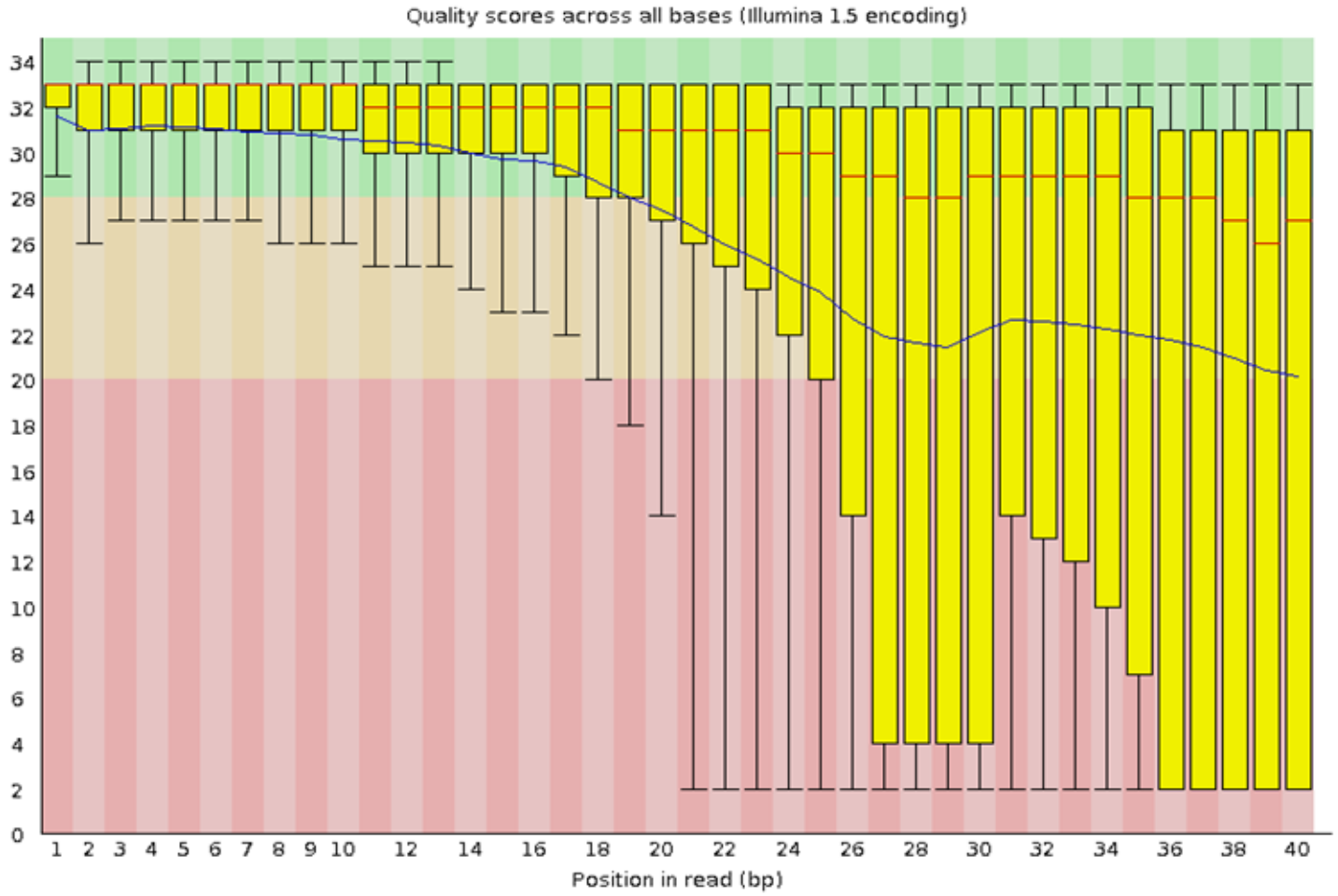
Basic Statistics

Measure	Value
Filename	M_19_0108_AM7962_AD004_L003_R1_001.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	5885062
Sequences flagged as poor quality	0
Sequence length	151
%GC	51

FastQC modules - Per base sequence quality

أكاديمية كاوست
KAUST ACADEMY

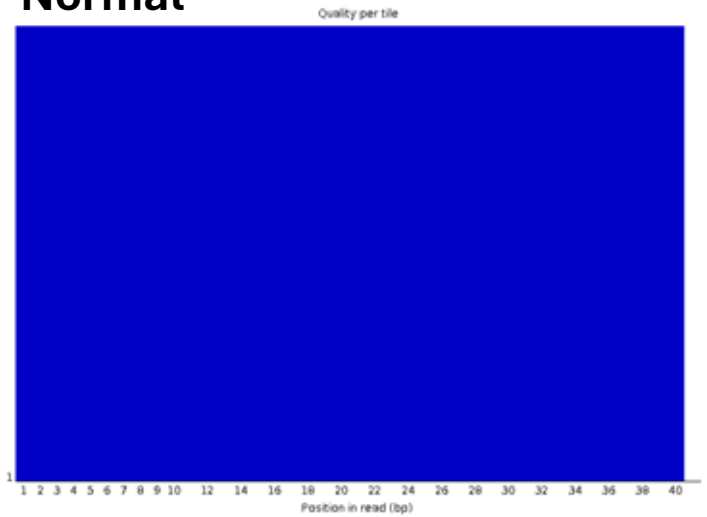
Per base sequence quality



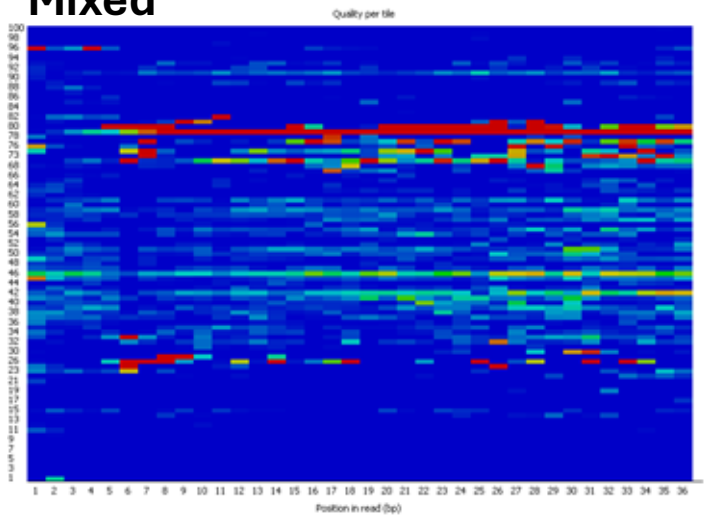
FastQC modules - Per tile sequence quality

Per tile sequence quality

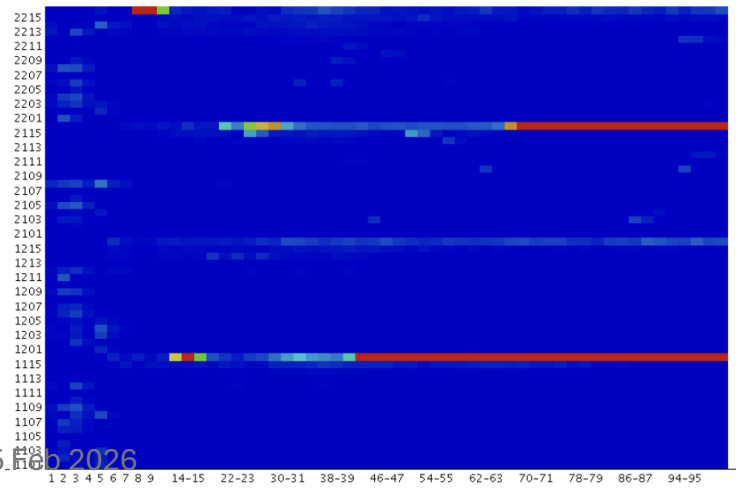
Normal



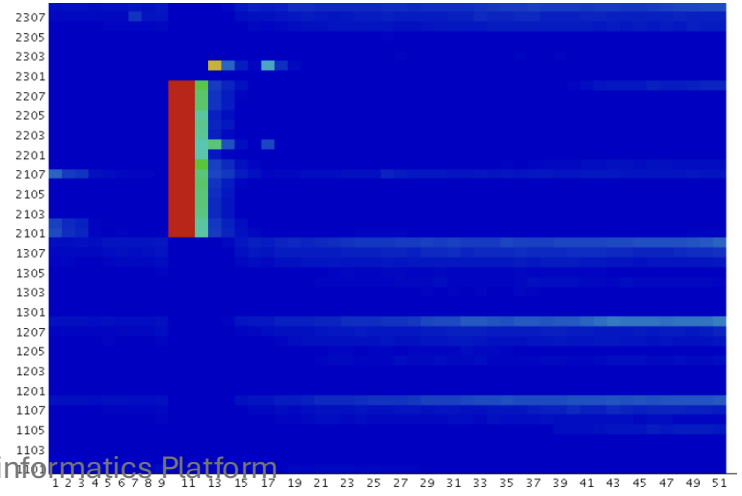
Mixed



Obstruction



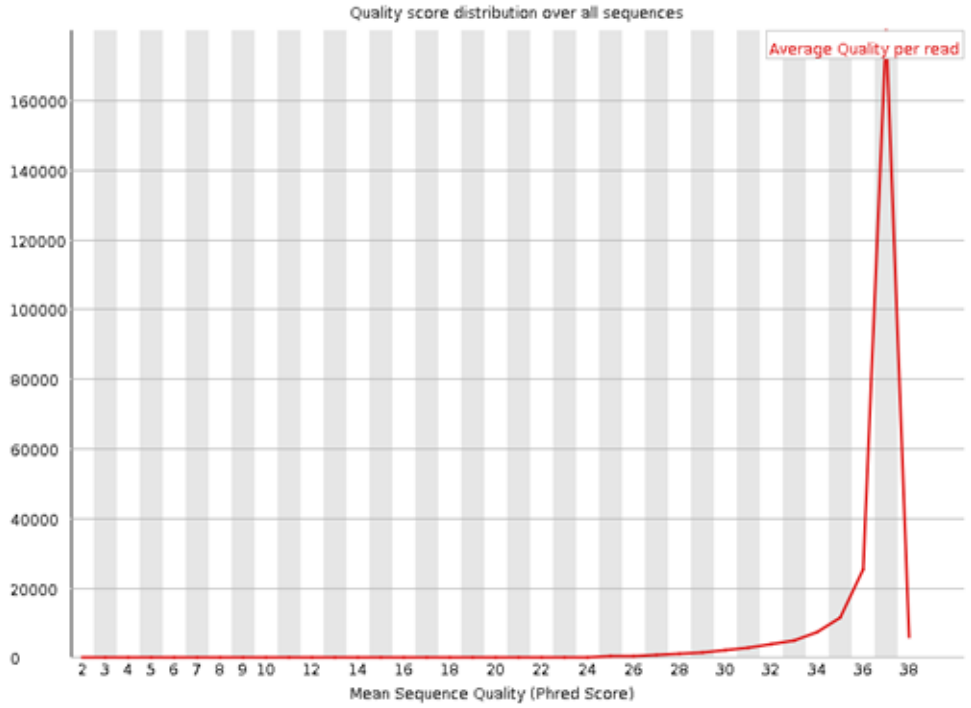
Bubble



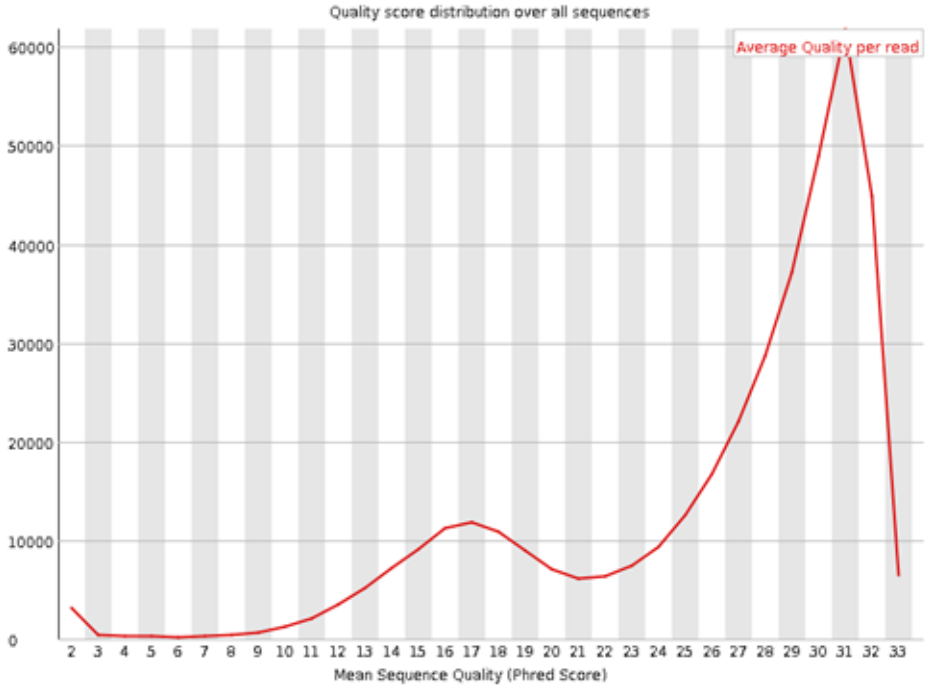
FastQC modules - Per sequence quality scores

Per sequence quality scores

High mean quality score



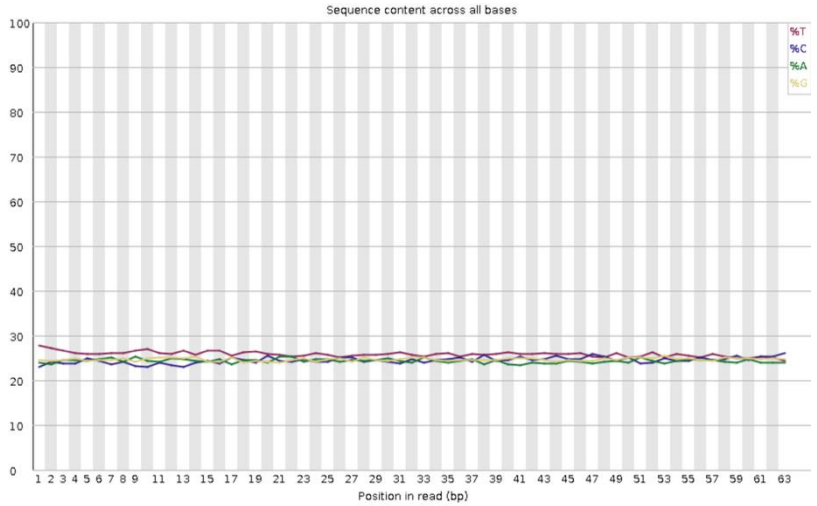
Subset of reads with lower Q scores



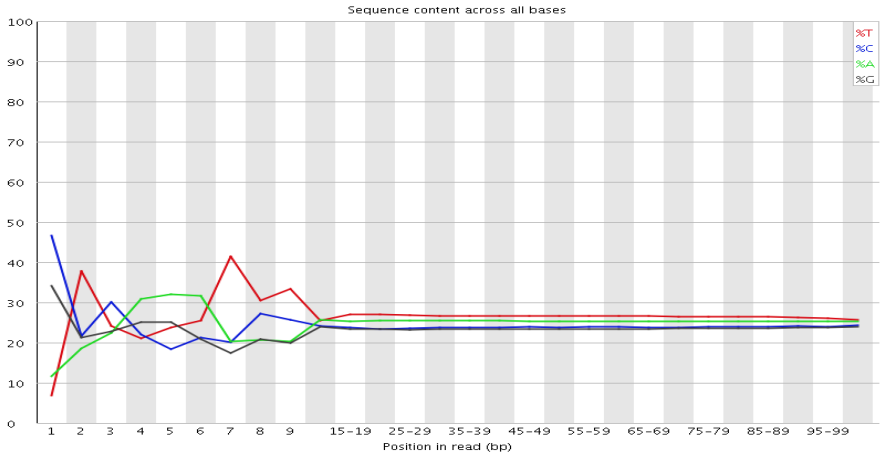
FastQC modules - Per base sequence content

Per base sequence content

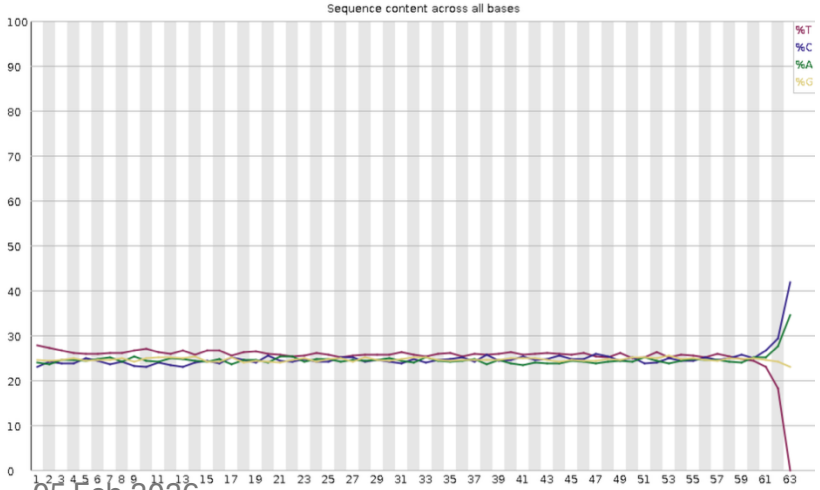
KO raw



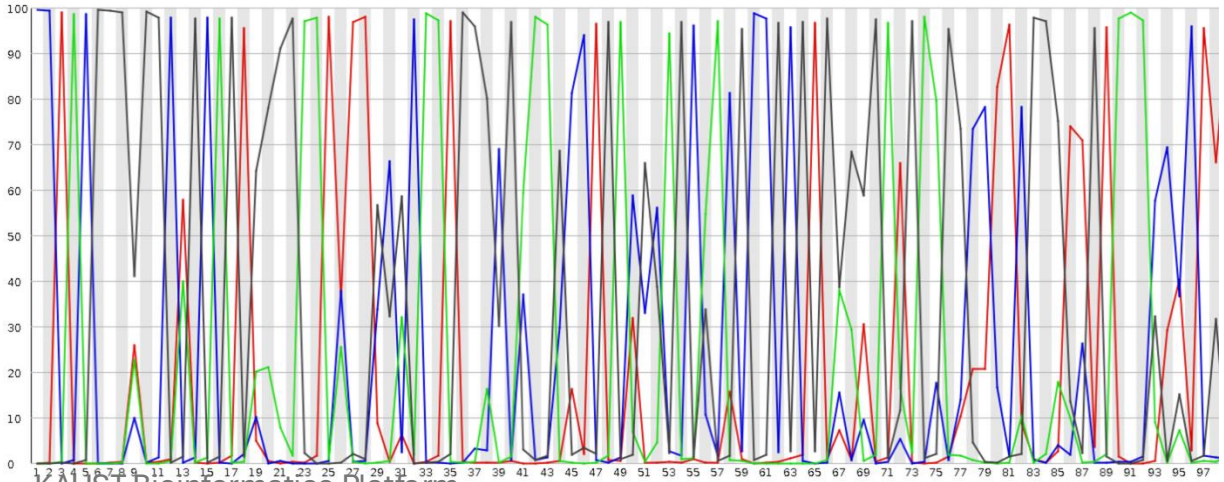
Typical RNA-seq profile



KO trimmed



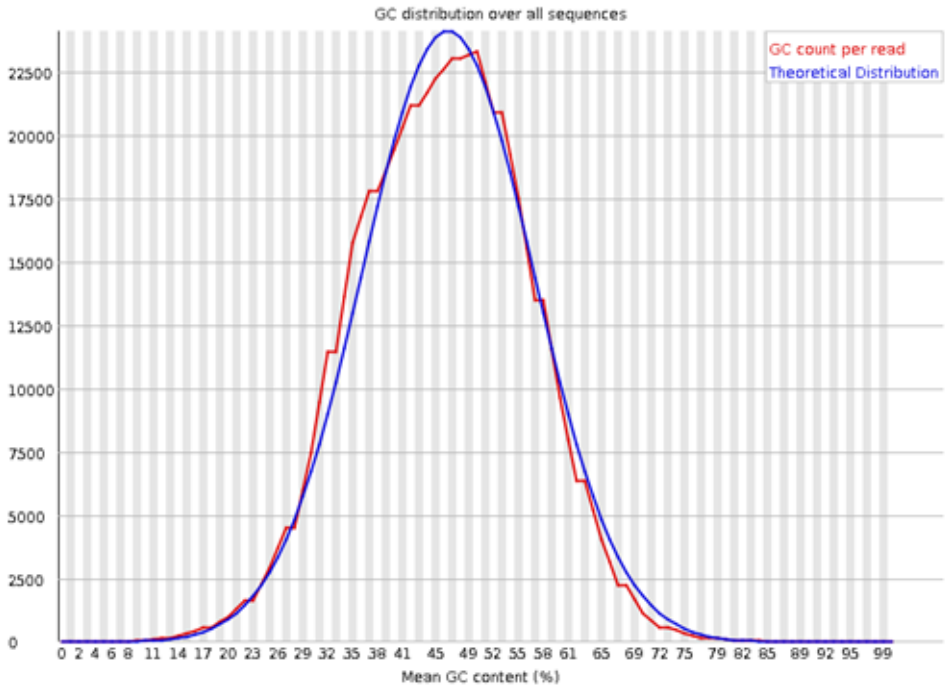
Low diversity



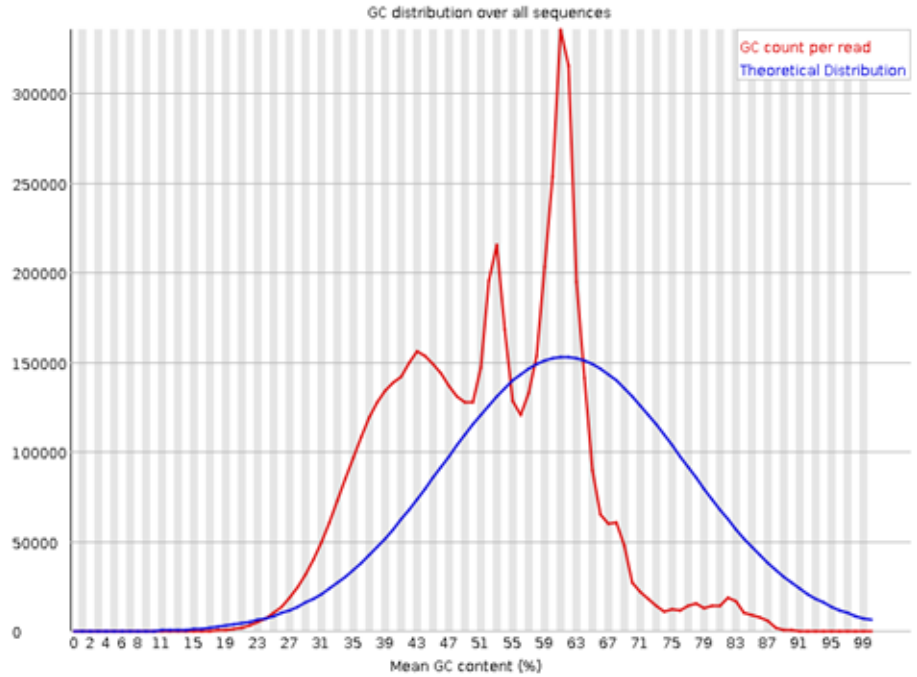
FastQC modules - Per sequence GC content

Per sequence GC content

Unimodal mean GC% distribution



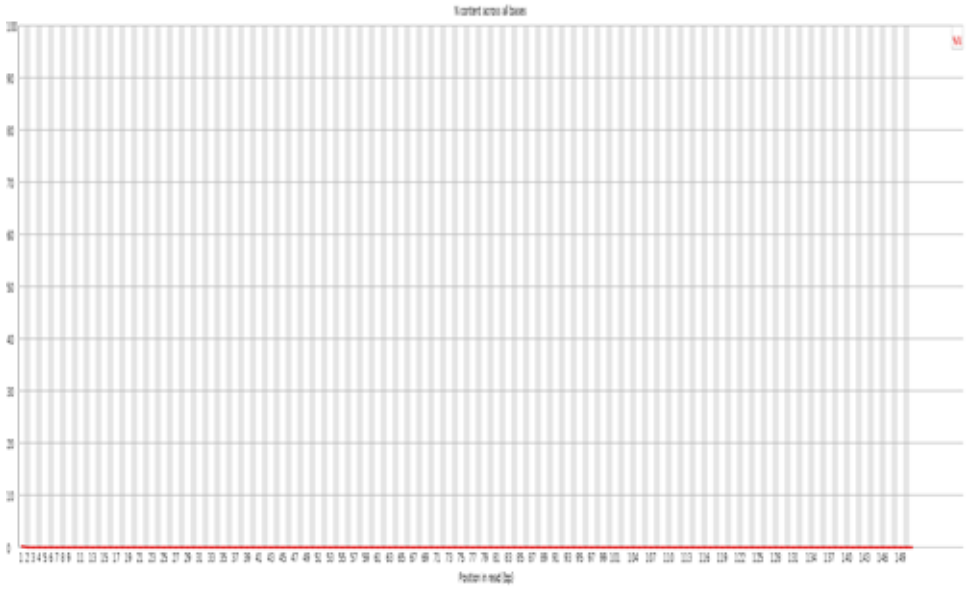
GC% distribution with sharp peaks



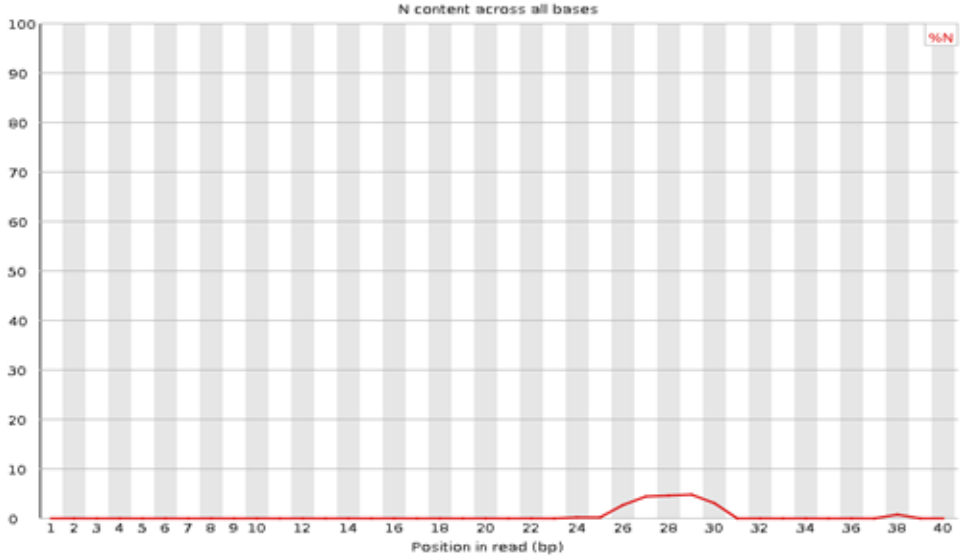
FastQC modules - Per base N content

Per base N content

Absence of N content

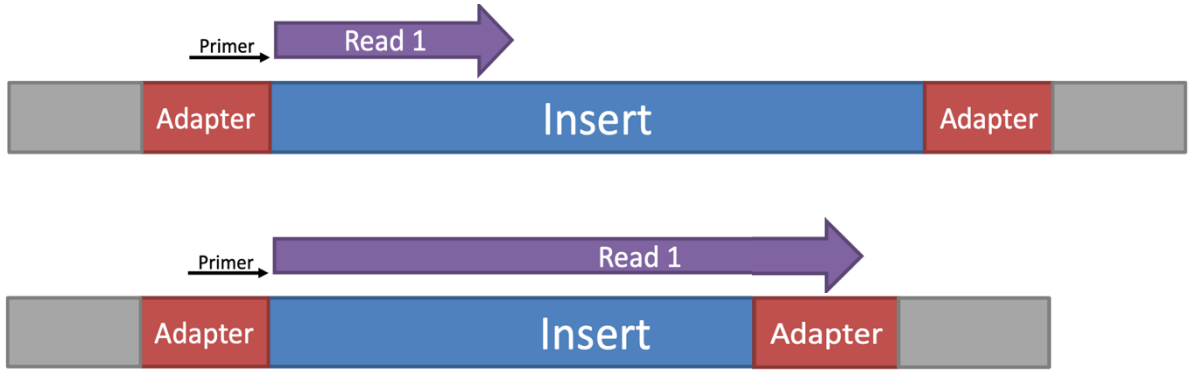


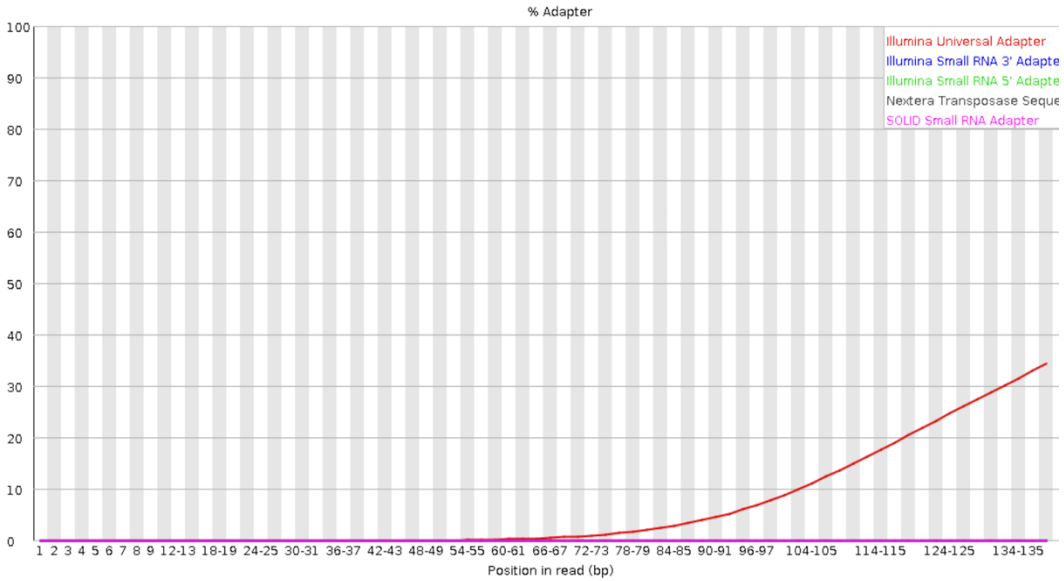
Few reads have N content



FastQC modules - Adapter content

Adapter Content



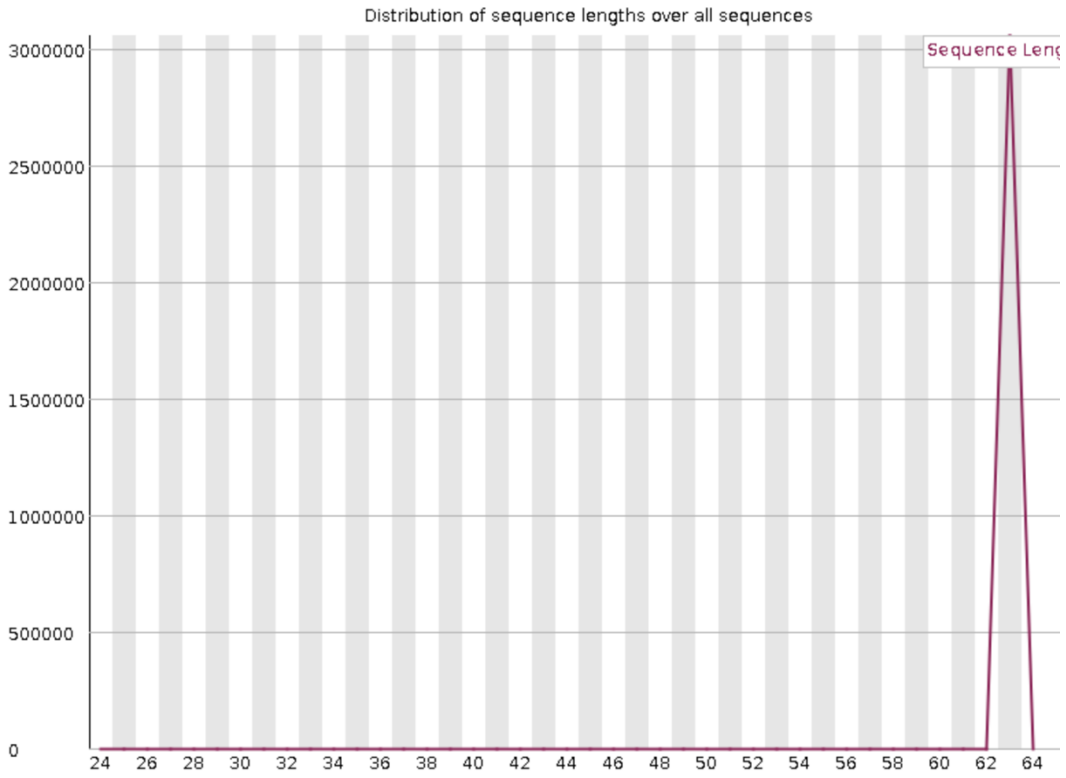


FastQC modules - Sequence length distribution

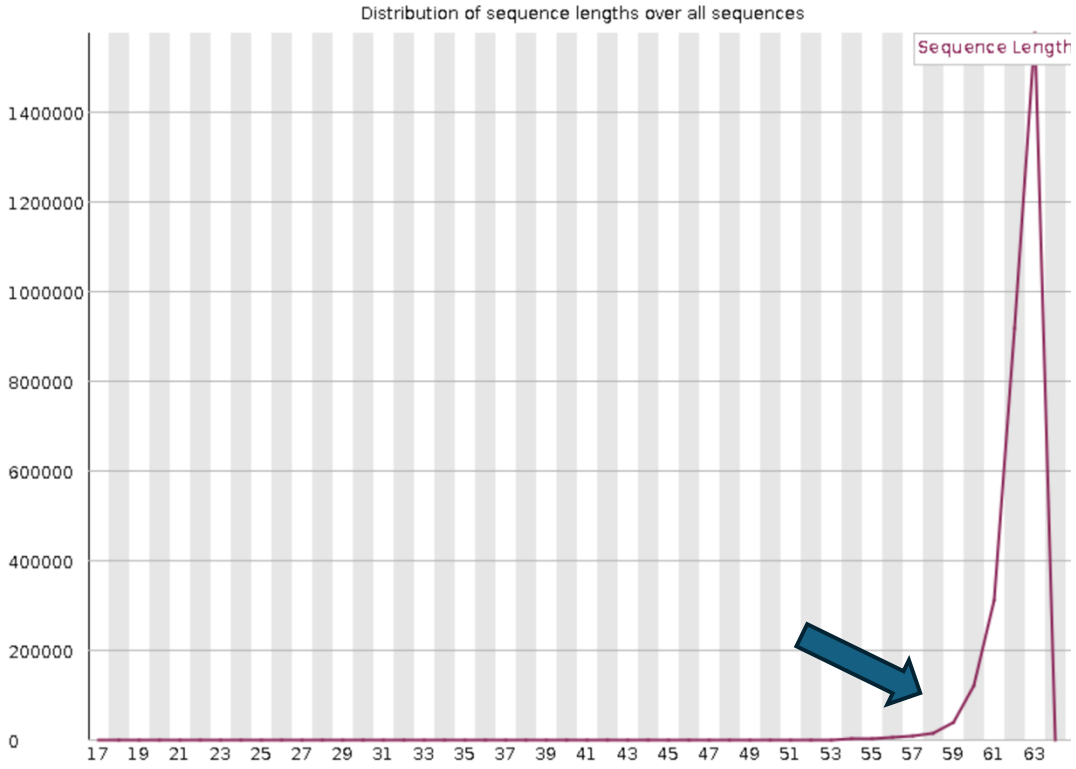
أكاديمية كاوست
KAUST ACADEMY

Sequence Length Distribution

Before Trimming



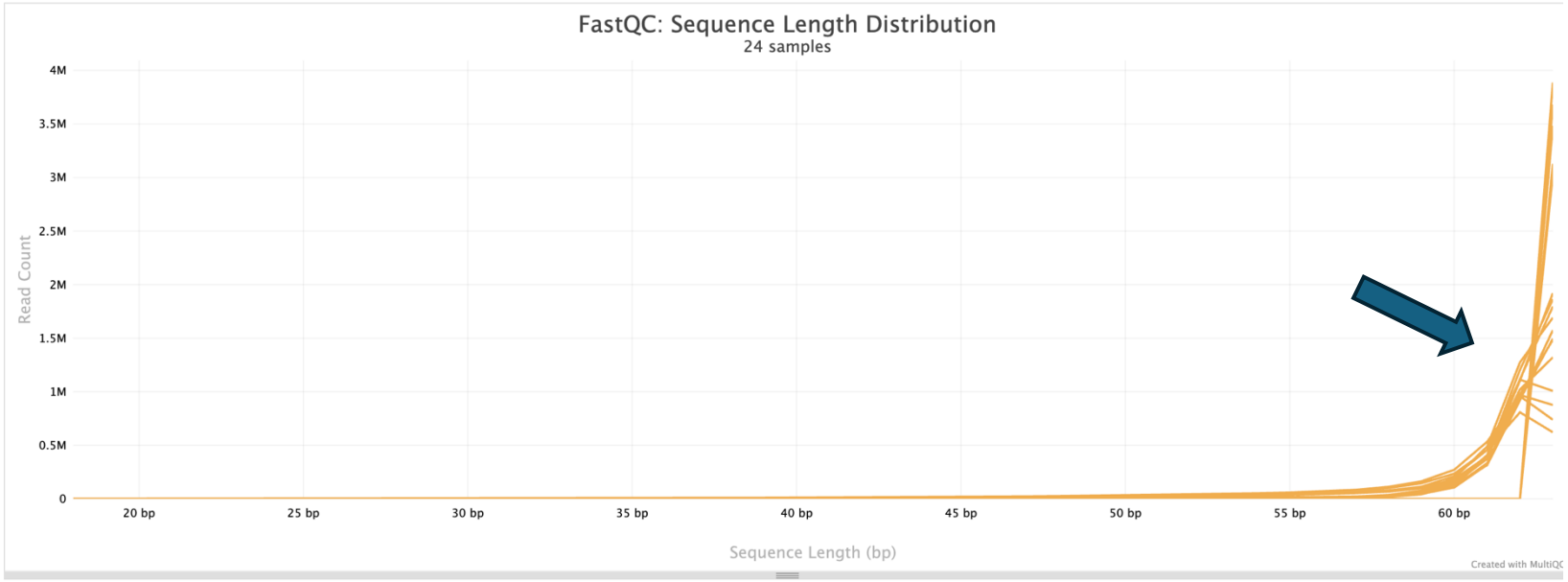
After Trimming



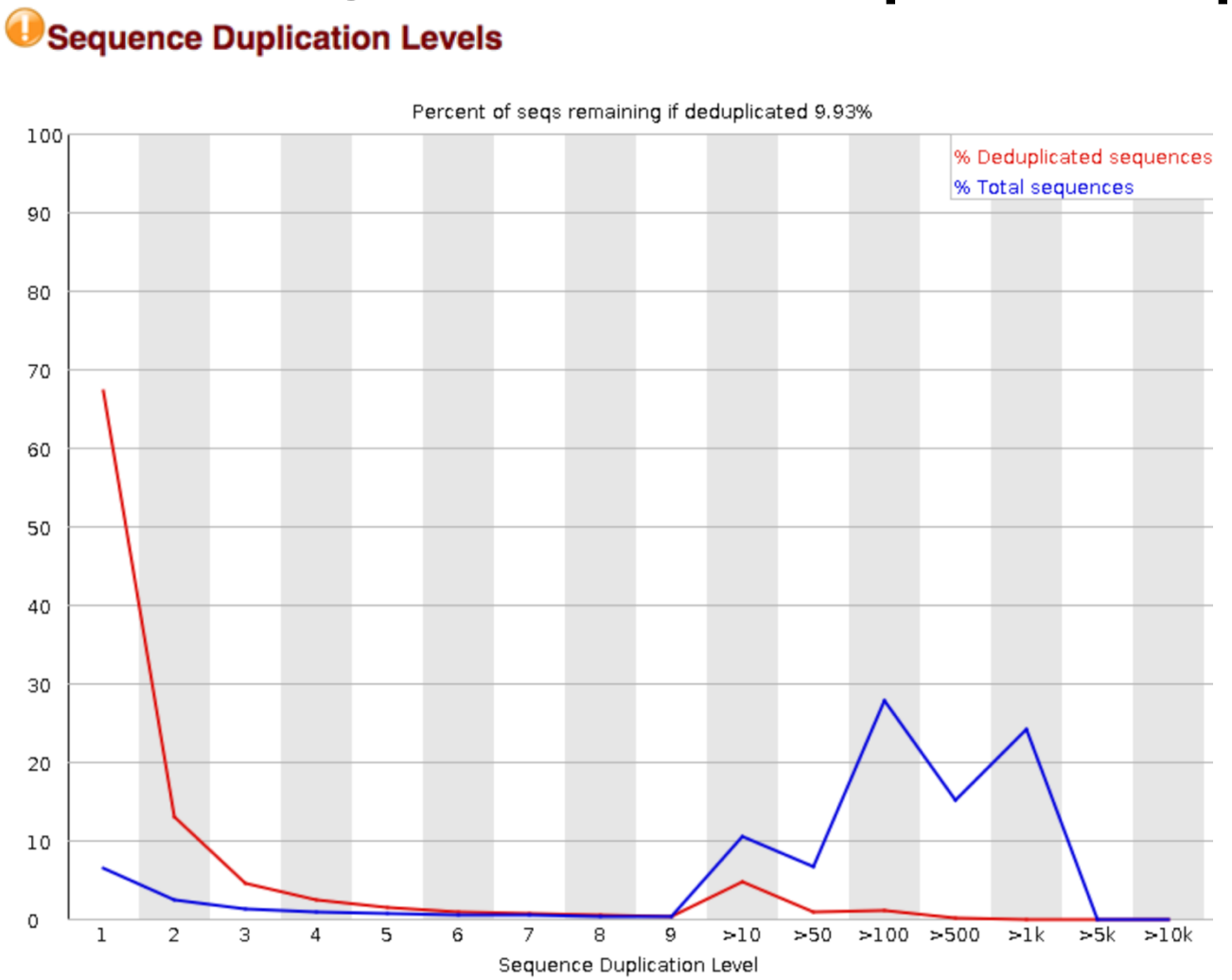
FastQC modules - Sequence length distribution

Sequence Length Distribution

Seq length distribution before and after trimming



FastQC modules - Sequence duplication levels



- Unique sequences
- Distinct sequences

ATTGGCCCA

ATTGAACCG
ATTGAACCG

GGCCAATTT
GGCCAATTT
GGCCAATTT

Num Seqs: 6
Num distinct: 3

FastQC modules - Overrepresented sequences

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
CTGGAGTGCAGTGGCTATTCACAGGCGCGATCCCACTACTGATCAGCACG	81075	1.377640541425052	No Hit
CCAGGCTGGAGTGCAGTGGCTATTCACAGGCGCGATCCCACTACTGATCA	61843	1.0508470429028616	No Hit
CTGGAGTCTTGGAAGCTTGACTACCCTACGTTCTCCTACAAATGGACCTT	57843	0.9828783452068983	No Hit
GGCTGGAGTGCAGTGGCTATTCACAGGCGCGATCCCACTACTGATCAGCA	56821	0.9655123429455799	No Hit
GGCAACCTGGTGGTCCCCCGCTCCCGGGAGGTCACCATATTGATGCCGAA	54501	0.9260904982819211	No Hit
CAGGCTGGAGTGCAGTGGCTATTCACAGGCGCGATCCCACTACTGATCAG	47865	0.8133304288043184	No Hit
CTTAGGCAACCTGGTGGTCCCCCGCTCCCGGGAGGTCACCATATTGATGC	47157	0.8012999693121329	No Hit
CCTTAGGCAACCTGGTGGTCCCCCGCTCCCGGGAGGTCACCATATTGATG	41412	0.703679927246306	No Hit
CTCAGGCTGGAGTGCAGTGGCTATTCACAGGCGCGATCCCACTACTGATC	39707	0.6747082698534017	No Hit
CACAAATTATGCAGTCGAGTTTCCCACATTTGGGGAAATCGCAGGGGTCA	35622	0.6052952373313993	No Hit

Agenda – Day 02

أكاديمية كاوست
KAUST ACADEMY

Time	Session	Topics & Activities
09:00-09:15	S1: Recap of Day 1	Review of previous day's content
09:15-10:00	S2: Quality Control	Sequencing quality metrics, Phred scores, technical variation, trimming, preprocessing
10:00-11:00	L3: Quality Control Hands-on	FastQC, fastp; inspect data quality metrics, trimming, adapter removal, pre/post-QC comparison View Lab Instructions →
11:00-12:00	S4: Reference Genome Alignment	Reference genomes, annotations, splice-aware alignment
14:00-15:00	L4: Genome Alignment Hands-on	Align reads, inspect BAM files, evaluate mapping metrics (samtools, STAR) View Lab Instructions →
15:00-15:45	S5: Read Counting & Quantification	Gene vs transcript quantification, expression matrices
15:45-16:30	L5: Quantification & Expression Matrix	Salmon, expression matrix inspection View Lab Instructions →
16:30-17:00	Q&A & Project Review	Questions and project progress support

أكاديمية كاوست
KAUST ACADEMY

Reference Genomes & Alignment

Day 02 – Session 03

Learning Objectives

By the end of this session, you should understand:

- What reference genomes are and why they could be essential for RNA-seq
- Where to find and download reference data
- What to do when NO reference genome exists
- Why RNA-seq alignment differs from DNA alignment
- Splicing and its impact on alignment
- Splice-aware alignment & pseudo-alignment approaches
- Quality metrics for evaluating alignment

Reference Genomes

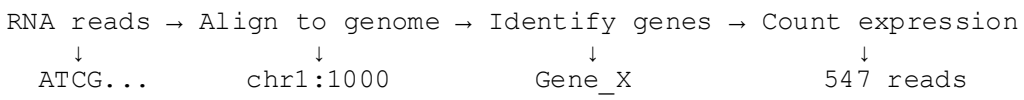
PART 1

What is a Reference Genome?

- **Definition**: A Reference Genome is a standardised representative sequence created from the genomes of multiple individuals, serving as a genetic baseline for comparing DNA changes.
- It is maintained and updated by scientific collaborations to improve accuracy in genomic analysis.

Why Do We Need Reference Genomes for RNA-seq?

- **Alignment**: Map millions of RNA reads to their genomic origin
- **Quantification**: Count reads per gene = expression level
- **Annotation**: Identify which genes are expressed
- **Comparison**: Enable cross-study reproducibility



Key Components You Need

1. Genome Sequence (FASTA format)

```
>chr1
ATCGATCGATCGTAGCTAGCTAGCTAGCTGATCGATCG...
>chr2
GCTAGCTAGCTAGCTGATCGATCGATCGATCGATCGATC...
```

2. Gene Annotation (GTF/GFF format)

gtf: *Gene Transfer Format*
gff: *General Feature Format*

```
chr1  HAVANA  gene        11869 14409  .  +  .  gene_id "ENSG00000223972"
chr1  HAVANA  transcript  11869 14409  .  +  .  gene_id "ENSG00000223972"
```

Critical: Genome and annotation must match the SAME version!

Major Reference Genome Databases

Database	Best for	URL
GENCODE	Human & Mouse (gold standard reference gene annotation)	https://www.gencodegenes.org
Ensembl	Wide species range, comprehensive	https://www.ensembl.org
NCBI Genome	Wide species range, comprehensive	https://www.ncbi.nlm.nih.gov/genome
UCSC	Integrated browser & tools	https://genome.ucsc.edu

Genome Builds & Versions

- **Human Genome Examples**
  - **hg19** (GRCh37) - older, still widely used
  - **hg38** (GRCh38) - current standard
- **Mouse Genome Examples**
  - **mm10** (GRCm38) - older
  - **mm39** (GRCm39) - current
- **CRITICAL BEGINNER MISTAKE**: Mixing genome versions
  - Using hg19 GTF with hg38 FASTA = WRONG!
  - All pipeline steps must use SAME version

Primary vs Full Assembly

- **Primary assembly**
  - Main chromosomes only (recommended for RNA-seq)
  - Excludes alternative haplotypes, patches, and scaffolds
  - Simplifies analysis and ensures consistency
- **Full assembly**
  - Includes alternative haplotypes, patches
  - More complex, larger
  - Usually unnecessary for standard RNA-seq

When No Reference Genome is Available

PART 2

Decision Tree: Which Approach?

```
Does a reference genome exist?
├─ YES → Use reference-based pipeline ✓
└─ NO
   ├─ Is there a close relative (<5-10% divergence)?
   │  └─ YES → Try related species genome
   │     ├─ Works well? ✓
   │     └─ Poor mapping? → Consider de novo
   │
   └─ NO close relative OR poor mapping
      ├─ Limited budget/time
      │  └─ De novo transcriptome assembly (Trinity)
      │
      └─ Good budget/genome important
         └─ De novo genome sequencing + assembly
```

Shell

Option 1: Using a Related Species Genome

- **When it works:**
  - Close evolutionary relationship (<5% divergence)
  - Well-annotated relative available
  - Exploratory analysis acceptable
- **Examples:**
  - Novel mosquito → Use *Anopheles gambiae*
  - New rodent species → Use mouse or rat
  - Rare primate → Use human or macaque
- **Limitations:**
  - Lower mapping rates
  - Bias against diverged sequences
  - May miss species-specific genes

Option 2: *De Novo* Transcriptome Assembly

- **Concept**:
  - Assemble RNA-seq reads directly into transcript sequences without a genome
- **Popular Tools:**
  - **Trinity** (most popular)
  - rnaSPAdes
  - SOAPdenovo-Trans
  - Trans-ABySS
  - RNA-Bloom

```
RNA-seq reads
    ↓
Quality control (Trimmomatic, fastp)
    ↓
De novo assembly (Trinity)
    ↓
Transcriptome FASTA output
    ↓
Quantification (back-map reads with Salmon/RSEM)
    ↓
Differential expression analysis
```

Shell

De Novo Transcriptome: Pros & Cons

- **Advantages:**
  - No genome needed
  - Captures expressed sequences
  - Identifies novel isoforms
- **Limitations:**
  - Only captures expressed genes
  - Fragmented assemblies possible
  - Difficult to distinguish paralogs
  - No genomic context (introns, regulatory regions)
  - Assembly quality varies with coverage

Option 3: *De Novo* Genome Sequencing

- **When to consider:**
  - Long-term research program on this organism
  - Budget available
  - Want to study genomic features
  - Community resource development
- **What you get:**
  - A reusable reference genome
  - Genomic context for all genes
  - Structural variant analysis
  - Comparative genomics possibilities

De Novo Genome Sequencing: Technologies

- **Long-read (PacBio HiFi / Oxford Nanopore)**
  - Quality: High (PacBio HiFi > 99.9%)
  - Contiguity: Chromosome-level possible
  - Best for: High-quality reference genomes
- **Hybrid Approach**
  - Combine HiFi + ONT Ultra-long + Hi-C or Optical Mapping
  - Current best approach for large genomes

De Novo Genome Assembly Workflow & Tools

1. DNA Extraction (high molecular weight)
   ↓
2. Library Preparation & Sequencing
   ↓
3. Quality Control (FastQC, Nanoplot, Filtlong, fasp)
   ↓
4. Genome Assembly (Hifiasm, Verkko, Flye)
   ↓
5. Polishing (Pilon, Racon)
   ↓
6. Quality Assessment (BUSCO, QUAST, merquery)
   ↓
7. Scaffolding/Hi-C (optional)
   ↓
8. Annotation (MAKER, BRAKER)

- **Assembly Tools:**
  - **SPAdes/dipSPAdes** - Short reads, small-medium genomes
  - **Flye** - Long reads (Nanopore)
  - **Hifiasm** - PacBio HiFi reads, ONT Ultra-long, Hi-C
  - **Verkko** - Hybrid Assembly
- **Quality Assessment:**
  - **BUSCO** - Checks for conserved genes (completeness)
  - **QUAST** - Assembly statistics (N50, contigs)
  - **Merqury** - *k*-mer based quality (QV score)
- **Annotation:**
  - **MAKER** - Combines evidence
  - **BRAKER** - Uses RNA-seq data
  - **AUGUSTUS** - Gene prediction

Key Considerations for *De Novo* Projects

- **Sample Quality:**
  - **Genome sequencing**: High molecular weight DNA
  - **Transcriptome**: High-quality RNA, RIN > 7

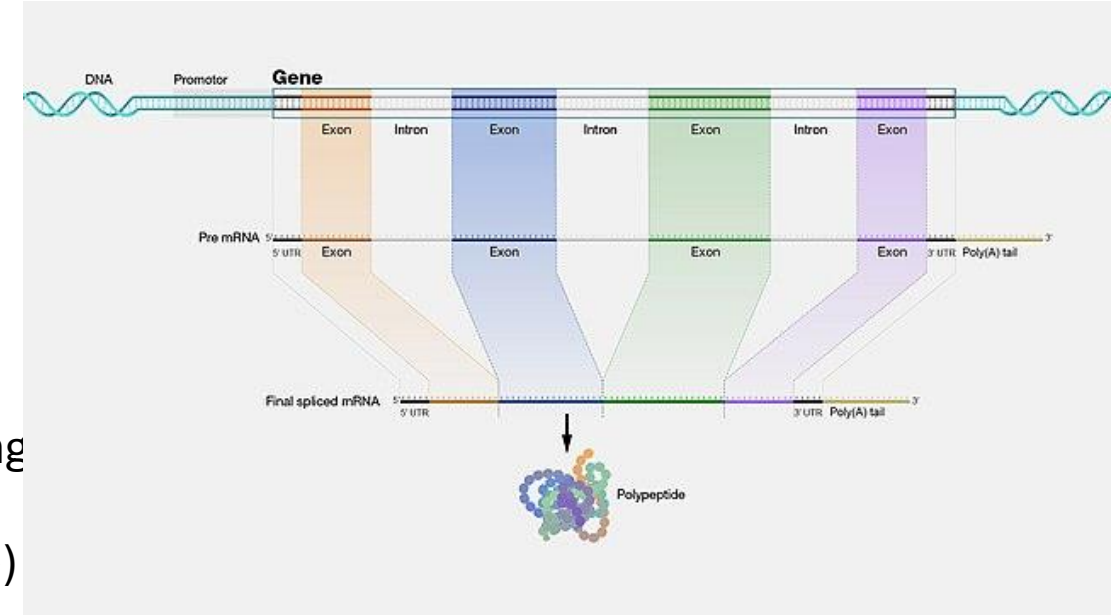
- **Biological Considerations:**
  - **Ploidy**: Diploid easier than polyploid
  - **Genome size**: Estimate before starting (flow cytometry)
  - **Heterozygosity**: High heterozygosity = harder assembly
  - **Repeats**: Repetitive genomes need long reads

Splicing

PART 3

What is Splicing?

- **Splicing**:
  - Process of removing introns from pre-mRNA
- **Key points**:
  - Occurs in the nucleus
  - Performed by the spliceosome complex
  - Creates mature mRNA for translation
- **Alternative splicing**:
  - One gene can produce multiple protein isoforms
- **Statistics (Human)**:
  - ~95% of multi-exon genes undergo alternative splicing
  - Average human gene: 8-9 exons, 7-8 introns
  - Introns can be very large (thousands to millions of bp)
  - Exons typically smaller (50-200 bp)

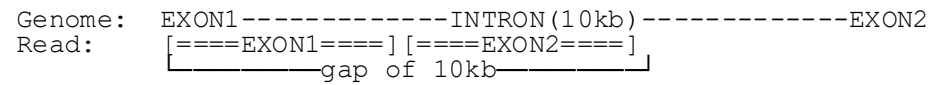


https://www.cshlp.org/ghg5_all/section/gene.shtml

https://microbeonline.com/exons-and-introns-functions-and-types/

The Alignment Challenge

- **Problem**:
  - RNA-seq reads come from mature mRNA (no introns),
  - but we align to genomic DNA (with introns)
- **Example**:



- **Standard aligners (BWA, Bowtie2) fail because**:
  - They expect continuous matches
  - Cannot handle large gaps (introns)
  - Don't recognize splice junctions
- **Solution**: Splice-aware aligners!

Splice-Aware Alignment

PART 4

What is Splice-Aware Alignment?

- **Definition**: Alignment method that can map reads across exon-exon junctions
- **Key features**:
  - Recognizes and handles introns (large gaps)
  - Identifies splice junctions
  - Uses gene annotation to guide alignment
  - Can discover novel splice junctions
- **Two main strategies**:
  - **Annotation-guided**: Uses GTF/GFF file with known junctions
  - ***De novo***: Discovers junctions from the data
- **Best practice**: Use annotation-guided when available!

Popular Splice-Aware Aligners

- **STAR** (Spliced Transcripts Alignment to a Reference)
  - Speed: Very fast (fastest available)
  - Memory: High (~30 GB for human)
  - Accuracy: Excellent
  - Best for: Standard RNA-seq workflows
- **HISAT2** (Hierarchical Indexing for Spliced Alignment)
  - Speed: Fast
  - Memory: Lower (~8 GB for human)
  - Accuracy: Excellent
  - Best for: When memory is limited
- **Recommendation for beginners**: Start with STAR or HISAT2

How Splice-Aware Aligners Work

- **Step 1**: Build genome index with junction database
  - Index the reference genome
  - Incorporate known splice junctions from GTF
  - Create search structures
- **Step 2**: Align reads
  - Try to align read continuously first
  - If fails, split read into segments
  - Find best mapping allowing large gaps at splice sites
- **Step 3**: Report alignments
  - Primary alignment + junction information
  - Calculate mapping quality scores

Building Genome Indices

- **STAR Index**

```
STAR --runMode genomeGenerate \
     --genomeDir ./star_index \
     --genomeFastaFiles genome.fa \
     --sjdbGTFfile annotation.gtf \
     --sjdbOverhang 99 \
     --runThreadN 8
```

- **Resources for indexing human genome:**
  - RAM: ~38GB
  - Time: ~3 hours using 2 cores
  - Index size: ~30GB

Running STAR Alignment

```
STAR --genomeDir ./star_index \
     --readFilesIn sample_R1.fastq.gz sample_R2.fastq.gz \
     --readFilesCommand zcat \
     --outFileNamePrefix sample_ \
     --outSAMtype BAM SortedByCoordinate \
     --runThreadN 16 \
     --outSAMattributes All \
     --quantMode GeneCounts
```

- **Key parameters:**

`--readFilesCommand zcat`	For gzipped files
`--outSAMtype BAM SortedByCoordinate`	Output sorted BAM
`--quantMode GeneCounts`	Generate gene counts directly

Advanced STAR Options

- **Two-pass alignment (recommended for better junction detection):**

```
##################################################
# First pass - basic alignment
STAR --genomeDir ./star_index \
     --readFilesIn sample_R1.fastq.gz sample_R2.fastq.gz \
     --readFilesCommand zcat \
     --runThreadN 16
##################################################
# Use detected junctions to create new index
# Second pass - re-align with novel junctions
STAR --genomeDir ./star_index \
     --readFilesIn sample_R1.fastq.gz sample_R2.fastq.gz \
     --readFilesCommand zcat \
     --runThreadN 16 \
     --sjdbFileChrStartEnd SJ.out.tab \
     --outFileNamePrefix sample_2pass_
##################################################
```

- **Benefit:**
  - It does not increase the number of detected novel junctions
  - but allows to detect more splices reads mapping to novel junctions

Understanding STAR Output Files

Key output files:

- **Aligned.sortedByCoord.out.bam**
  - Main alignment file (sorted)
  - Ready for downstream analysis
- **Log.final.out**
  - Summary statistics
  - Mapping rates, multi-mapping reads
- **SJ.out.tab**
  - Splice junctions detected
  - Novel and annotated junctions
- **ReadsPerGene.out.tab** (if `--quantMode GeneCounts`)
  - Gene-level read counts
  - Can use directly for DESeq2

STAR Log.final.out - Key Metrics

```
Number of input reads         | 50000000
Uniquely mapped reads number | 42500000
Uniquely mapped reads %       | 85.00%
Number of splices: Total      | 25000000
Number of splices: Annotated | 24000000
Mismatch rate per base, %     | 0.30%
Multi-mapping reads %         | 10.00%
Unmapped reads %              | 5.00%
```

What to look for:

- Uniquely mapped >70% (ideally >80%)
- Most splices annotated
- Low mismatch rate (<1%)
- High unmapped? Check contamination/wrong genome

Alignment File Format: SAM/BAM

- **SAM** (Sequence Alignment/Map): Text format
- **BAM**: Binary compressed version (smaller, faster)
- **Example SAM line:**

```
READ1  99  chr1  1000  255  50M1000N50M  =  1200  250  ATCG...  IIII...
```

- **Key fields:**
  - Read name
  - FLAG (mapping info)
  - Chromosome
  - Position
  - MAPQ (mapping quality)
  - **CIGAR string** (alignment pattern)

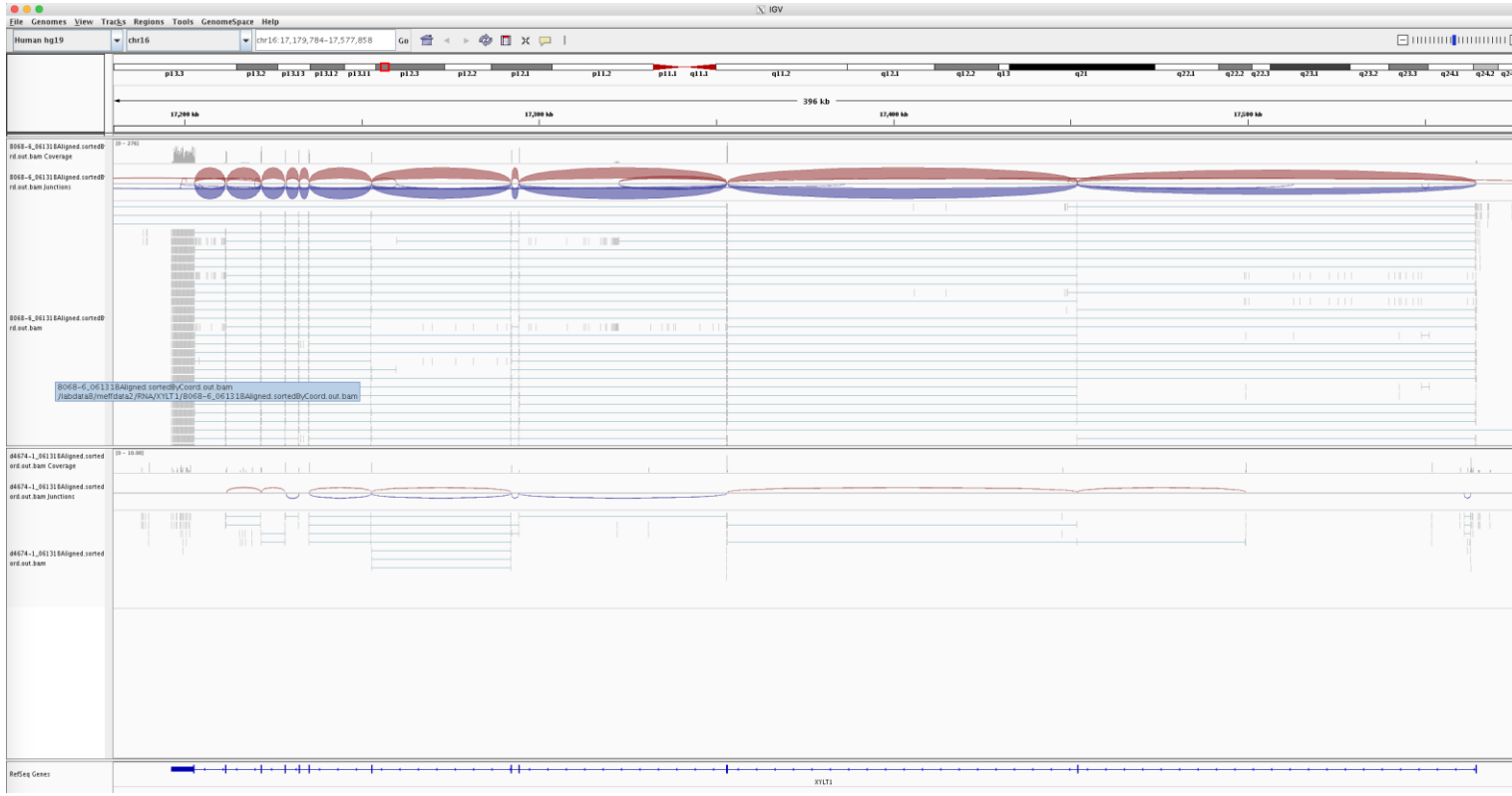
  - More details at **https://samtools.github.io/hts-specs/SAMv1.pdf**

Useful tip:
use samtools to read SAM & BAM Files

Visualization of Alignments

Using IGV (Integrative Genomics Viewer)

- **Steps**:
  - Load reference genome
  - Load BAM file + index (.bai)
  - Load gene annotation (GTF)
  - Navigate to gene of interest
- **What to look for**:
  - Read coverage patterns
  - Splice junctions (arcs connecting exons)
  - Novel junctions
  - Strand specificity
  - SNPs/variants
- **Tip**: Use "Sashimi plots" to visualize junction reads!



Pseudo-Alignment

PART 5

Pseudo-Alignment: A Faster Alternative

- **What if you only need gene/transcript counts?**
- **Two paradigms:**
  - **Genome alignment** → Count reads per gene
  - **Pseudo-alignment** → Count reads per transcript directly
- **Key insight**: For differential expression analysis, you don't always need exact genomic positions!
- **Result**: 10-100x faster quantification

What is Pseudo-Alignment?

- **Definition**: Lightweight method to quantify transcript abundance without full alignment
- **Key differences from traditional alignment:**
  - Maps to **transcriptome** (not genome)
  - Determines **which transcript** (not exact position)
  - Uses *k*-mer matching instead of base-by-base alignment
  - No CIGAR strings, no BAM files
- **Output**: Transcript/gene counts directly
- **Popular tools**: Salmon, Kallisto

How Pseudo-Alignment Works

- **Traditional alignment:**

```
Read → Genome → Find exact position → CIGAR string → BAM file
Time: High | Memory: High
```

- **Pseudo-alignment:**

```
Read → k-mers → Match to transcripts → Assign counts
Time: Low | Memory: Low
```

- **Key innovation**: Uses *k*-mer hashing
- Break read into *k*-mers (substrings of length *k*)
- Check which transcripts contain these k-mers
- Assign read to compatible transcript(s)

Salmon: Overview

- **Salmon** - Fast, accurate transcript quantification
- **Key features:**
  - Alignment-free quantification
  - Handles multi-mapping reads probabilistically
  - Corrects for sequence biases
  - Ultra-fast (minutes per sample)
  - Low memory (~4-8 GB)

Building Transcriptome Index

```shell
salmon index \
    -t gencode.v44.transcripts.fa.gz \
    -i salmon_index \
    -k 31 \
    --gencode
```

Shell

Running Salmon Quantification

```
salmon quant \
    -i salmon_index \
    -l A \
    -1 sample_R1.fastq.gz \
    -2 sample_R2.fastq.gz \
    -o sample_salmon_quant \
    -p 8 \
    --validateMappings \
    --gcBias \
    --seqBias
```

Key parameters:

- `-l A`: Auto-detect library type
- `--validateMappings`: More accurate
- `--gcBias`: Correct for GC content bias
- `--seqBias`: Correct for sequence-specific bias

Salmon Output

```
Name              Length  EffectiveLength  TPM    NumReads
ENST00000456328   1657    1461.000         0.234  5.000
ENST00000450305   632     436.000          0.000  0.000
```

Shell

Length: actual nucleotide length of transcript
Effective Length: the computed *effective* length of the target transcript.

When to Use Each Approach

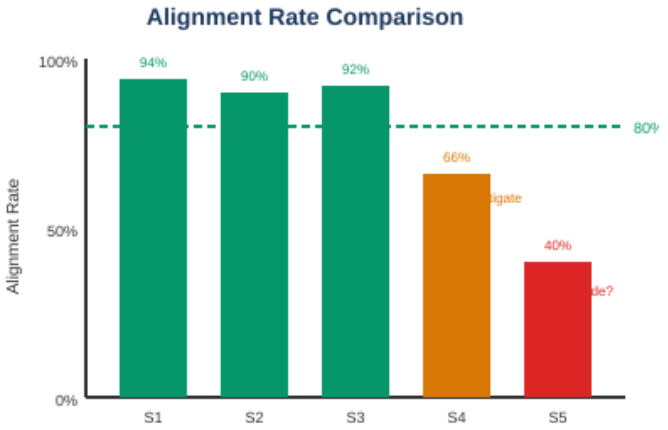
- **Use Traditional Alignment (STAR/HISAT2) when:**
  - You need genomic positions (variant calling, visualization)
  - Discovering novel splice junctions is important
  - You want to visualize in IGV
  - Studying allele-specific expression
  - Need to detect fusion genes
- **Use Pseudo-Alignment (Salmon/Kallisto) when:**
  - Only need differential expression analysis
  - Working with many samples (fast!)
  - Limited computational resources
  - Focus on transcript-level quantification
  - Time is critical

Quality Control & Best Practices

PART 6

Quality Metrics to Check

1. **Overall Mapping Rate**
   - **Good**: >70% uniquely mapped
   - **Excellent**: >80% uniquely mapped
   - **Problem**: <60% (wrong genome? contamination?)
2. **Multi-mapping Rate**
   - **Normal**: 5-15% for mammalian genomes
   - **High**: >20% (repetitive sequences, gene families)
3. **Splice Junction Metrics**
   - **% Annotated junctions**: >90% expected
   - **Novel junctions**: Some expected, but not majority
4. **Mismatch Rate**
   - **Good**: <1%
   - **Problem**: >2% (sequencing quality? wrong genome?)



Quality Metrics (cont.)

5. **Strand Specificity (if stranded library)**
   - Check reads align to correct strand
   - Use infer_experiment.py from RSeQC
6. **Gene Body Coverage**
   - Should be relatively uniform
   - 3' bias indicates degraded RNA
   - Use geneBody_coverage.py from RSeQC
7. **Insert Size Distribution (paired-end)**
   - Should match library prep
   - Typically 150-400 bp for RNA-seq
8. **Duplication Rate**
   - Higher than DNA-seq (normal for highly expressed genes)
   - Very high (>40%) may indicate PCR bias

Post-Alignment QC Tools

- **RSeQC Package**

```
# Infer experiment (strand specificity)
infer_experiment.py -i sample.bam -r genes.bed
# Gene body coverage
geneBody_coverage.py -i sample.bam -r genes.bed -o sample
# Read distribution
read_distribution.py -i sample.bam -r genes.bed
```

- **Qualimap 2**

```
qualimap rnaseq -bam sample.bam -gtf annotation.gtf -outdir qc_out
```

- **PICARD**

```
Flag PCR duplicates
```

- **MultiQC** - Aggregates all QC reports

```
multiqc ./project_dir
```

Common Alignment Issues

Issue 1: Low Mapping Rate (<60%)

- **Possible causes:**
  - Wrong reference genome (species mismatch)
  - Wrong genome version
  - Contamination (rRNA, adapters)
  - Poor quality reads
- **Solutions:**
  - Verify organism and genome version
  - Check for contamination (BLAST unmapped reads)
  - Perform adapter trimming
  - Check FastQC reports

Issue 2: High Multi-mapping Rate (>25%)

- **Possible causes:**
  - Repetitive genome regions
  - Paralogous genes
  - Inadequate read length
- **Solutions:**
  - Check if expected for your organism
  - Consider using longer reads
  - Use tools that handle multi-mappers (e.g., Salmon)

Issue 3: Many Novel Junctions

- **Possible causes:**
  - Novel isoforms (good!)
  - Incomplete annotation
  - Sequencing errors
  - Genomic contamination
- **Solutions:**
  - Validate with splice site motifs (GT-AG)
  - Check junction support (# reads)
  - Compare across samples

Summary: Key Takeaways

- **Reference genomes are essential** for standard RNA-seq analysis
- **Document everything**: versions, sources, dates
- **Check compatibility**: genome and annotation must match
- **Non-model organisms**: evaluate all options (related species, *de novo*)
- **Splice-aware alignment** (STAR/HISAT2) needed for RNA-seq
- **Pseudo-alignment** (Salmon/Kallisto) is fast and accurate for quantification
- **Quality metrics matter**: check mapping rates immediately
- **Choose the right approach** based on your needs and resources

Hands-on: Genome Alignment

From raw reads to aligned reads

Day 02 – Lab 04

Lab 4: Genome Alignment Hands-on

Learning Objectives

- Build a STAR genome index for chromosome 11
- Align RNA-seq reads to the reference genome using STAR
- Understand BAM file format and alignment statistics
- Use samtools to inspect and manipulate BAM files

Agenda – Day 02

أكاديمية كاوست
KAUST ACADEMY

Time	Session	Topics & Activities
09:00-09:15	S1: Recap of Day 1	Review of previous day's content
09:15-10:00	S2: Quality Control	Sequencing quality metrics, Phred scores, technical variation, trimming, preprocessing
10:00-11:00	L3: Quality Control Hands-on	FastQC, fastp; inspect data quality metrics, trimming, adapter removal, pre/post-QC comparison View Lab Instructions →
11:00-12:00	S4: Reference Genome Alignment	Reference genomes, annotations, splice-aware alignment
14:00-15:00	L4: Genome Alignment Hands-on	Align reads, inspect BAM files, evaluate mapping metrics (samtools, STAR) View Lab Instructions →
15:00-15:45	S5: Read Counting & Quantification	Gene vs transcript quantification, expression matrices
15:45-16:30	L5: Quantification & Expression Matrix	Salmon, expression matrix inspection View Lab Instructions →
16:30-17:00	Q&A & Project Review	Questions and project progress support

Read Counting & Quantification

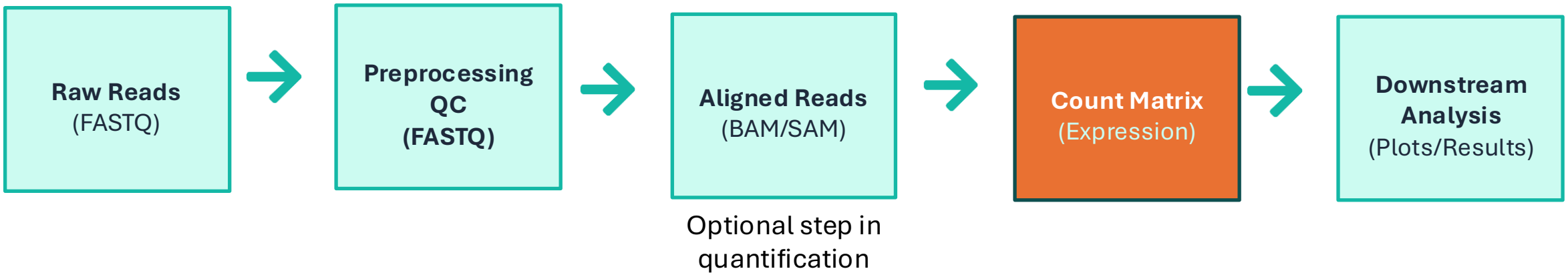
From Aligned Reads to Expression Matrices

Day 02 – Session 04

What is Read Counting?

The Core Question

After alignment, we need to answer: How many reads came from each gene or transcript?



The count values form the basis for all downstream expression analysis

What is Read Quantification?

The process of determining how many RNA sequencing reads originate from each gene or transcript in a sample, forming the foundation of differential expression analysis.

Traditional Approach

1. Align reads to genome (STAR, HISAT2)
2. Count overlapping reads (HTSeq, featureCounts)
3. Generate count matrix

⏱️ Slower, disk-intensive

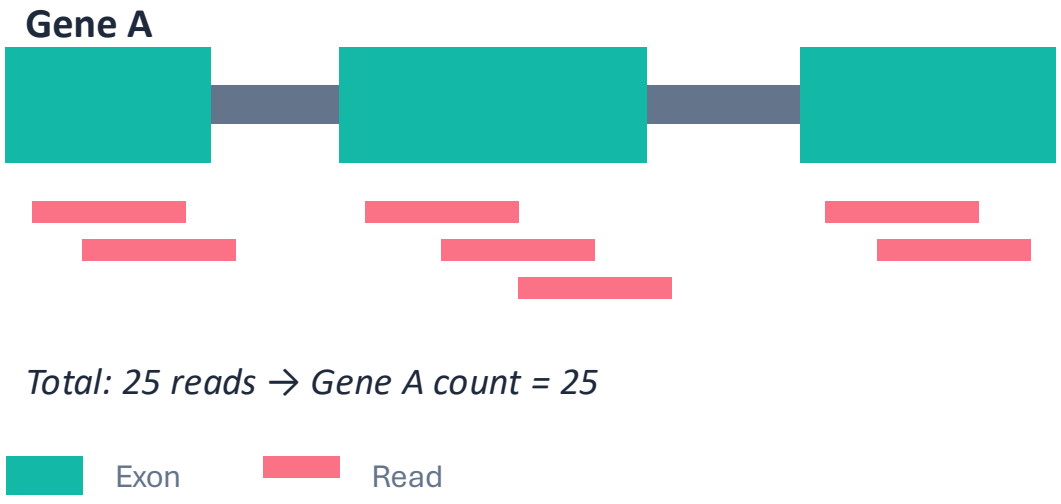
Pseudo Approach (Salmon/kallisto)

1. Quasi-mapping to transcriptome
2. Direct abundance estimation
3. Bias correction built-in

⚡ 10-100x faster, less memory

Gene-Level Quantification

How it works:



Characteristics

- All reads mapping to any isoform are summed
- Collapses transcript complexity into one value per gene
- Simpler interpretation: one number = one gene
- Loses isoform-specific information

Common Tools

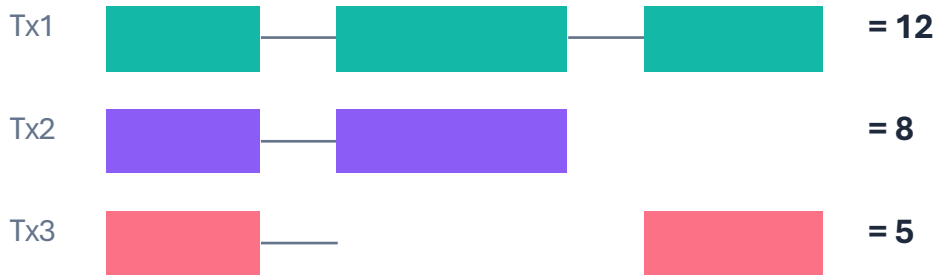
featureCounts (Subread) - fast, simple
HTSeq-count - flexible options

When to use: Standard differential expression analysis, when isoform differences are not the focus

Transcript-Level Quantification

How it works:

Gene A has 3 isoforms:



Uses probabilistic models to assign reads to specific transcripts, even when they share exons

Gene-level sum: 12 + 8 + 5 = 25

Characteristics

- Quantifies each transcript isoform separately
- Uses EM algorithm or similar to handle shared reads
- Enables differential isoform usage analysis

Common Tool

Salmon — *Don't count . . . quantify!*

When to use: Alternative splicing analysis, isoform switching studies, single-cell RNA-seq, or when biological question involves transcript-level changes

Gene-level vs Transcript-level Quantification

Gene-level

Sums counts from ALL transcripts of a gene

✓ More robust statistics
✓ Easier interpretation
✓ Better for differential expression
✓ Lower false discovery rate
✓ Standard for most analyses

Use for: DGE, pathway analysis, biomarker discovery

Transcript-level

Individual isoform abundance estimates

✓ Detects isoform switching
✓ Splice variant analysis
⚠ Higher uncertainty
⚠ More complex statistics
⚠ Requires specialized tools

Use for: DTU, splicing events, cancer isoforms

Building an Expression Matrix

The expression matrix is the fundamental data structure for RNA-seq analysis

		Samples (columns)			
		Ctrl_1	Ctrl_2	Treat_1	Treat_2
Genes (rows)	ACTB	1523	1487	2341	2198
	TP53	234	256	89	102
	GAPDH	8932	9102	8756	8821
	MYC	456	423	1234	1156

Key Properties

- Rows = Genes/Transcripts
- Columns = Samples
- Values = Raw counts (integers)
- Typically stored as a matrix or data frame

Typical dimensions:
20,000 genes × n samples
100,000+ transcripts × n samples

This matrix is the input for normalization, QC, and differential expression analysis

Salmon

Wicked-fast transcript quantification from RNA-seq reads

Ultra-Fast

10-100x faster than alignment-based methods

Accurate

Bias-aware quantification with GC correction

Memory Efficient

No large BAM files required

Lightweight

Quasi-mapping instead of full alignment

Patro et al., Nature Methods (2017) • combine-lab.github.io/salmon

How Salmon Works: The Two-Phase Algorithm

PHASE 1: Online/Streaming

- Quasi-mapping of reads
- Initial abundance estimates
- Learn sample-specific biases
- Build equivalence classes

→

PHASE 2: Offline Inference

- Apply bias corrections
- EM/VBEM optimization
- Final abundance estimates
- Bootstrap/Gibbs sampling

Built-in Bias Corrections

GC Content

Fragment GC composition bias

`--gcBias`

Sequence

5' and 3' end sequence effects

`--seqBias`

Positional

Coverage variation along transcript

`--posBias`

Salmon: from FASTQ to Counts

1 Index

Build index from transcriptome FASTA (one-time)

```
salmon index -t transcripts.fa -i salmon_index
```

2 Quantify

Map reads and estimate abundances per sample

```
#Based on fastq reads:
salmon quant -i salmon_index -l <LIBTYPE> --gcBias -1 reads_1.fq -2 reads_2.fq -o output

#Based on Aligned reads:
salmon quant -t transcripts.fa -l <LIBTYPE> --gcBias -a aln.bam -o output
```

3 Import

Aggregate to gene-level with tximport (R)

```
txi <- tximport(files, type="salmon", tx2gene=tx2gene)
```

Output: quant.sf contains Name, Length, EffectiveLength, TPM, NumReads per transcript

Building A Gene Expression Matrix

Combining multiple samples into a single matrix for downstream analysis

Expression Matrix Structure

	Sample_1	Sample_2	Sample_3	...	Sample_n
Gene_A	245	312	198	...	287
Gene_B	1024	856	1102	...	943
Gene_C	56	78	45	...	62
⋮	⋮	⋮	⋮	⋮	⋮
Gene_m	421	389	456	...	402

Key Concepts

Rows: Genes/Transcripts

Columns:
Samples

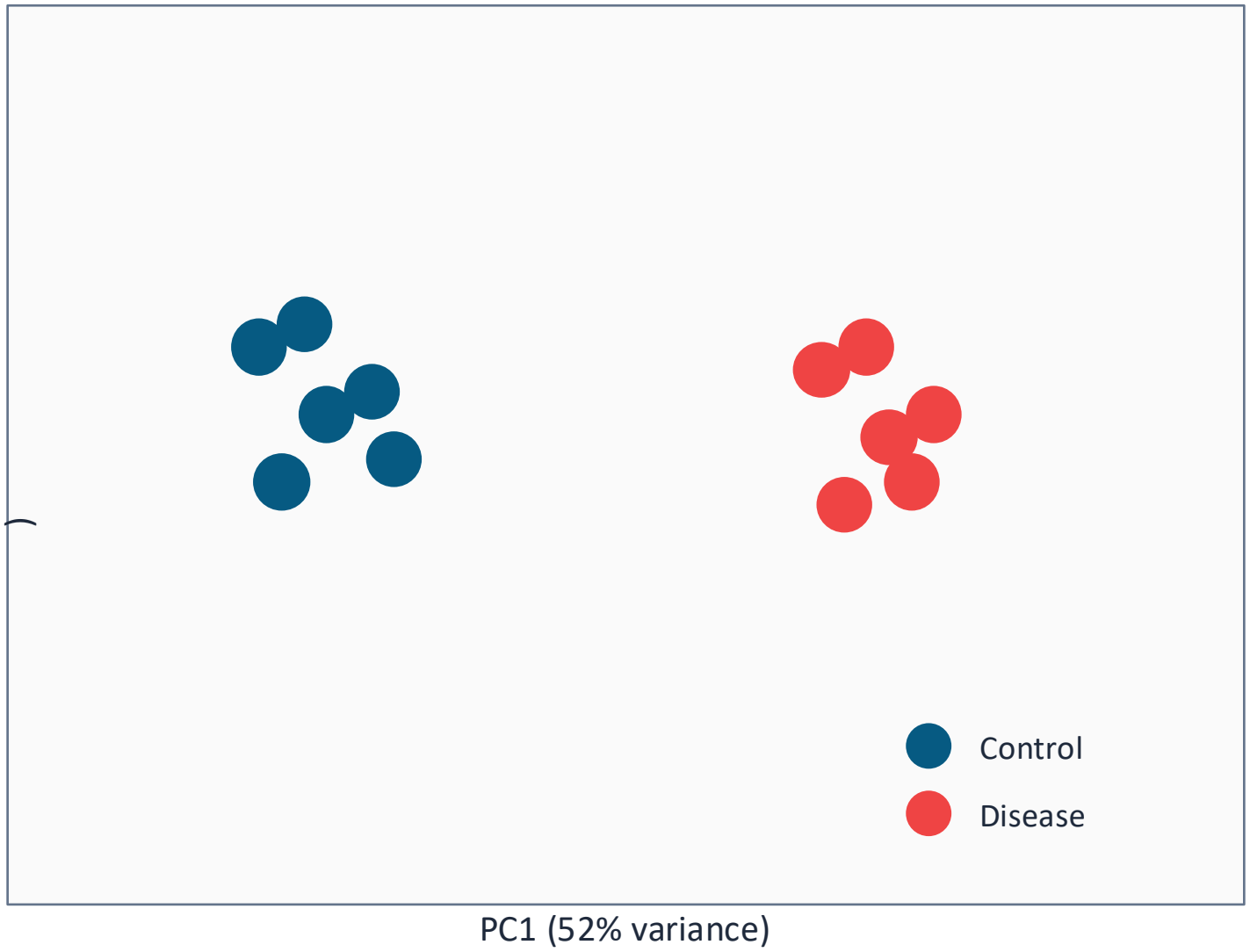
Values:
Raw counts (not normalized)

tximport handles:
- Transcript → Gene aggregation
- Length offset calculation
- Isoform usage correction

```
# R code with tximport
txi <- tximport(files, type="salmon", tx2gene=tx2gene)
dds <- DESeqDataSetFromTximport(txi, colData=samples, design=~condition)
```

Quality Control

Quality Control: Good PCA Results

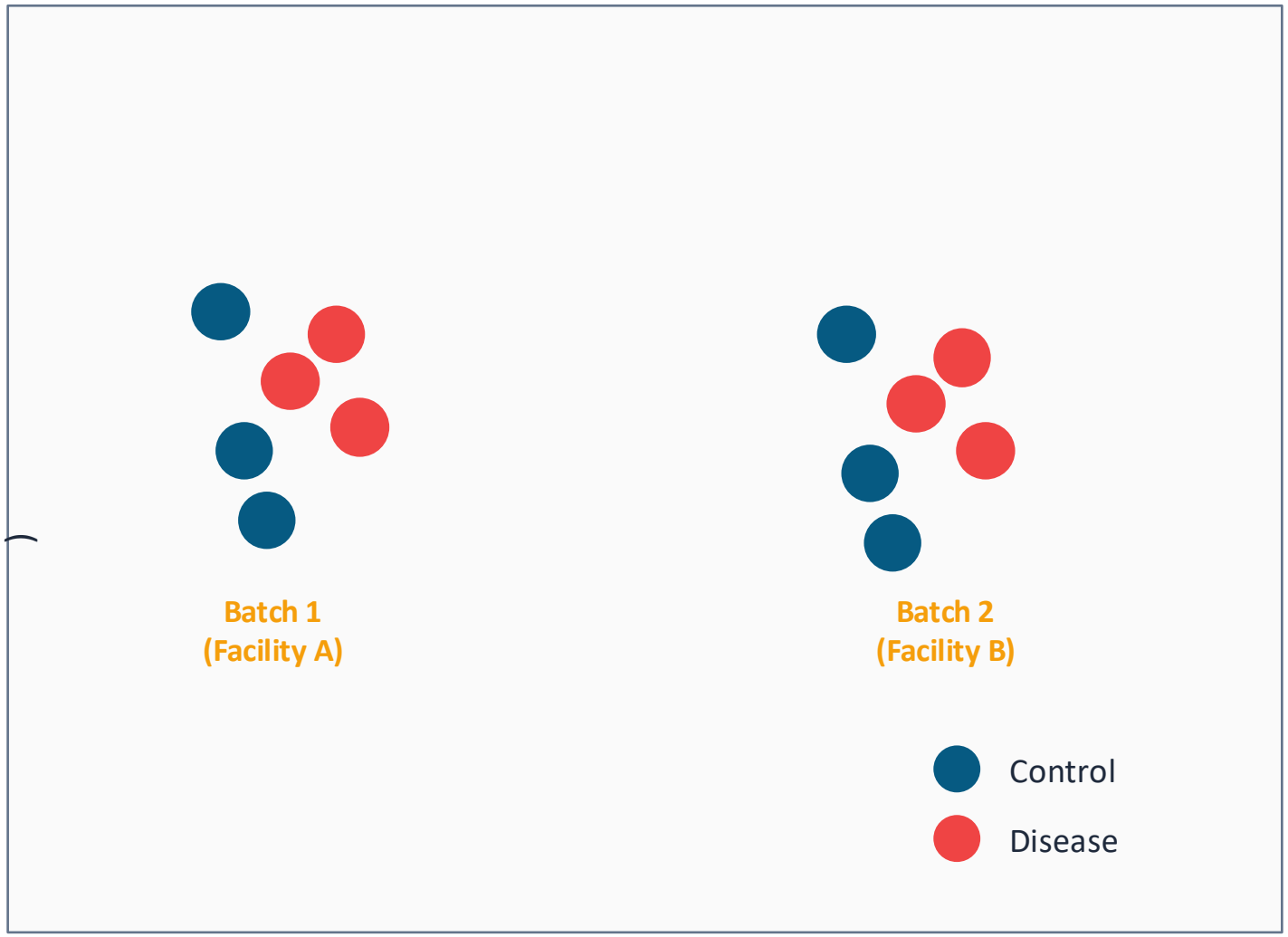


✓ What's Good

Clear separation between conditions

- Biological signal dominates PC1
- Samples cluster by phenotype
- No batch effects visible
- Tight within-group clustering
- High variance explained (52%)

Quality Control: Problematic PCA - Batch Effects



✗ Problems Detected

Samples cluster by BATCH not condition!

- Technical variation > biological
- PC1 explains batch, not biology
- Controls & disease mixed
- False positives likely

Solution: ComBat, limma::removeBatchEffect

When Quantification Goes Wrong

✗ Analysis Failures

No GC bias correction	Spurious DE in GC-rich genes	→ Add --gcBias flag
Used raw counts directly	Length bias confounds results	→ Use tximport offsets
Ignored batch effects	False discovery rate ~40%	→ Include batch in model
Mixed transcript/gene IDs	Incorrect aggregation	→ Consistent annotation

⚠ Lesson: Always validate with PCA/heatmaps before proceeding to differential expression!

Quality Considerations for Expression Matrices

Things to Check

- Library size variation (total counts per sample)
- Assignment rate (% reads mapped to features)
- Multi-mapping rates (handled consistently?)
- Sample correlations (detect outliers)
- Gene detection rates across samples

Common Issues

- Inconsistent gene IDs between samples
- Missing values (genes with 0 in all samples)
- Batch effects from different sequencing runs
- Strand-specificity mismatches
- Annotation version differences

Recommended QC Visualizations

- **Library size barplot** – identify samples with abnormally low/high counts
- **PCA/MDS plot** – check sample groupings, identify outliers and batch effects
- **Sample correlation heatmap** – verify replicates cluster together

Gene-Level Quantification

Simpler, higher statistical power, best for standard differential expression when isoform-level detail isn't needed.

Transcript-Level Quantification

Essential for splicing analysis and isoform studies. Can be aggregated to gene level

Expression Matrix

Genes/transcripts × samples matrix of raw counts. The foundation for all downstream analysis.

Quality Control is Critical

Always check library sizes, assignment rates, and sample correlations before proceeding to DE analysis.

Next: Normalization and Differential Expression Analysis

Hands-on:
Read Counting & Quantification

From Aligned Reads to Expression Matrices

Lab 5: Counting & Expression Matrix Construction

Learning Objectives

- Understand transcript vs gene-level quantification
- Run Salmon for fast transcript quantification
- Use tximport to aggregate transcript counts to gene level
- Build and explore an expression count matrix

Applied RNA-seq project scenarios.

Which genes and pathways are affected by TDP-43 knockout in your assigned chromosome ?

Project Background

- TDP-43 is a ubiquitously expressed RNA-binding protein that regulates thousands of target genes across the genome.

- During the labs we focus on data from chromosome 11
- Each group will analyze data from one of the chromosomes, check your chromosome (group) number.

- Your challenge is to identify:
  - Which genes on your chromosome are affected by TDP-43 knockout
  - What biological processes are disrupted
  - Whether your chromosome has unique or particularly strong responses to TDP-43 loss
  - Potential links to ALS/FTD pathology based on affected genes

Project Tasks 2

أكاديمية كاوست
KAUST ACADEMY

Understand the processed data:

1. How many genes were expressed per sample
2. What was the minimum and maximum alignment rate across the samples
3. What was the duplication level in the samples
4. How many genes and how many transcripts are expressed?
5. Plot the distribution of gene expression data across the samples
6. Show what samples are most similar.
7. Is the gene expression data that you have obtained normalized?

Understand the input data:

1. Get statistics about your chromosome, number of nucleotides, genes, transcripts, etc.

2. Get stats on your fastq datasets.

Team formation

- Visit the project web page and identify your group:
  https://bioinfo-kaust.github.io/academy-stage3-2026/html/projects.html

- Please let us know if your group need to be changed

SCAN ME

Thank you