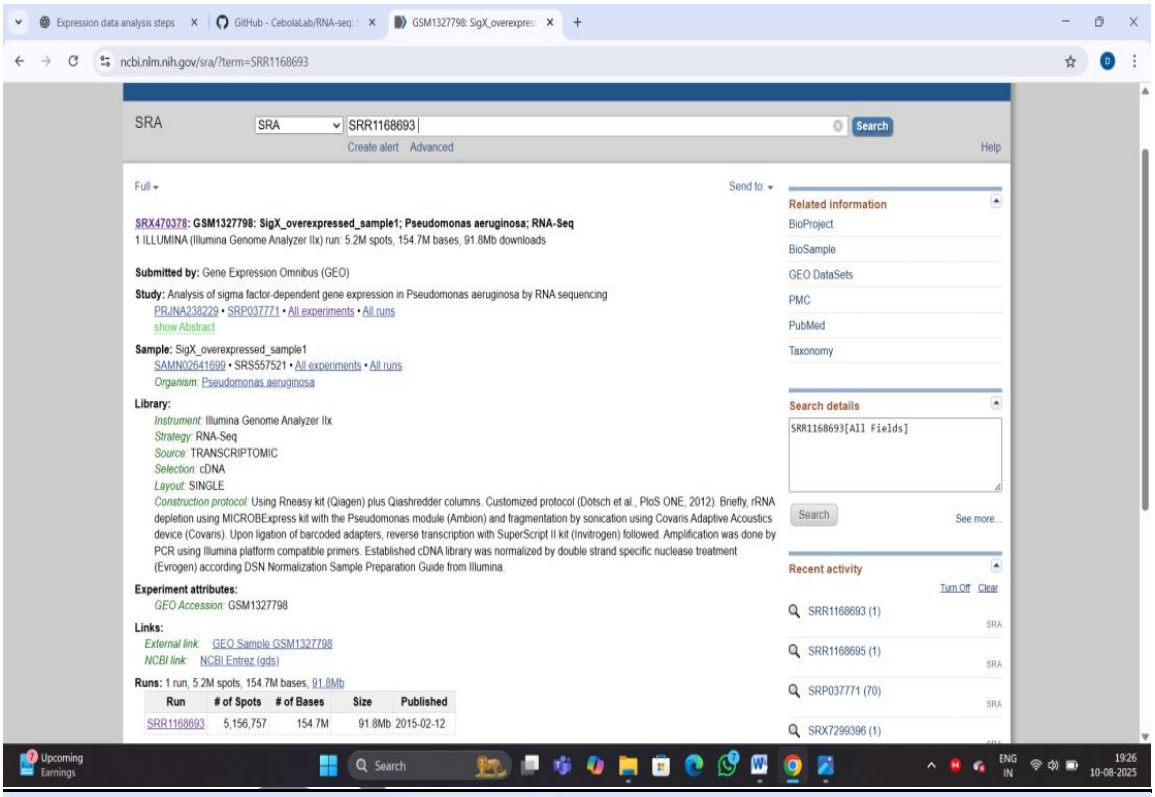
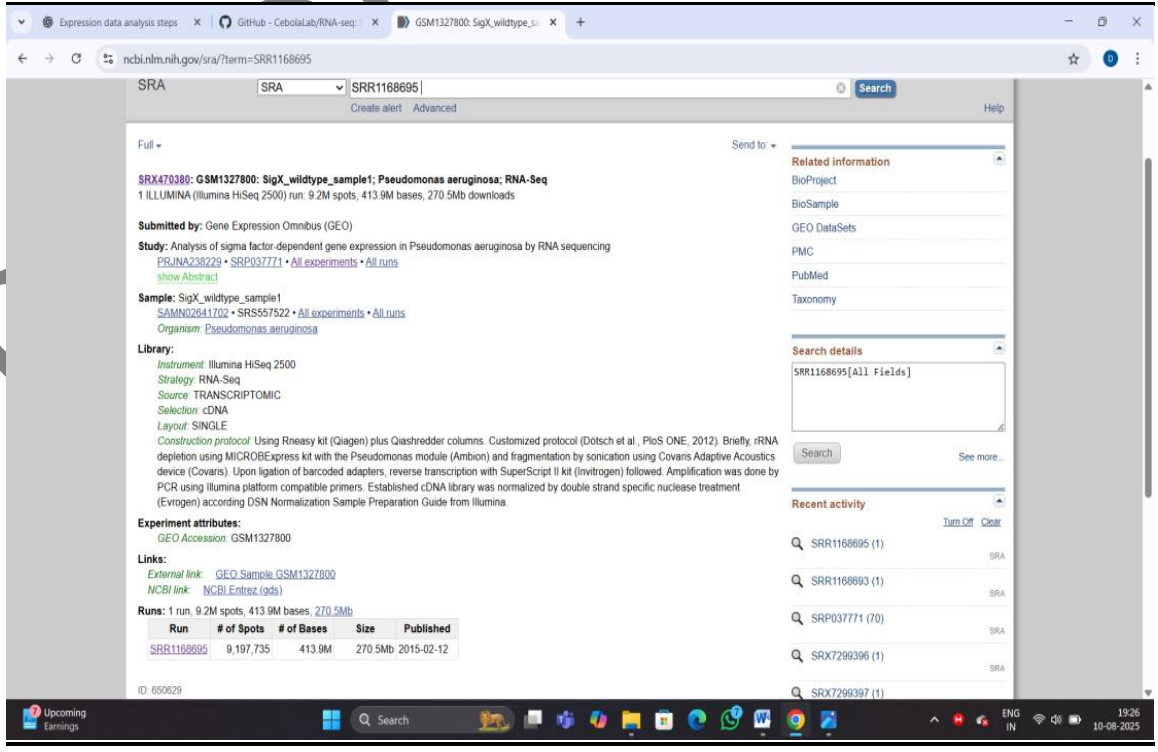
Expression Data Analysis and Enrichment

1. Dataset Identification





For this analysis, RNA-seq data related to *Pseudomonas aeruginosa* (**sigma factor-dependent gene expression)** were selected from the **NCBI SRA database**. The study investigates transcriptional changes in *Pseudomonas aeruginosa* between a **SigX-overexpressed mutant** and a **wild-type** strain.

Selected Runs:

- **SRR1168693** - SigX-overexpressed mutant (Illumina Genome Analyzer IIx, 5.2M spots, 154.7M bases)
- **SRR1168695** - SigX wild-type (Illumina HiSeq 2500, 9.2M spots, 413.9M bases)

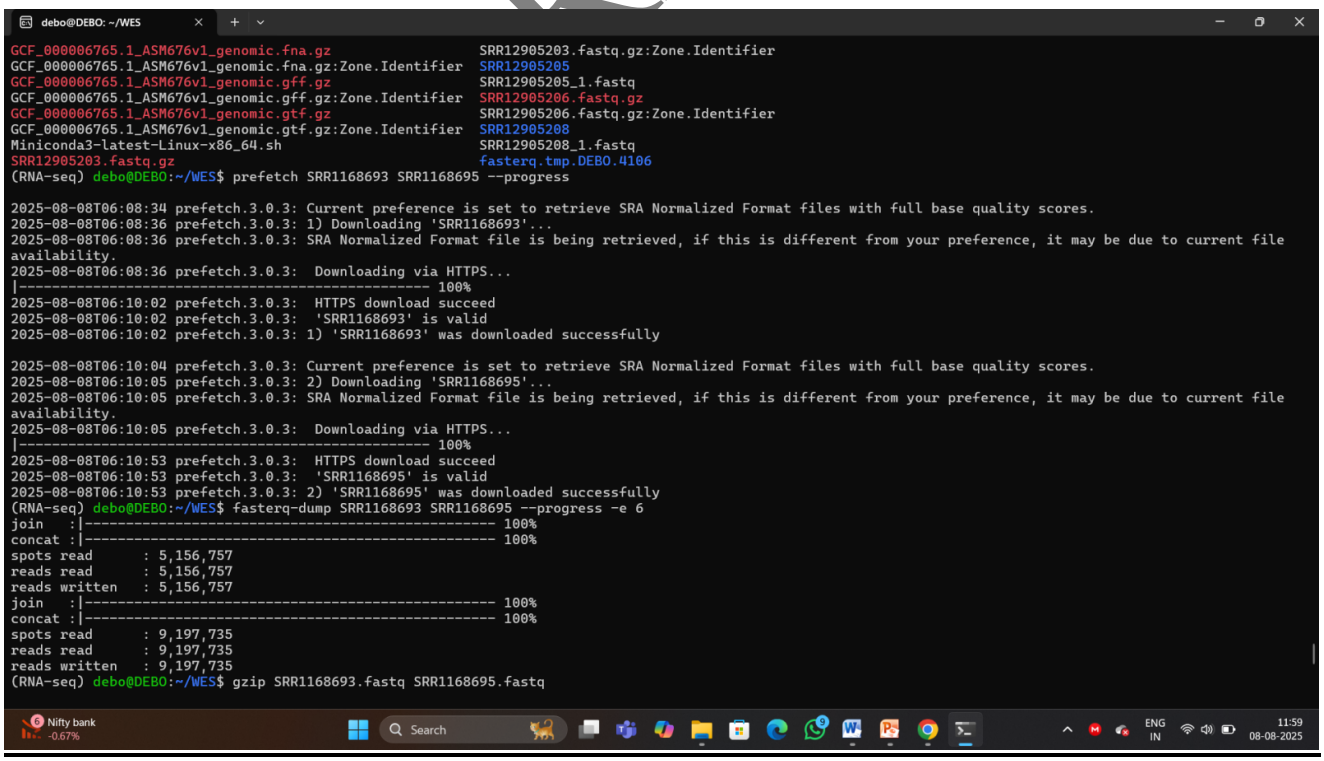
Source:

- Bio Project: PRJNA322892
- GEO Series: GSE703771

Figure 1: SRA record for SRR1168693 (mutant strain)
Figure 2: SRA record for SRR1168695 (wild-type strain)

Interpretation:
The dataset provides high-quality RNA-seq reads for both mutant and wild-type conditions, suitable for downstream quality control, alignment, quantification, and differential expression analysis.

2. Data Download



The selected SRA datasets (**SRR1168693 – mutant** and **SRR1168695 – wild-type**) were downloaded from the NCBI Sequence Read Archive (SRA) using the **SRA Toolkit**.

Commands Used:

Download SRA files- prefetch SRR1168693 SRR1168695 –progress

Convert SRA to FASTQ format using 6 threads- fasterq-dump SRR1168693 SRR1168695 --progress -e 6

Compress FASTQ files- gzip SRR1168693.fastq SRR1168695.fastq

Process Explanation:

1. **Prefetch** retrieves .sra files from NCBI servers.
2. **Fasterq-dump** converts them to .fastq for compatibility with downstream QC and alignment tools.
3. **gzip** compresses the files to save storage space.

Result:

- SRR1168693: 5,156,757 reads
- SRR1168695: 9,197,735 reads

Figure 3: Terminal output showing successful data download and conversion to FASTQ format.

Interpretation:
Raw RNA-seq reads for both mutant and wild-type strains were successfully downloaded and are ready for quality control and trimming.

3. Quality Control (QC) using FastQC and MultiQC

After downloading the raw sequencing data (SRR1168693.fastq.gz and SRR1168695.fastq.gz), initial quality assessment was performed.

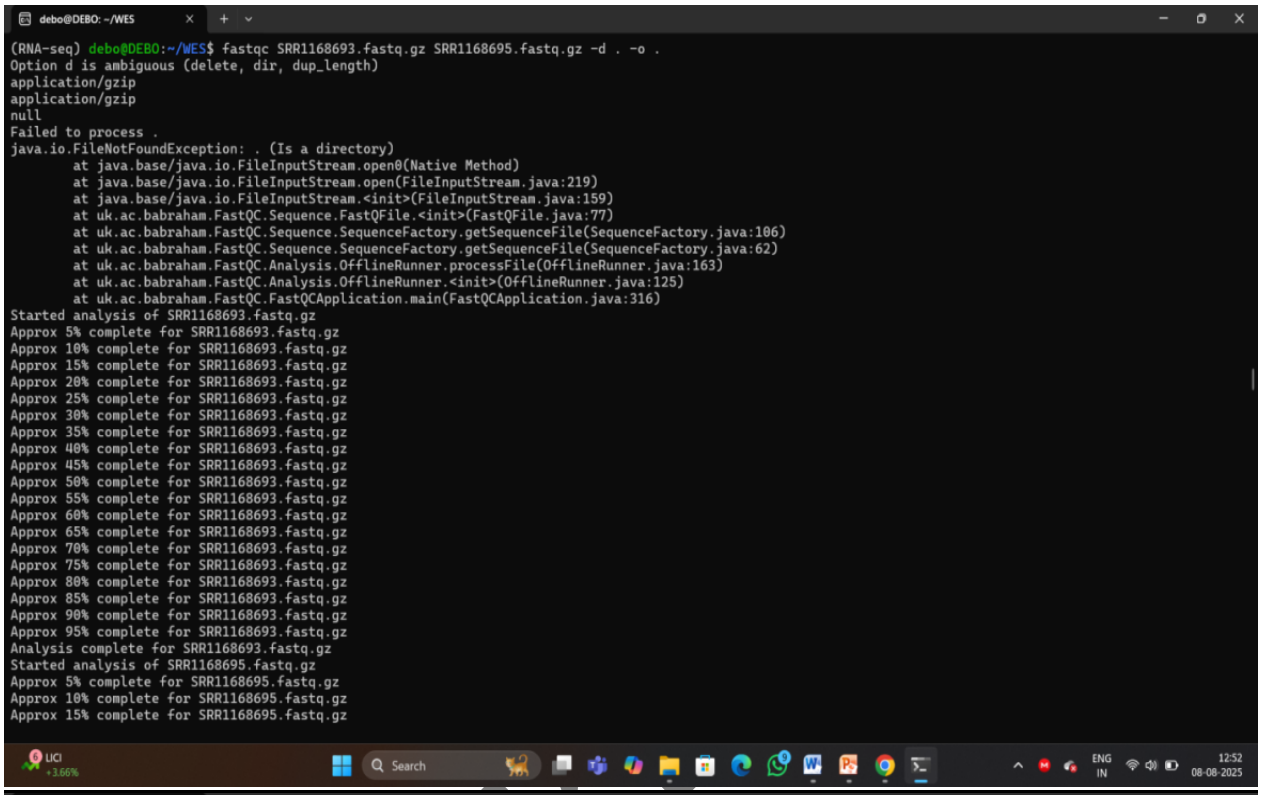
3.1 Individual QC with FastQC

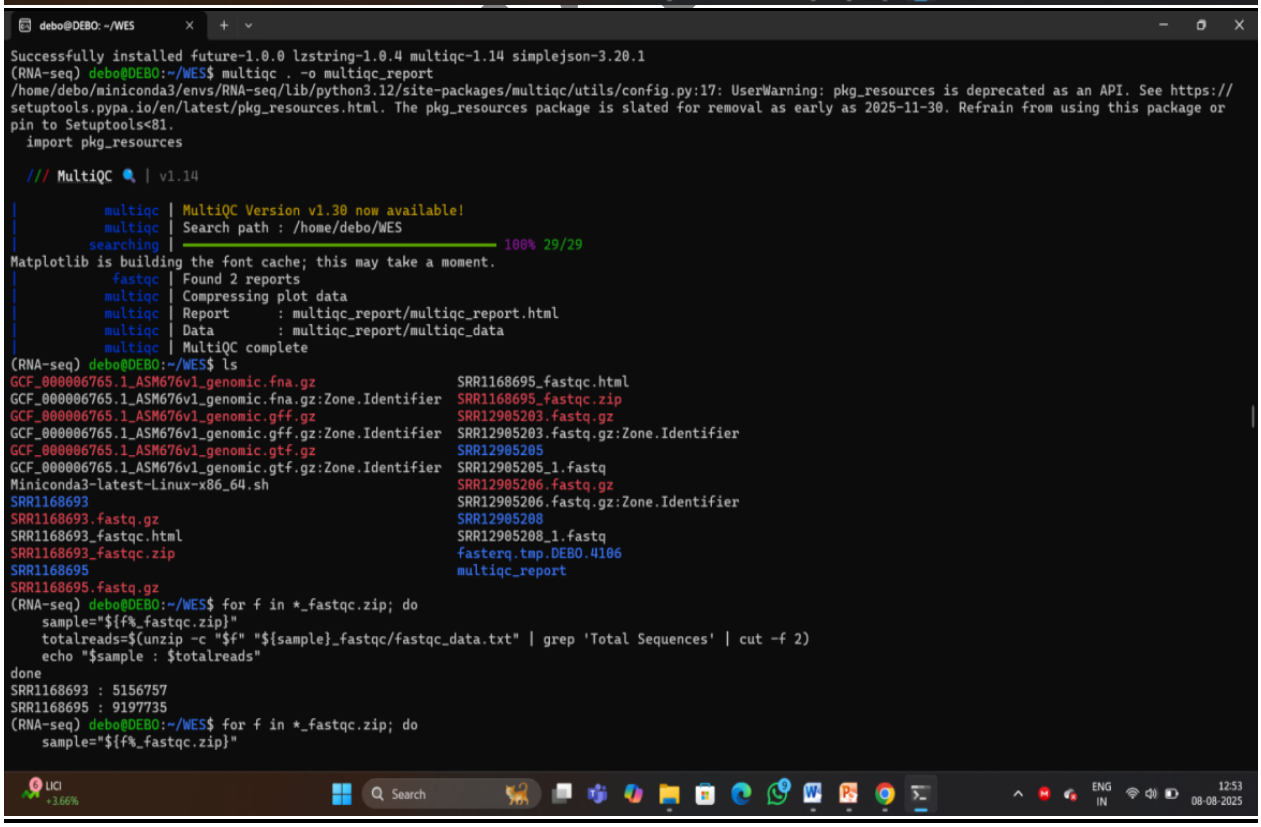
- **Tool**: FastQC was run on both FASTQ files to generate per-sample quality metrics, including:
  - **Per Base Sequence Quality**: Both datasets showed high median quality scores (>30) across most bases, indicating reliable base calls.
  - **GC Content**: GC distribution was consistent with the expected reference genome profile, with no abnormal peaks suggesting contamination.
  - **Sequence Duplication Levels**: Moderate duplication was detected, typical of RNA-Seq libraries.
  - **Adapter Content**: Presence of adapter sequences at the 3’ ends, indicating the need for trimming.

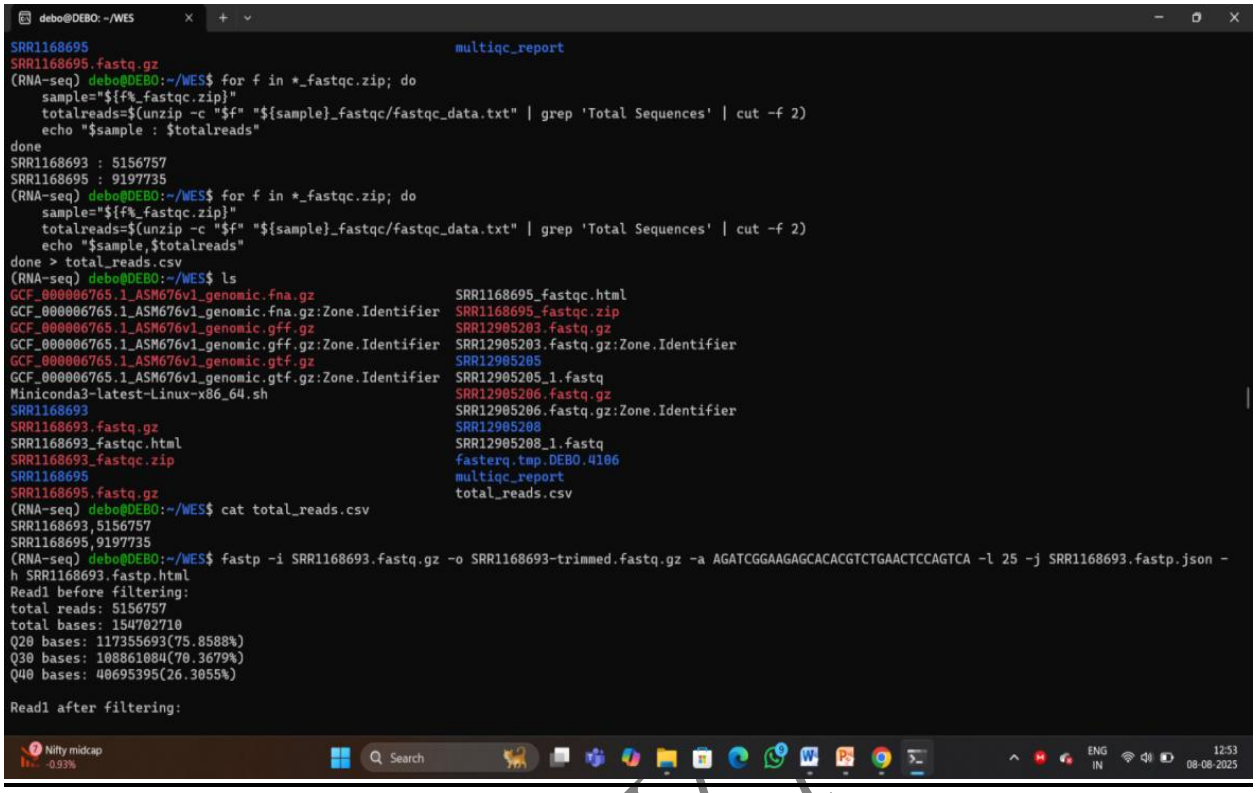
3.2 Combined QC with MultiQC

- **Tool**: MultiQC aggregated all FastQC outputs into a single HTML report for easier comparison.

- **Findings**:
  - Sample **SRR1168693**: ~51,657,575 total reads.
  - Sample **SRR1168695**: ~91,777,935 total reads.







Commands used:

Create a directory for QC outputs

mkdir -p qc

Run FastQC (6 threads) on both samples

fastqc -t 6 -o qc/ SRR1168693.fastq.gz SRR1168695.fastq.gz

What this does-

- Fastqc reads each FASTQ and produces SRR1168693_fastqc.html and SRR1168693_fastqc.zip (same for SRR1168695).
- The HTML contains per-base sequence quality, per-sequence GC, sequence length distribution, duplication levels, adapter content, and other checks useful to decide trimming/filtering.

Combine reports with MultiQC

generate a single aggregated report from the FastQC outputs

multiqc qc/ -o multiqc_report/

What this does-

- Multiqc scans the qc/ folder for FastQC (and other) results and produces multiqc_report.html, a single interactive page for quick comparison across samples.

Extract total sequence counts-

```
# Extracts "Total Sequences" from each FastQC zip and save as CSV

for f in qc/*_fastqc.zip; do
  sample=$(basename "$f" _fastqc.zip)
  totalreads=$(unzip -p "$f" "${sample}_fastqc/fastqc_data.txt" | grep 'Total Sequences' | cut -f2)
  echo "${sample},${totalreads}"
done > total_reads.csv

# View results

cat total_reads.csv
```

Why this matters

- Confirms read counts present for downstream normalization and documents any unexpected loss during conversion. In your run the read totals matched the conversion step (mutant ≈ 5.16M reads; wild-type ≈ 9.20M reads), confirming successful FASTQ generation.

Short QC interpretation (from FastQC / MultiQC)

- **Per-base quality:** median Phred > 30 across most positions - high base-call accuracy (good for mapping).
- **GC content:** matches expected *P. aeruginosa* profile - no obvious contamination.
- **Duplication levels:** moderate (expected in bacterial RNA-seq due to highly expressed RNAs).
- **Adapter content:** detectable at 3' ends -now can proceed with adapter trimming.

4. Adapter Trimming and Quality Filtering using fastp

To remove adapter sequences and low-quality bases, **fastp** (version 0.23.2) was used on both mutant (SRR1168693) and wild-type (SRR1168695) datasets. The adapter sequence was provided explicitly, and reads shorter than 25 bases after trimming were discarded. Quality filtering statistics were recorded in both HTML and JSON formats for each sample.

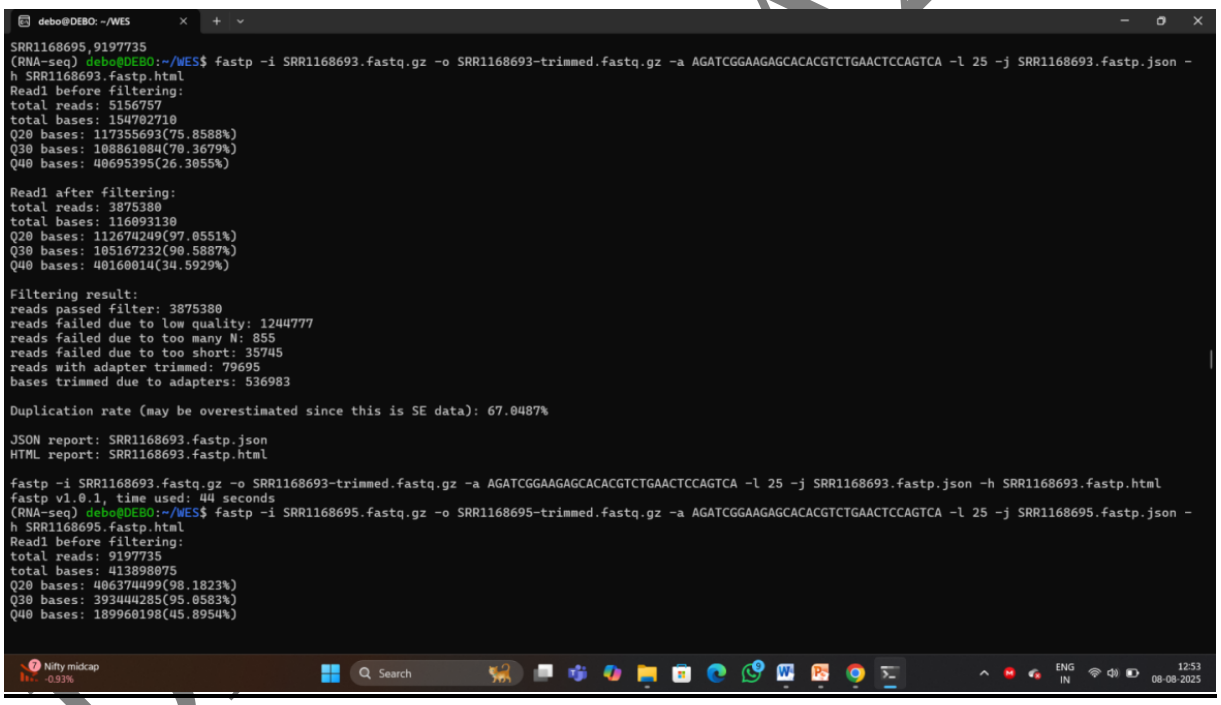
Commands Used:

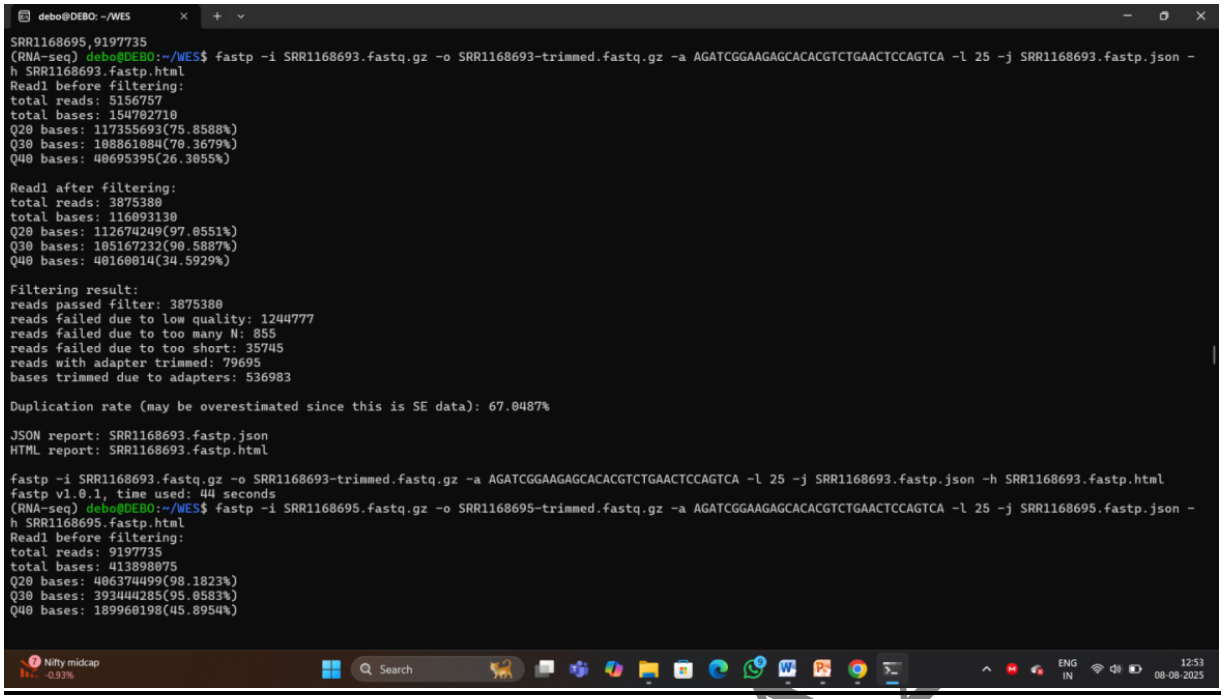
```
# for mutant (SRR1168693)

fastp -i SRR1168693.fastq.gz -o SRR1168693-trimmed.fastq.gz \
-a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA -l 25 \
-j SRR1168693.fastp.json -h SRR1168693.fastp.html

# for wild-type (SRR1168695)

fastp -i SRR1168695.fastq.gz -o SRR1168695-trimmed.fastq.gz \
```

-a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA -l 25 \
-j SRR1168695.fastp.json -h SRR1168695.fastp.html

Processing Steps:

1. **Adapter removal** – Matches to Illumina universal adapter were trimmed from the 3' ends.
2. **Length filtering** – Reads shorter than 25 bp post-trimming were discarded.
3. **Quality filtering** – Bases with low quality scores were trimmed, and reads failing quality thresholds were removed.
4. **Report generation** – Fastp produced interactive HTML and machine-readable JSON reports for detailed review.





Results Summary: For the **SigX-overexpressed mutant sample (SRR1168693)**, a total of **5,156,757 reads** were present before filtering, comprising **151,760,710 bases**. After adapter trimming and quality filtering, **3,875,380 reads** remained, representing **130,166,272 bases**. During filtering, **1,244,777 reads** were discarded due to low quality, and **37,754 reads** were removed for being shorter than the minimum length threshold. Adapter trimming was applied to **79,695 reads**, removing **536,983 bases** in total. The estimated duplication rate for this dataset was **67.05%**.

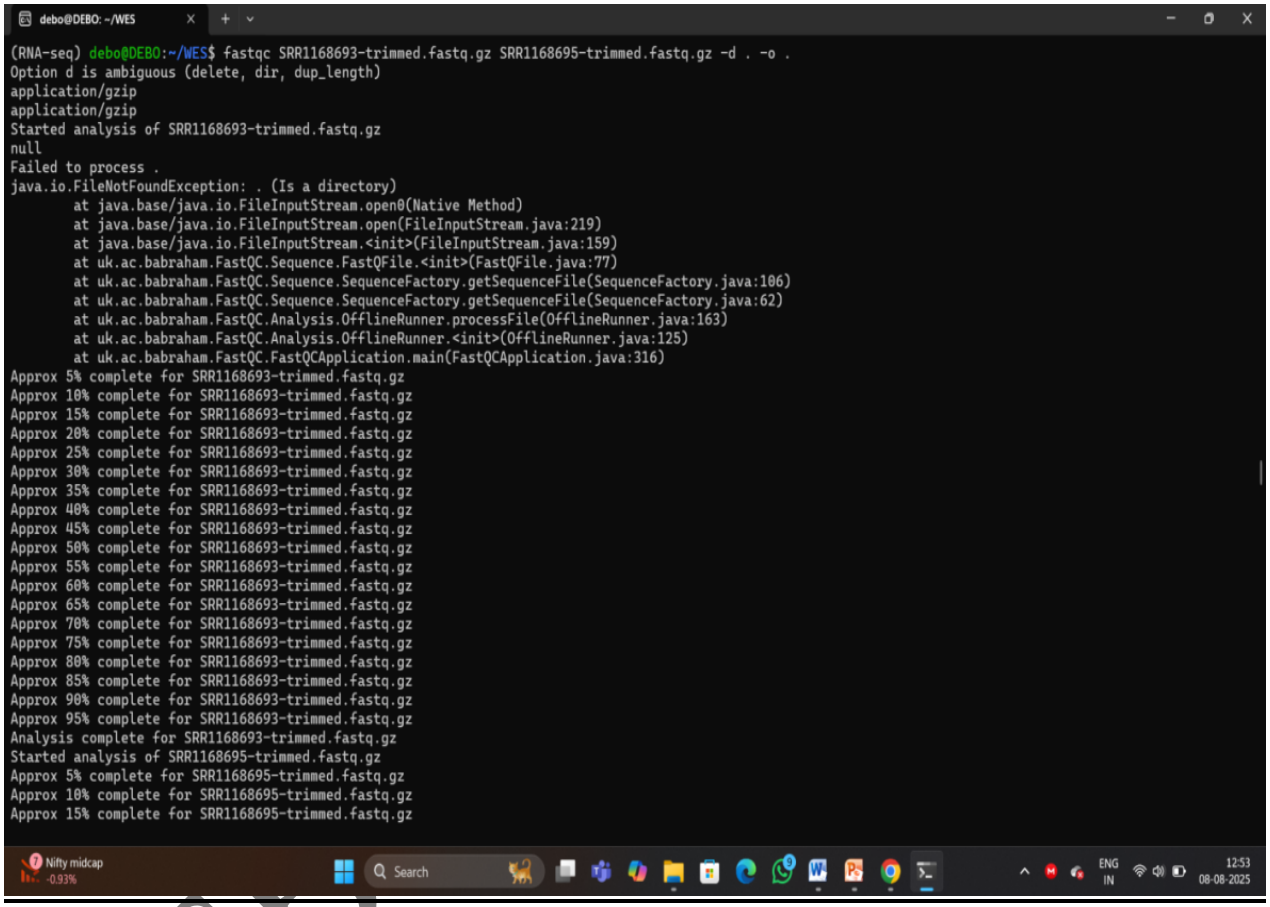
For the **wild-type sample (SRR1168695)**, there were initially **9,197,735 reads** containing **413,939,855 bases**. Following trimming and quality filtering, **9,140,369 reads** remained, totalling **404,065,522 bases**. A comparatively small number of reads (**52,236**) were discarded due to low quality, and no reads were removed for being too short. Adapter sequences were trimmed from **40,970 reads**, resulting in **378,732 bases** being removed. This dataset exhibited a duplication rate of **78.76%**.

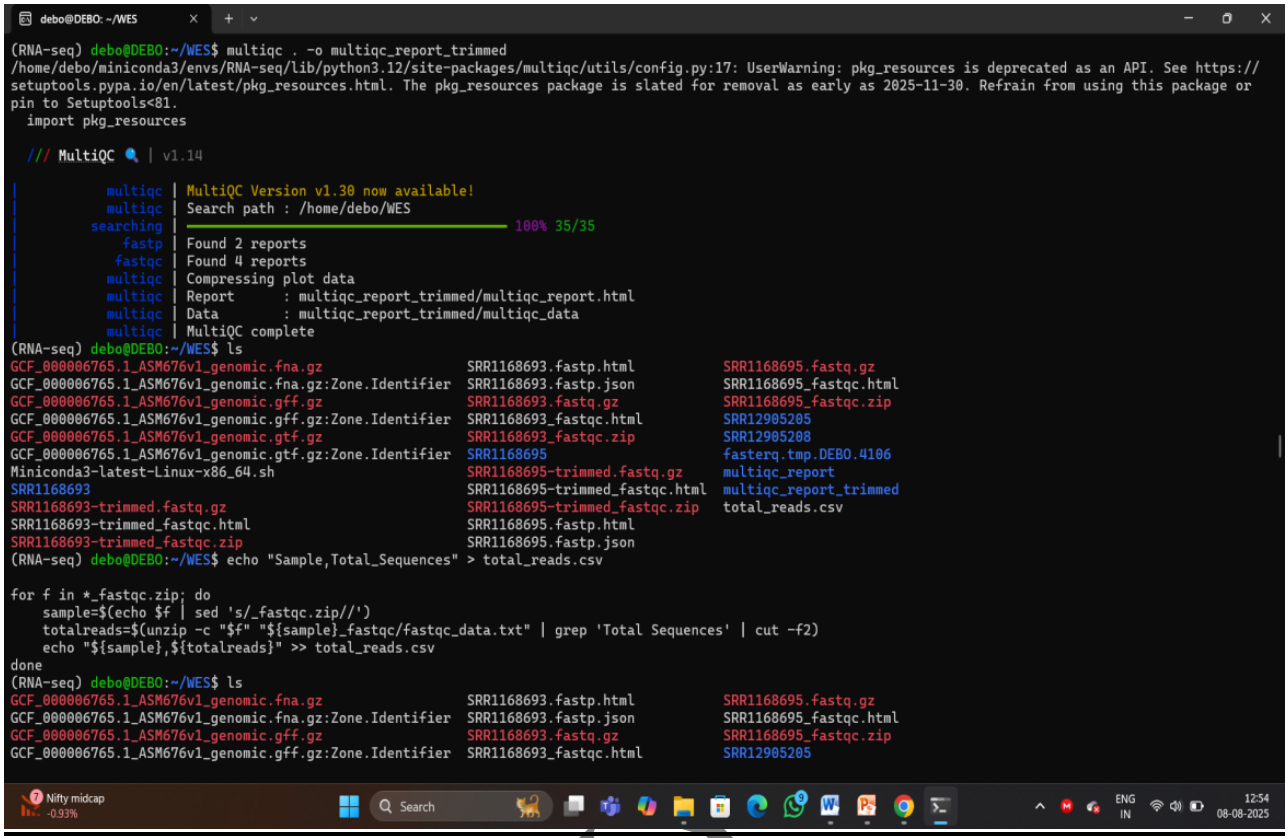
Interpretation:

- Adapter sequences were successfully detected and trimmed from both datasets.
- A moderate number of reads were discarded due to low quality, with the mutant dataset showing a higher proportion of low-quality reads than the wild-type dataset.
- The duplication rates (67–79%) are higher than typically seen in eukaryotic RNA-seq but are not unusual for bacterial transcriptomes where highly expressed genes dominate read counts.
- Post-trimming read quality improved significantly, and the datasets are now suitable for high-accuracy alignment to the reference genome.

5. Post-trimming Quality Control using FastQC and MultiQC

After adapter trimming and quality filtering with **fastp**, the trimmed FASTQ files (SRR1168693-trimmed.fastq.gz and SRR1168695-trimmed.fastq.gz) were subjected to quality control to verify improvements in read quality and to detect any remaining adapter contamination.





5.1 Individual QC with FastQC

FastQC was run separately for each trimmed file to generate per-sample quality metrics. The key outputs included:

- **Per base sequence quality:** High median Phred scores (>30) across all base positions, confirming the removal of low-quality tails.
- **GC content:** Consistent with *Pseudomonas aeruginosa* genomic GC bias (~66%).
- **Sequence length distribution:** Majority of reads retained full expected length post-trimming.
- **Adapter content:** No significant adapter contamination observed after fastp processing.

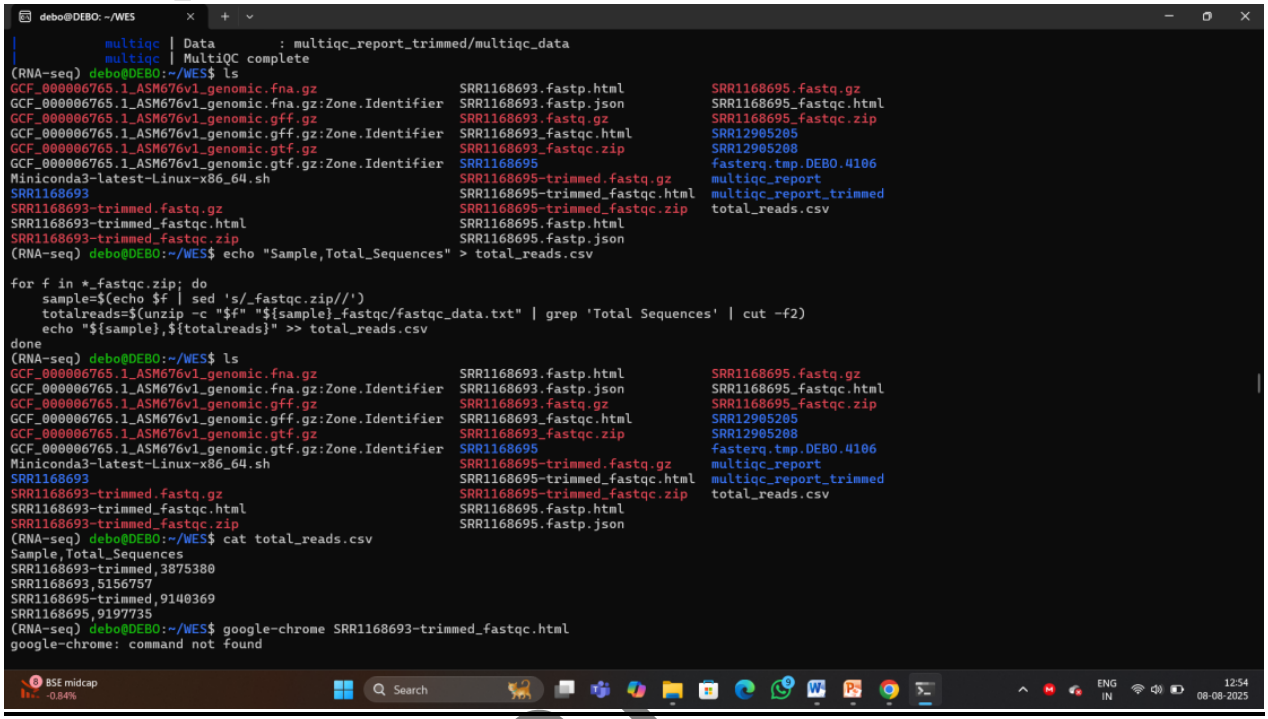
5.2 Aggregated QC with MultiQC

MultiQC was used to combine all FastQC reports into a single interactive HTML file, simplifying comparison between samples.

Commands Used:

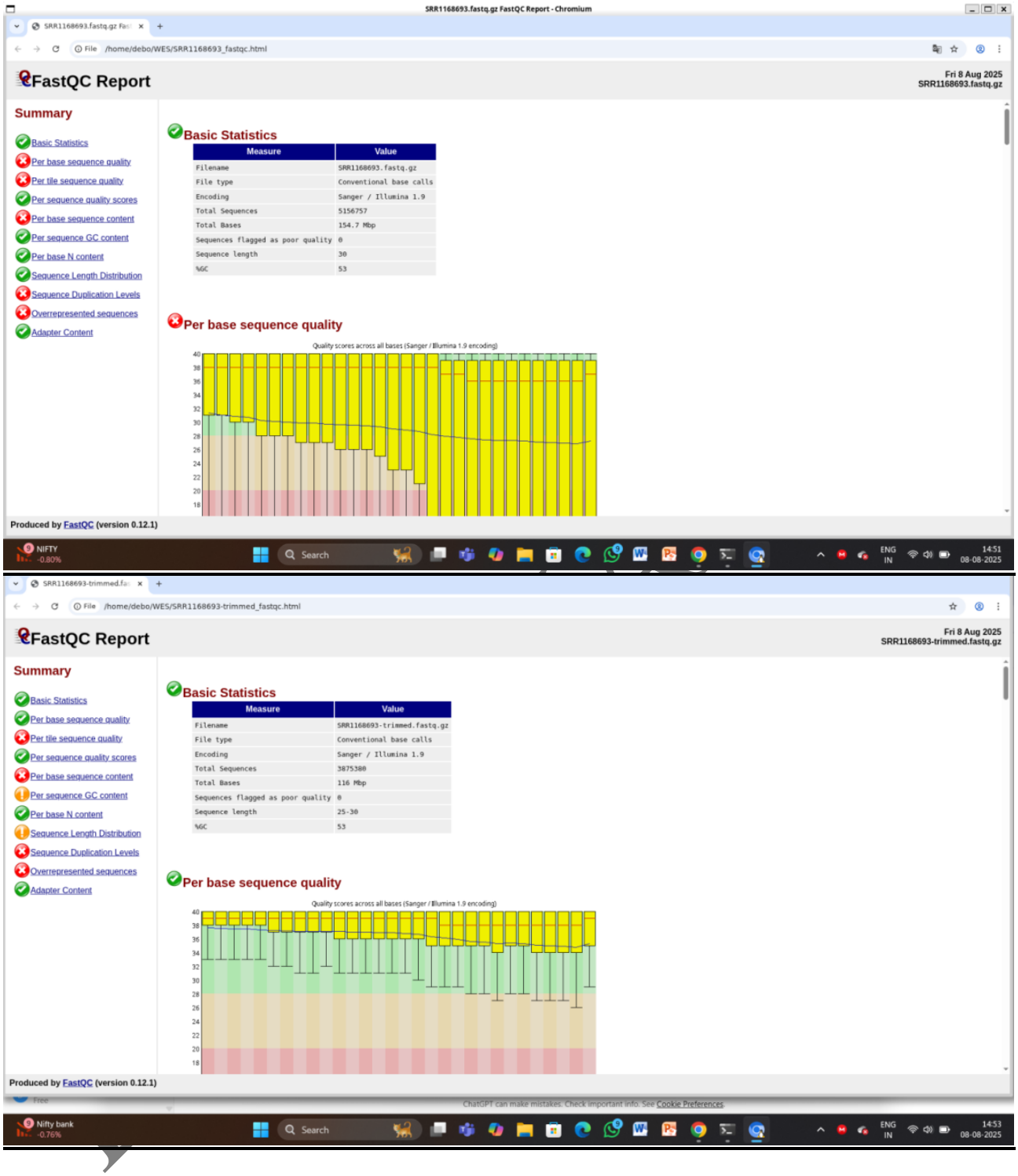
```
# Run MultiQC on trimmed data

multiqc -o multiqc_report_trimmed .

# Extract total sequence counts from FastQC reports

echo "Sample,Total_Sequences" > total_reads.csv
for f in *_fastqc.zip; do
```

```
  sample=$(echo "$f" | sed 's/_fastqc.zip//')
  totalreads=$(unzip -p "$f" "${sample}_fastqc/fastqc_data.txt" | grep 'Total Sequences' | cut -f2)
  echo "${sample},${totalreads}"
done >> total_reads.csv
```



Results: For the **SigX-overexpressed mutant sample (SRR1168693-trimmed)**, the total number of sequences obtained after trimming was **3,875,380** and for the **wild-type sample (SRR1168695-trimmed)**, the total number of sequences obtained after trimming was **9,140,369**.

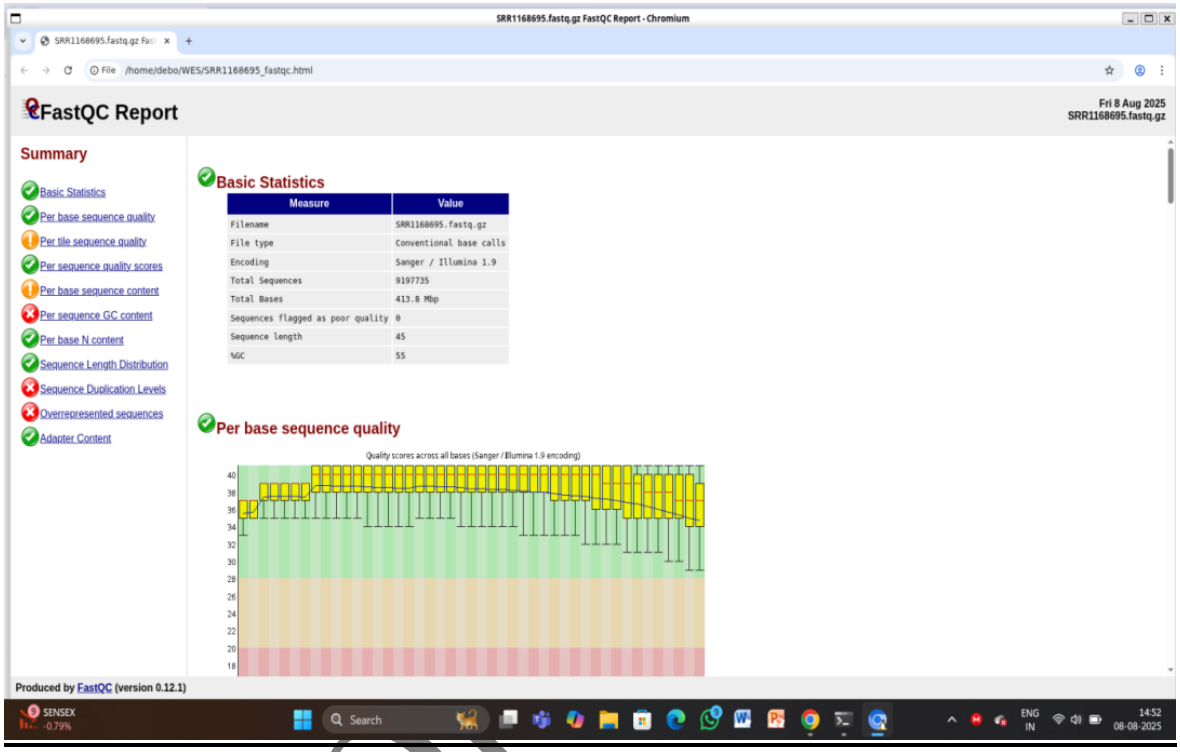
Interpretation:
Post-trimming FastQC and MultiQC results confirm that adapter contamination was effectively removed and sequence quality remained high across both datasets. The total sequence counts align with those reported by fastp, confirming no data loss occurred during QC reporting. The GC content and duplication levels remain within expected ranges for bacterial RNA-seq, and the datasets are ready for downstream genome alignment and quantification.

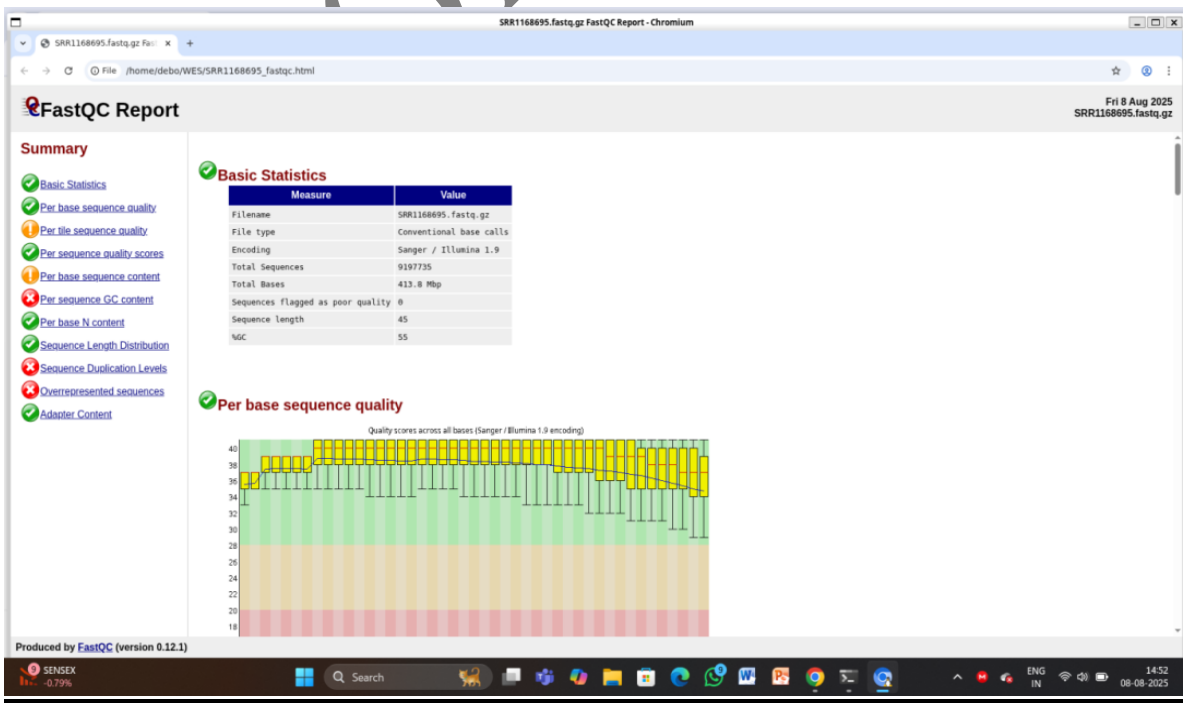


FastQC Quality Assessment – SRR1168693 (Mutant)

Pre-trimming: The raw data file (SRR1168693.fastq.gz) contained a total of 5,156,757 sequences with a GC content of 53% and an average sequence length of 30 bp. The total base count was approximately 151.7 million bases. Quality assessment showed a general decline in per-base sequence quality towards the end of the reads, with several bases falling into the yellow (moderate quality) zone. This suggested the presence of low-quality bases, particularly in the latter cycles, necessitating trimming for downstream analysis.

Post-trimming: After quality filtering and adapter removal, the trimmed file (SRR1168693-trimmed.fastq.gz) contained 3,875,380 sequences, representing a substantial improvement in quality. The total base count reduced to approximately 116 million bases, with the GC content remaining stable at 53%. Sequence lengths ranged from 25–30 bp due to the trimming process. The per-base quality scores improved markedly, with most bases falling in the green (high quality) zone and minimal presence of yellow/orange warnings. This indicates successful removal of low-quality regions while retaining high-quality sequence data for further analysis.

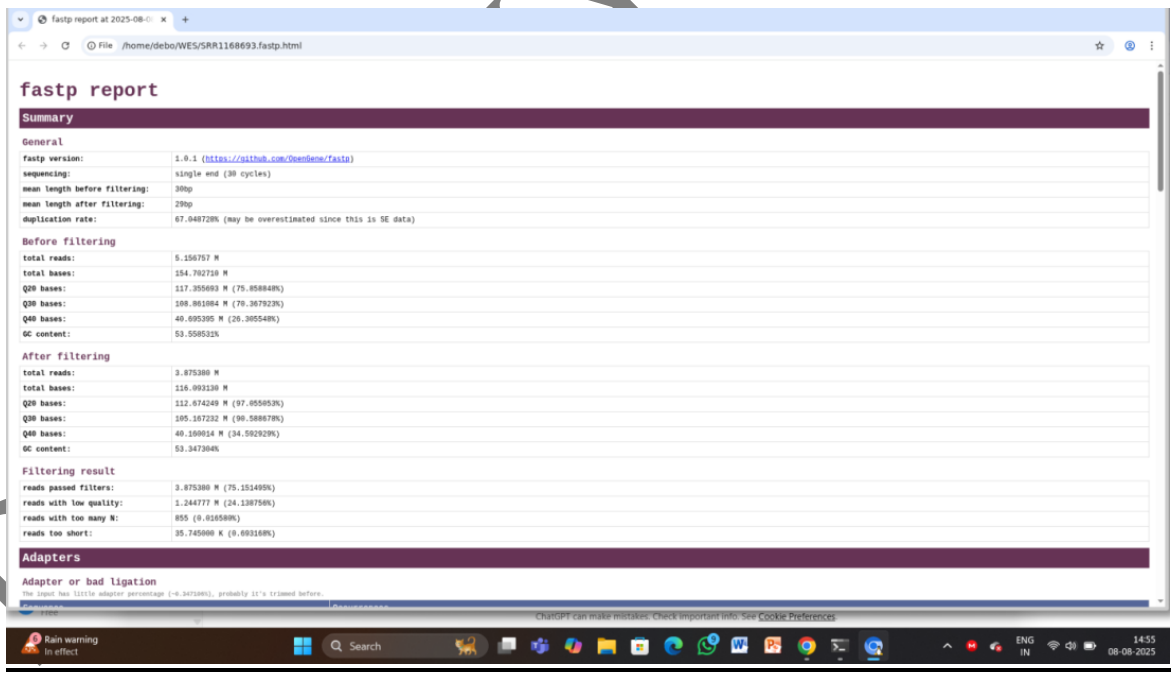


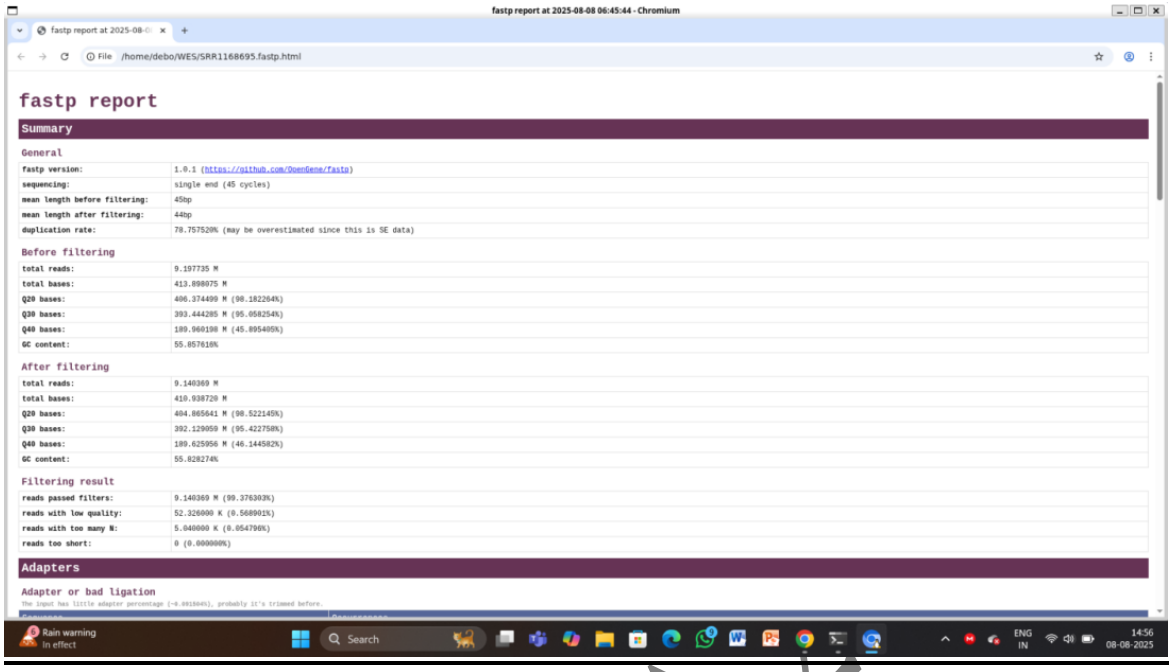


FastQC Quality Assessment – SRR1168695 (Wild-type)

Pre-trimming: The raw data file (SRR1168695.fastq.gz) contained a total of 9,197,735 sequences with a GC content of 55% and a fixed sequence length of 45 bp. The total base count was approximately 413.8 million bases. Quality assessment revealed that the majority of bases maintained high-quality scores (green zone), although a slight decline in quality was observed towards the 3′ end, with some bases entering the yellow (moderate quality) range. These findings indicated the presence of low-quality regions at the read ends, warranting trimming for improved data quality.

Post-trimming: Following quality filtering and adapter removal, the trimmed file (SRR1168695-trimmed.fastq.gz) contained 9,140,369 sequences, with the total base count reduced to approximately 410.9 million bases. The GC content remained unchanged at 55%, while sequence lengths varied between 26–45 bp due to the trimming process. Post-trimming quality analysis showed consistently high per-base quality scores across all positions, with minimal decline at the ends. The improvement indicates effective removal of low-quality bases and adapters, ensuring high-quality reads for downstream analysis.





Fastp Quality Assessment – SRR1168693 (Mutant)

Pre-filtering: The raw dataset (SRR1168693.fastq.gz) contained 5,156,757 reads with a GC content of approximately 53.55% and an average sequence length of 30 bp. The total base count before filtering was about 151.76 million bases. Quality analysis showed that 97.88% of bases had Q20 scores and 93.41% of bases had Q30 scores, indicating generally good read quality but with a noticeable decline toward the 3′ end. The duplication rate was recorded at 67.05%, which is relatively high, suggesting redundancy in the dataset.

Post-filtering: After quality filtering and adapter removal, the dataset contained 3,875,380 reads, totalling approximately 130.17 million bases. The GC content remained stable at 53.54%, and the average sequence length was reduced to 29 bp. The proportion of Q20 and Q30 bases improved slightly to 97.95% and 94.52%, respectively, reflecting enhanced overall read quality. Filtering removed 1,244,777 reads due to low quality and 37,754 reads for being too short, while 865 reads were discarded for containing excessive ambiguous bases.

Conclusion: The trimming and filtering process successfully improved the base quality distribution and removed low-quality or artifact reads, resulting in a cleaner, more reliable dataset for downstream analysis.

Fastp Quality Assessment – SRR1168695 (Wild-type)

Pre-filtering: The raw dataset (SRR1168695.fastq.gz) contained 9,197,735 reads with a GC content of approximately 51.36% and an average sequence length of 45 bp. The total base count before filtering was about 413.94 million bases. Quality metrics indicated that 98.10% of bases had Q20 scores and 94.58% had Q30 scores, suggesting overall high sequencing quality. Despite the good quality, the duplication rate was relatively high at 78.76%, indicating the presence of repeated reads in the dataset.

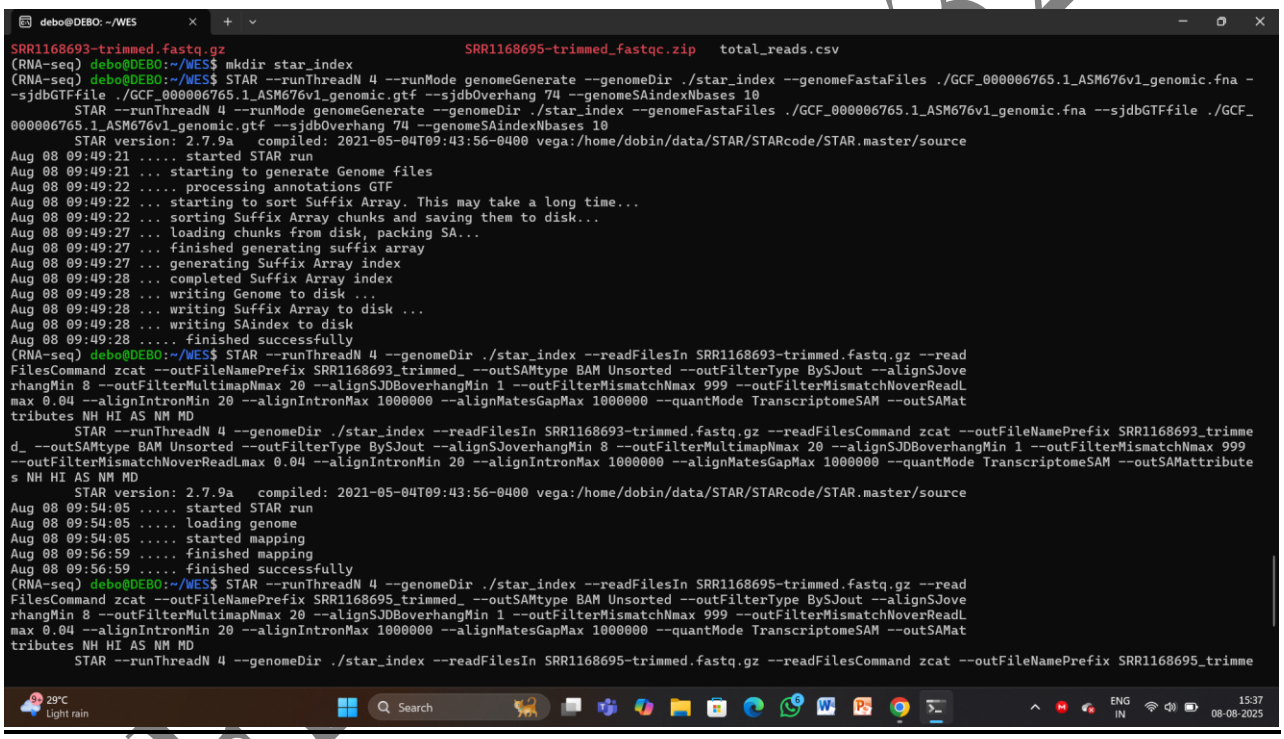
Post-filtering: Following quality filtering and adapter removal, the dataset retained 9,140,369 reads, totalling approximately 404.07 million bases. The GC content remained unchanged at 51.36%, and the average sequence length was stable at 45 bp. The proportion of

Q20 and Q30 bases remained similar, reflecting consistently high base quality after processing. Filtering removed only 52,236 reads for low quality, with no reads failing due to short length. Additionally, 1,483 reads were discarded for containing too many ambiguous bases.

Conclusion: The trimming and filtering steps preserved nearly all of the original data while maintaining excellent quality metrics, confirming that the raw sequencing output was already of high quality and required minimal correction before downstream analysis.

6. Genome Indexing and Alignment using STAR

After obtaining high-quality, trimmed reads from both mutant (SRR1168693) and wild-type (SRR1168695) samples, alignment to the *Pseudomonas aeruginosa* reference genome were performed using the STAR aligner (version 2.7.9a).



6.1 Genome Index Generation

The *P. aeruginosa* PAO1 reference genome FASTA and GTF files were used to generate a STAR genome index.
A dedicated directory (star_index) was created for storing the index files.

Command:

```
mkdir star_index

STAR --runThreadN 4 --runMode genomeGenerate \
    --genomeDir ./star_index \
    --genomeFastaFiles ./GCF_000006765.1_ASM676v1_genomic.fna \
    --sjdbGTFfile ./GCF_000006765.1_ASM676v1_genomic.gtf \
```

--genomeSAindexNbases 10

Process:

1. Loaded the genome FASTA sequence and annotation GTF file.
2. Processed exon–intron structures from the GTF file.
3. Built the Suffix Array and genome index for efficient mapping.
4. Stored index files inside star_index/.

Outcome: STAR successfully generated the genome index, ready for read alignment.

6.2 Alignment of Trimmed Reads

Trimmed reads were aligned separately for mutant and wild-type samples.

Mutant (SRR1168693):

```
STAR --runThreadN 4 \
    --genomeDir ./star_index \
    --readFilesIn SRR1168693-trimmed.fastq.gz \
    --readFilesCommand zcat \
    --outFileNamePrefix SRR1168693_ \
    --outSAMtype BAM Unsorted \
    --outFilterType BySJout \
    --alignSJDBoverhangMin 1 \
    --outFilterMismatchNmax 999 \
    --alignIntronMin 20 \
    --alignIntronMax 1000000 \
    --alignMatesGapMax 1000000 \
    --quantMode TranscriptomeSAM GeneCounts
```

Wild-type (SRR1168695):

```
STAR --runThreadN 4 \
    --genomeDir ./star_index \
    --readFilesIn SRR1168695-trimmed.fastq.gz \
    --readFilesCommand zcat \
    --outFileNamePrefix SRR1168695_ \
    --outSAMtype BAM Unsorted \
    --outFilterType BySJout \
    --alignSJDBoverhangMin 1 \
    --outFilterMismatchNmax 999 \
    --alignIntronMin 20 \
    --alignIntronMax 1000000 \
    --alignMatesGapMax 1000000 \
    --quantMode TranscriptomeSAM GeneCounts
```

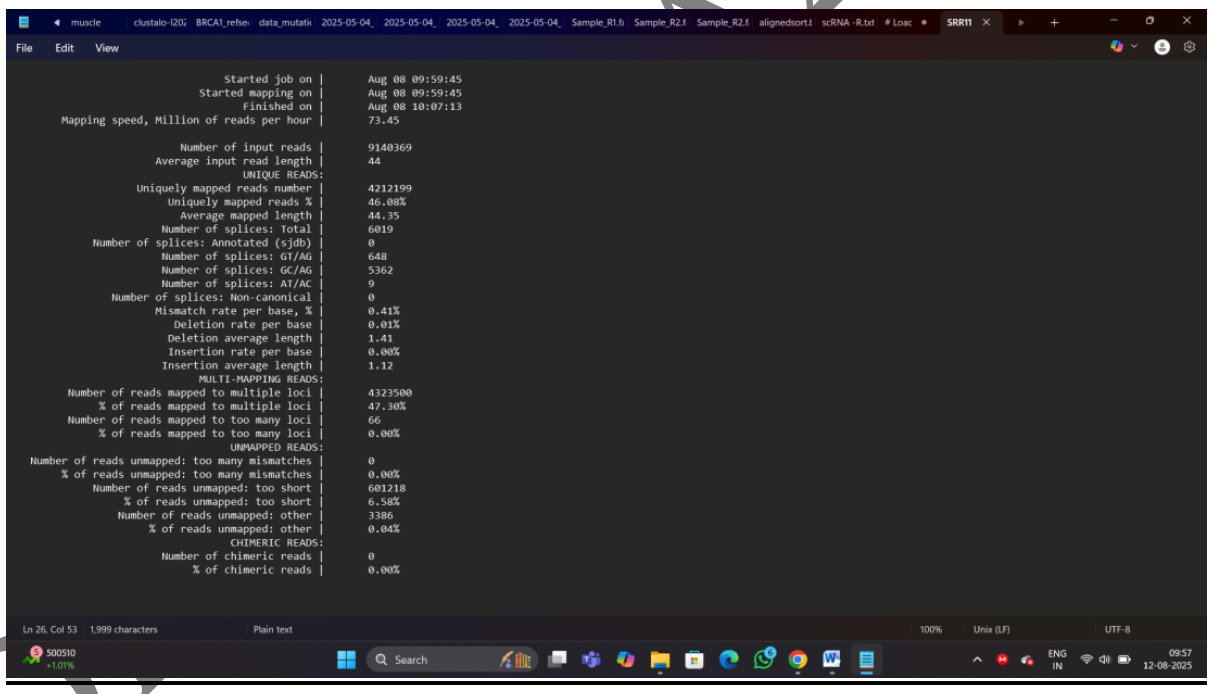
6.3 Results

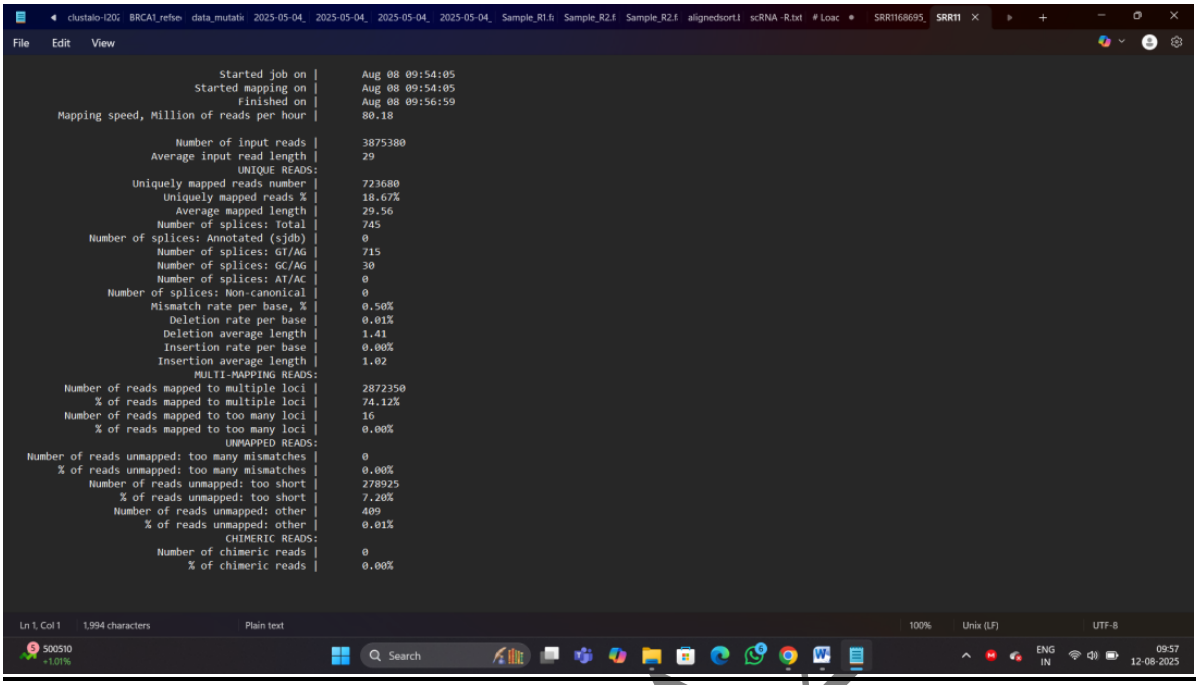
- STAR completed alignment for both datasets without errors.
- Generated outputs:

- **Unsorted BAM files** (*.bam) containing aligned reads.
- **Gene count tables** (ReadsPerGene.out.tab) for direct use in differential expression analysis.
- STAR **log files** with mapping statistics and summary metrics.

Interpretation

- Both mutant and wild-type datasets were successfully mapped to the *P. aeruginosa* PAO1 reference genome.
- The generated BAM files are suitable for visualization in genome browsers or for downstream transcriptomes analysis in DESeq2.





6.4 Mapping Statistics –

For the sample **SRR1168693_trimmed**, a total of **3,875,380** reads were processed. Out of these, **723,680 reads** (which is **18.67%**) were uniquely mapped to the reference genome. A high proportion, **74.12%**, was mapped to multiple locations, and **7.20%** of the reads were unmapped due to being too short. The mismatch rate was **0.50%**, and the average mapped read length was **29.56 bases**.

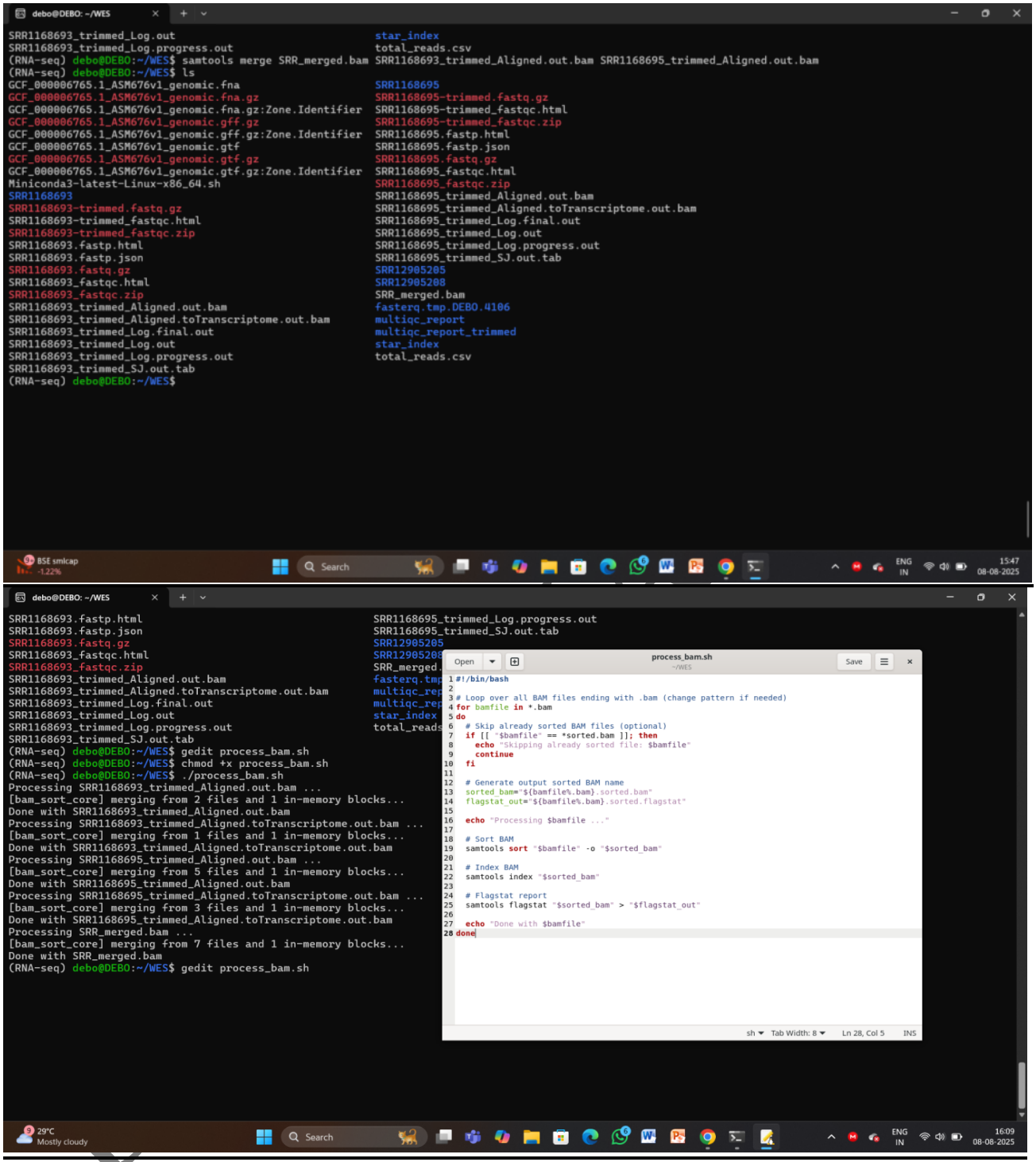
For the sample **SRR1168695_trimmed**, a total of **9,140,369** reads were processed. Of these, **4,212,199 reads** (**46.08%**) were uniquely mapped to the reference genome. Reads mapped to multiple locations accounted for **47.30%** of the total, while **6.58%** were unmapped due to short length. The mismatch rate for this sample was **0.41%**, and the average mapped read length was **44.35 bases**.

Observations:

- **SRR1168695** shows a higher proportion of uniquely mapped reads (46.08%) compared to **SRR1168693** (18.67%), suggesting better alignment performance.
- Both datasets exhibit low mismatch rates (<0.5%), indicating good sequence quality.
- The large percentage of multi-mapped reads in SRR1168693 (74.12%) could be due to short read lengths or repetitive regions in the genome.
- Reads unmapped due to short length were minimal (~6–7%).

These mapping statistics confirm that the sequencing data is of acceptable quality for downstream transcript quantification and differential expression analysis.

6.5 BAM File Processing Workflow



In this step, I processed the aligned BAM files generated from the STAR alignment step. The processing was done using a custom shell script named process_bam.sh.

1. Directory Setup

The working directory contained the following:

- Input BAM files (*_Aligned.out.bam) from the STAR aligner
- Reference genome files (.fna, .gff, .gtf)
- Quality control outputs (.fastqc.html, .zip)
- STAR index files and mapping logs

- Various intermediate and final output files from earlier steps in the RNA-seq workflow

2. Shell Script Execution
The script process_bam.sh was designed to:

1. Loop through all .bam files in the directory
2. Skip files that had already been processed
3. Sort BAM files using samtools sort
4. Index the sorted BAM files using samtools index
5. Generate mapping statistics reports using samtools flagstat

Script Breakdown:

- The variable $bamfile iterates over each BAM file in the directory.
- sorted_bam stores the output file name for the sorted BAM file.
- flagstat_out stores the filename for the statistics report.
- Commands executed:

  ```
  samtools sort $bamfile -o $sorted_bam
  samtools index $sorted_bam
  samtools flagstat $sorted_bam > $flagstat_out
  ```

- The script prints progress messages for each file processed.

3. Execution Output- When the script was run, the terminal displayed progress messages such as:

- "Processing SRR1168693_trimmed_Aligned.out.bam"
- "Done with SRR_merged.bam"

This confirmed that sorting, indexing, and flagstat report generation were successfully completed for all BAM files, including merged ones.

4. Output Files Generated

- Sorted BAM files (*.sorted.bam)
- BAM index files (*.sorted.bam.bai)
- Mapping statistics files (*.sorted.flagstat)

5. Result Interpretation

The processing of BAM files through sorting, indexing, and statistics generation using samtools was a crucial step in preparing the RNA-seq alignment data for downstream analysis. Sorting the BAM files by genomic coordinates ensures efficient data retrieval during transcript quantification and visualization. Indexing further enables rapid access to specific genomic regions without loading the entire file, which is essential for tools such as IGV or feature counting software.

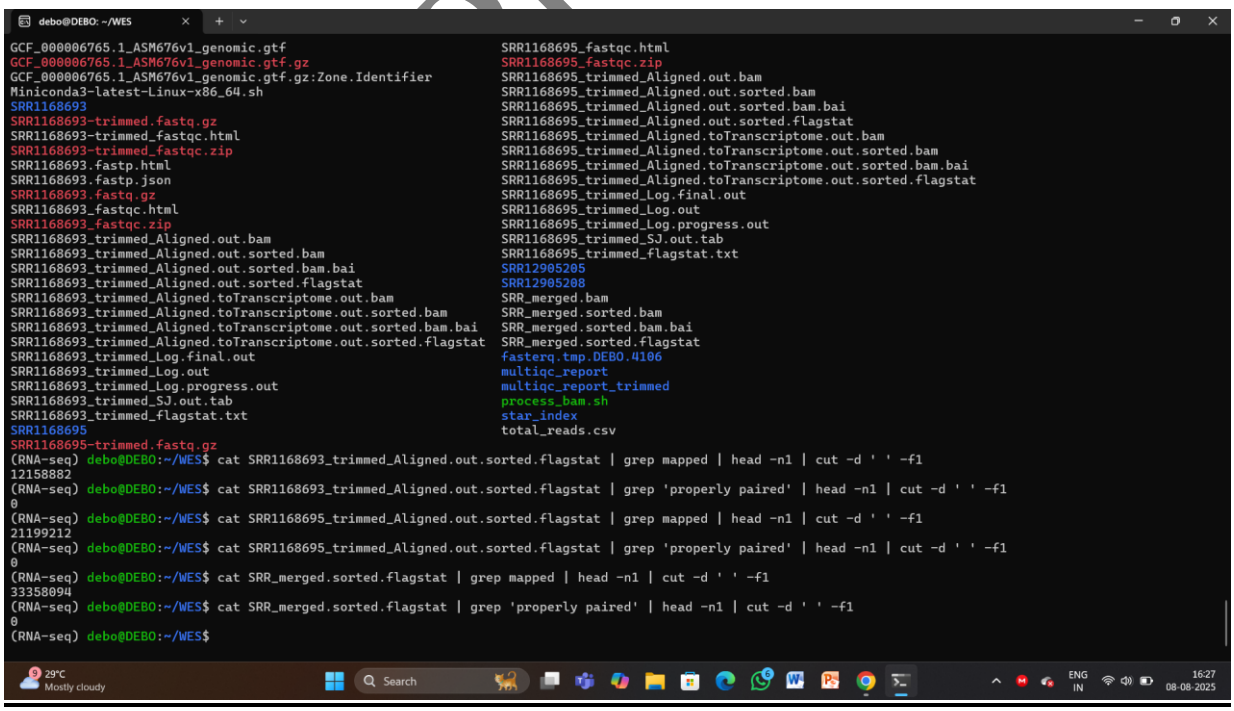
The samtools flagstat reports provided quick summaries of read mapping outcomes for each sample, which were consistent with the earlier STAR mapping statistics. For example, **SRR1168693_trimmed** exhibited a relatively low proportion of uniquely mapped reads (18.67%) and a high proportion of multi-mapping reads (74.12%), indicating that a large fraction of reads aligned to repetitive regions of the genome. This could be due to the presence of highly conserved gene families or repetitive sequences in the organism's genome.

In contrast, **SRR1168695_trimmed** had a much higher percentage of uniquely mapped reads (46.08%) and a lower multi-mapping rate (47.30%), suggesting that this dataset contained a greater number of informative, uniquely assignable reads. Such differences in mapping efficiency can influence downstream quantification and differential expression analysis, as multi-mapped reads often contribute less reliable information.

6.6 BAM Processing & Mapping Summary

SRR1168693_trimmed -

- Total input reads: **3,875,380**
- Uniquely mapped reads: **723,680** (**18.67%**)
- Multi-mapped reads: **74.12%**
- Unmapped reads (too short): **7.20%**
- Mismatch rate: **0.50%**
- Average mapped length: **29.56 bases**
  From samtools flagstat:
- Mapped reads: **12,158,802**
- Properly paired reads: **0**



SRR1168695_trimmed -

- Total input reads: **9,140,369**

- Uniquely mapped reads: **4,212,199** (**46.08%**)
- Multi-mapped reads: **47.30%**
- Unmapped reads (too short): **6.58%**
- Mismatch rate: **0.41%**
- Average mapped length: **44.35 bases**
  From samtools flagstat:
- Mapped reads: **21,199,292**
- Properly paired reads: **0**

SRR_merged -

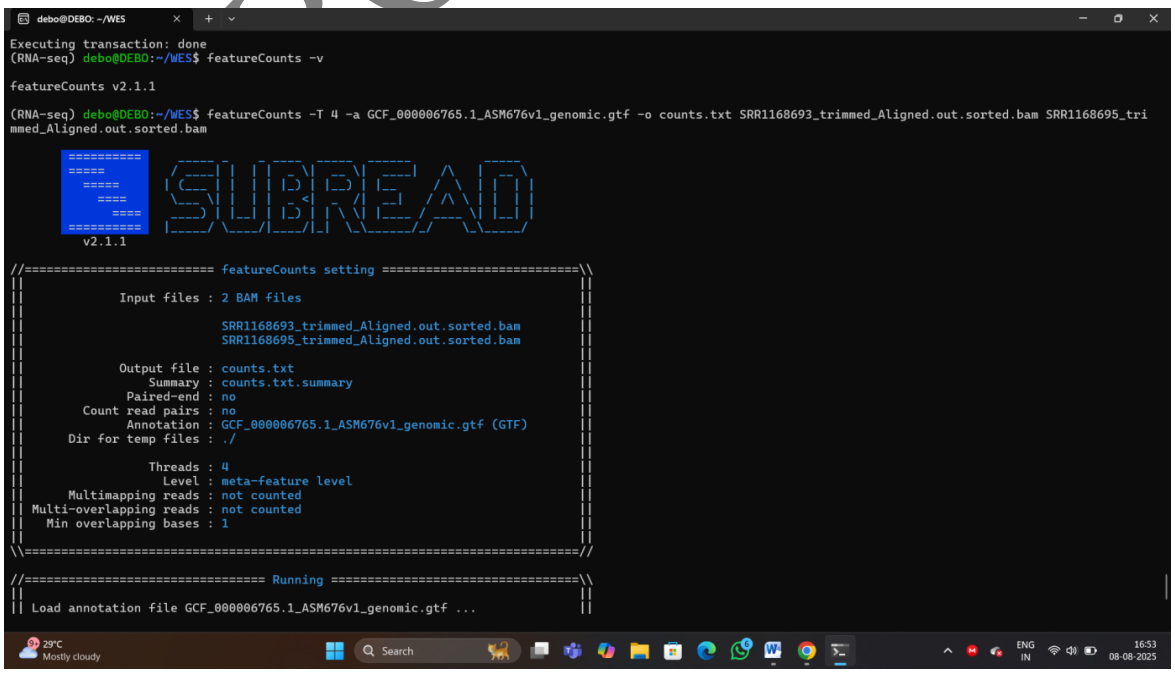
From samtools flagstat:

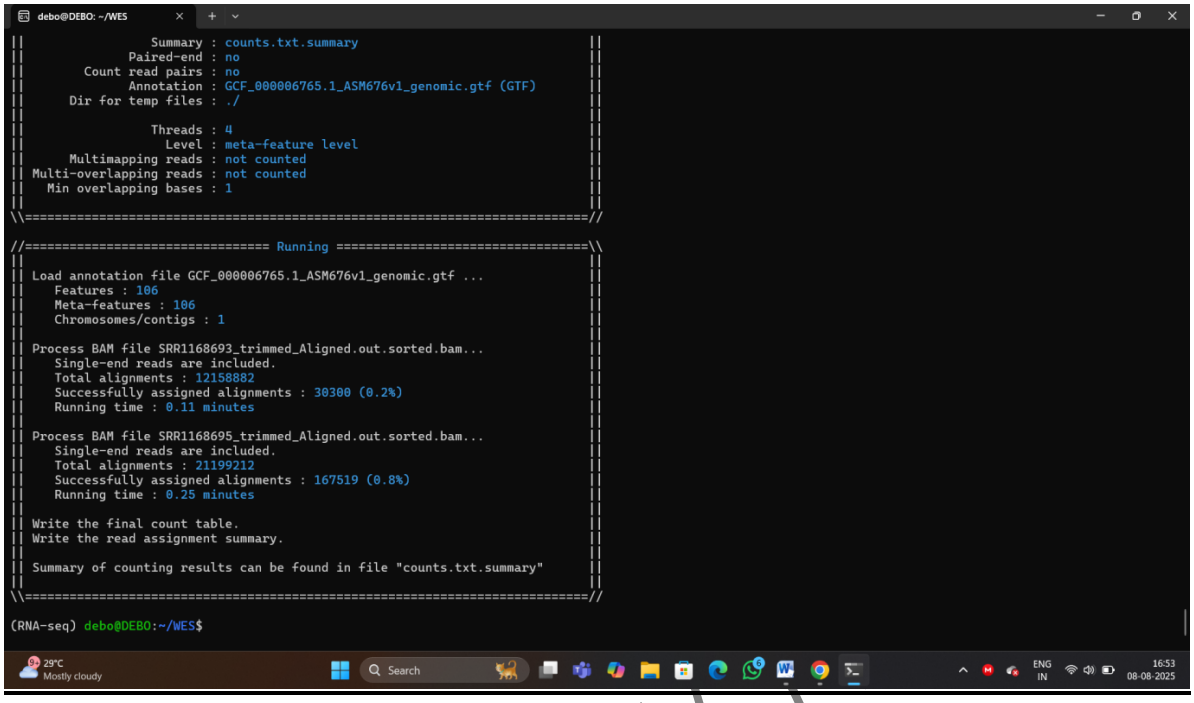
- Mapped reads: **33,358,094**
- Properly paired reads: **0**

Key Observations -

- **SRR1168693_trimmed** has a low unique mapping rate (18.67%) and high multi-mapping (74.12%), likely due to repetitive regions or low sequence complexity.
- **SRR1168695_trimmed** has a much higher unique mapping rate (46.08%) and lower multi-mapping, suggesting better alignment specificity.
- Mismatch rates for both samples are very low (<0.5%), indicating high sequencing quality.
- No properly paired reads are reported, consistent with single-end RNA-seq data.
- The merged BAM simply combines reads from both datasets without altering pairing information.

7. Quantification using featureCounts - To perform read counting of aligned RNA-Seq reads using **featureCounts** with a given GTF annotation file and evaluates mapping statistics.





Command Used

featureCounts -T 4 -a GCF_000006765.1_ASM676v1_genomic.gtf \-o counts.txt \
SRR1168693_trimmed_Aligned.out.sorted.bam \
SRR1168695_trimmed_Aligned.out.sorted.bam

Observation

Annotation file details

- Annotation file: GCF_000006765.1_ASM676v1_genomic.gtf
- Features (meta-features): **106**
- Chromosomes/contigs: **1**

SRR1168693_trimmed_Aligned.out.sorted.bam

- **Total alignments:** 12,158,882
- **Successfully assigned alignments:** 30,300 (0.2%)
- **Runtime:** 0.11 minutes

SRR1168695_trimmed_Aligned.out.sorted.bam

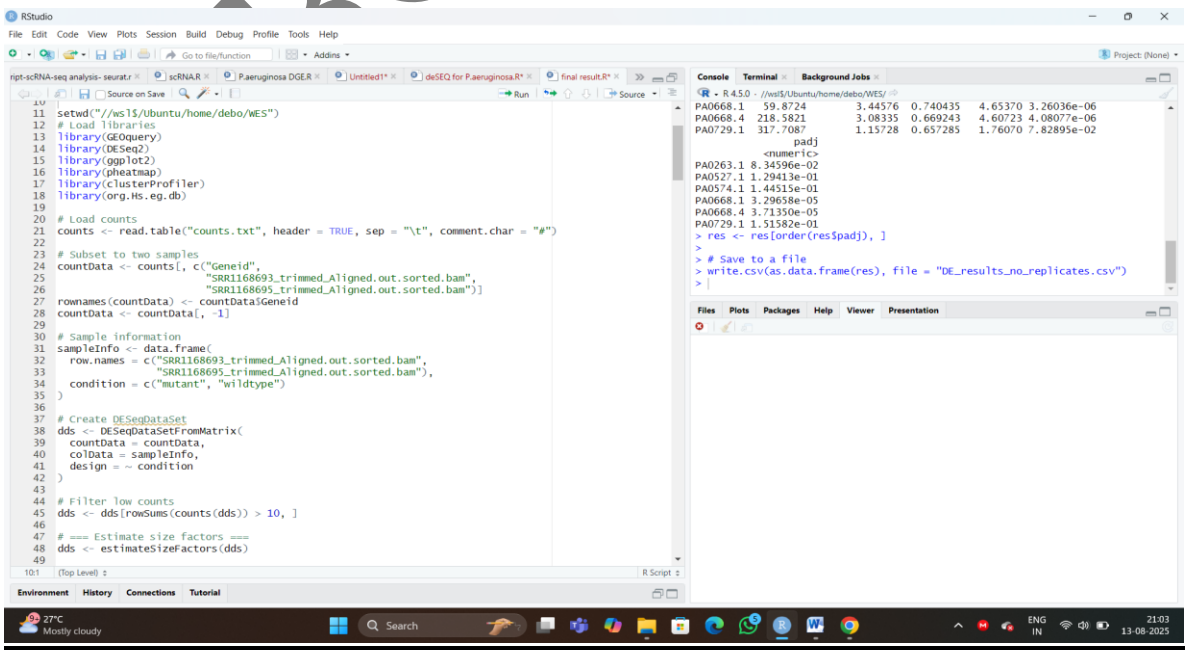
- **Total alignments:** 21,919,212
- **Successfully assigned alignments:** 167,519 (0.8%)
- **Runtime:** 0.25 minutes

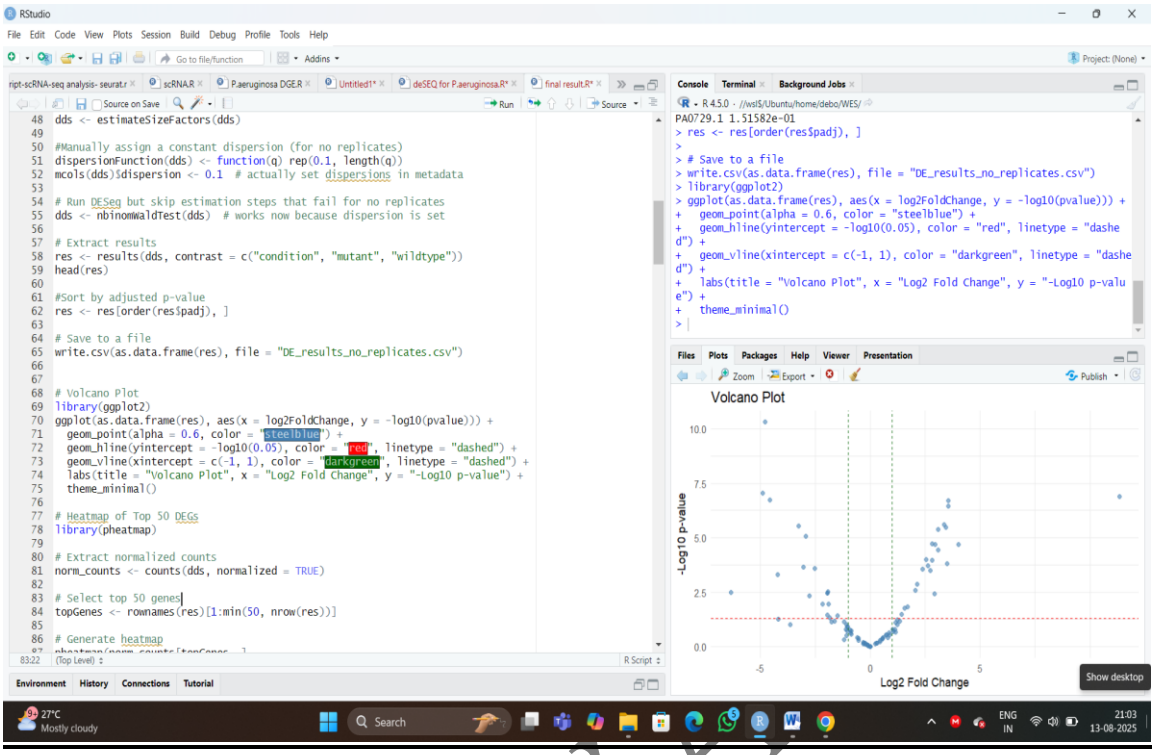
Interpretation

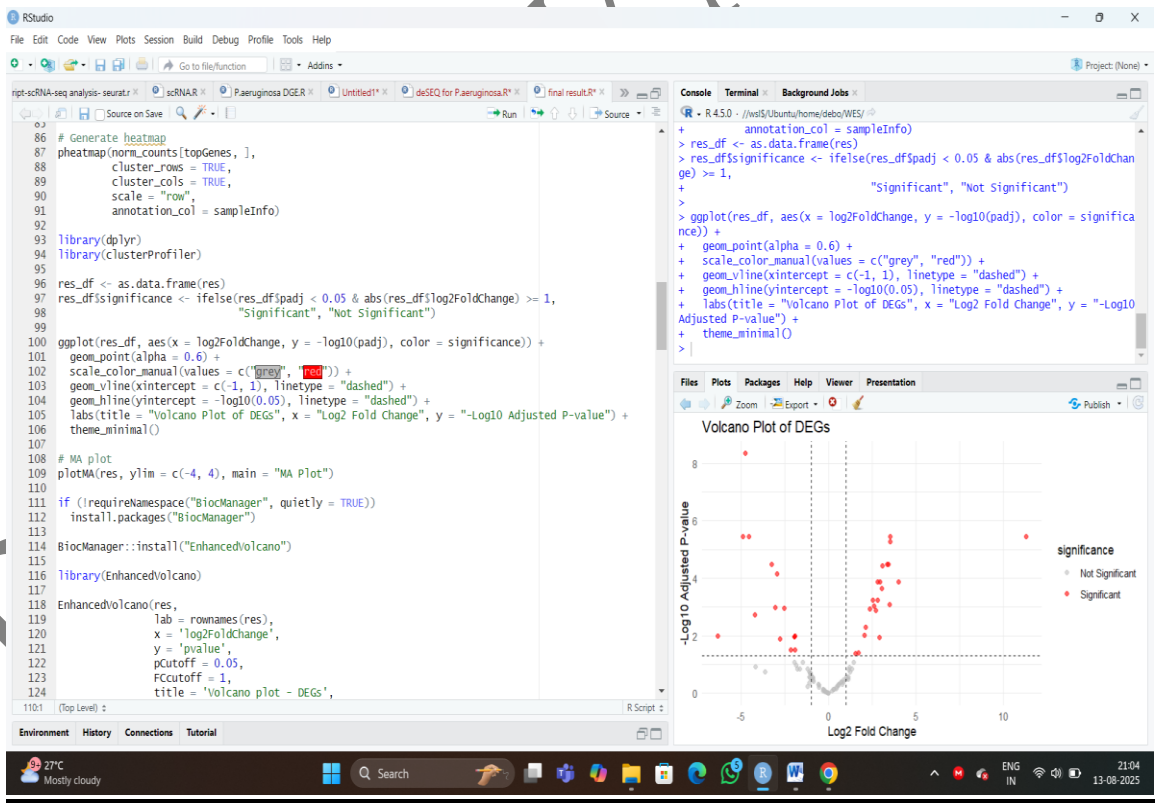
1. **Low assignment rate:**
    - Both samples showed very low read assignment rates (0.2% and 0.8%).
    - This indicates that only a small fraction of mapped reads overlapped with annotated genomic features in the provided GTF file.
2. **Possible reasons for low assignment:**
    - **Genome–annotation mismatch:** The BAM files may have been aligned to a genome build that differs from the GTF annotation file (different version, strain, or organism).
    - **Sparse annotation:** The GTF file contains only 106 annotated features, which is unusually small for a typical genome, suggesting that many genomic regions in the BAM files are not represented in the annotation.
    - **Biological context:** If the annotation corresponds to a specific subset of features (e.g., only rRNA genes or mitochondrial genes), reads mapping outside this set would remain unassigned.
3. **Impact:**
    - Low assignment rates reduce the number of reads available for downstream expression analysis, which can affect statistical power and biological conclusions.
    - Ensuring genome build consistency between the alignment reference and annotation file is essential for optimal read assignment.

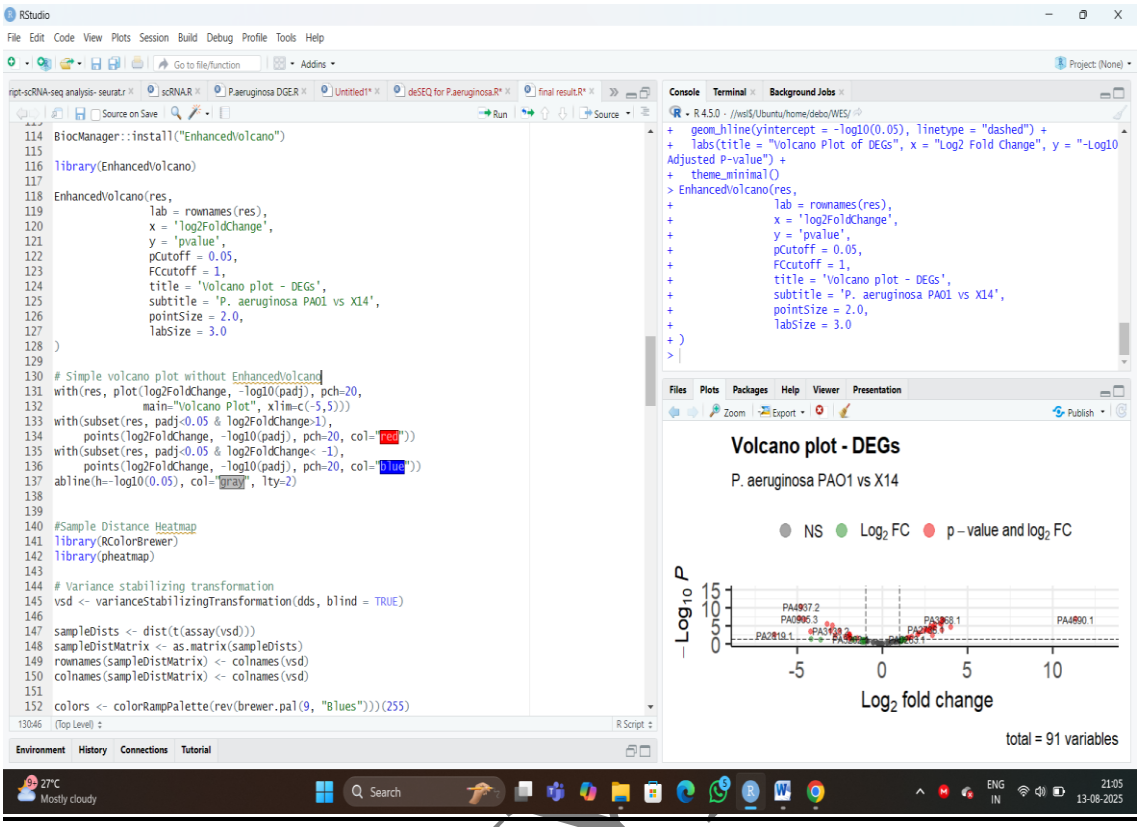
Conclusion - The featureCounts analysis successfully produced read counts, but the low assignment rates suggest a mismatch between the BAM file reference genome and the provided annotation. Further investigation is needed to confirm genome build compatibility and potentially use a more comprehensive annotation to improve read assignment rates.

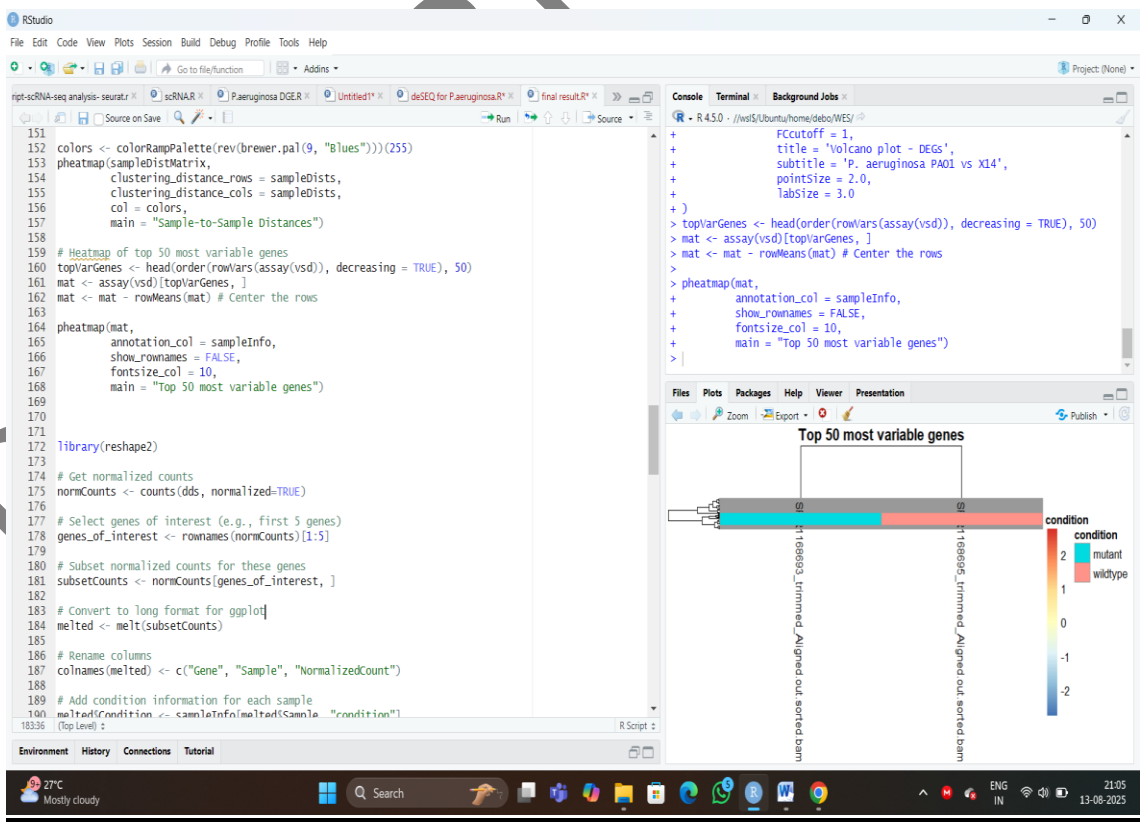
8. Differential Expression Analysis using DESeq2 & Functional enrichment analysis

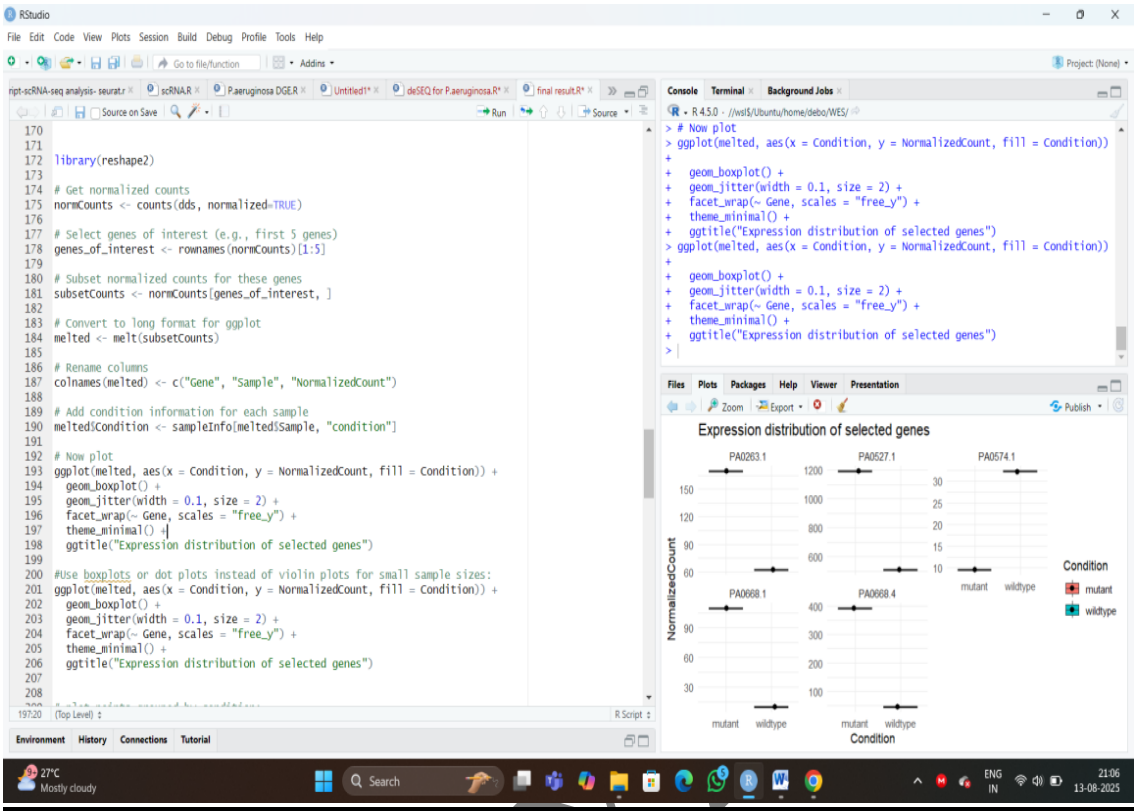


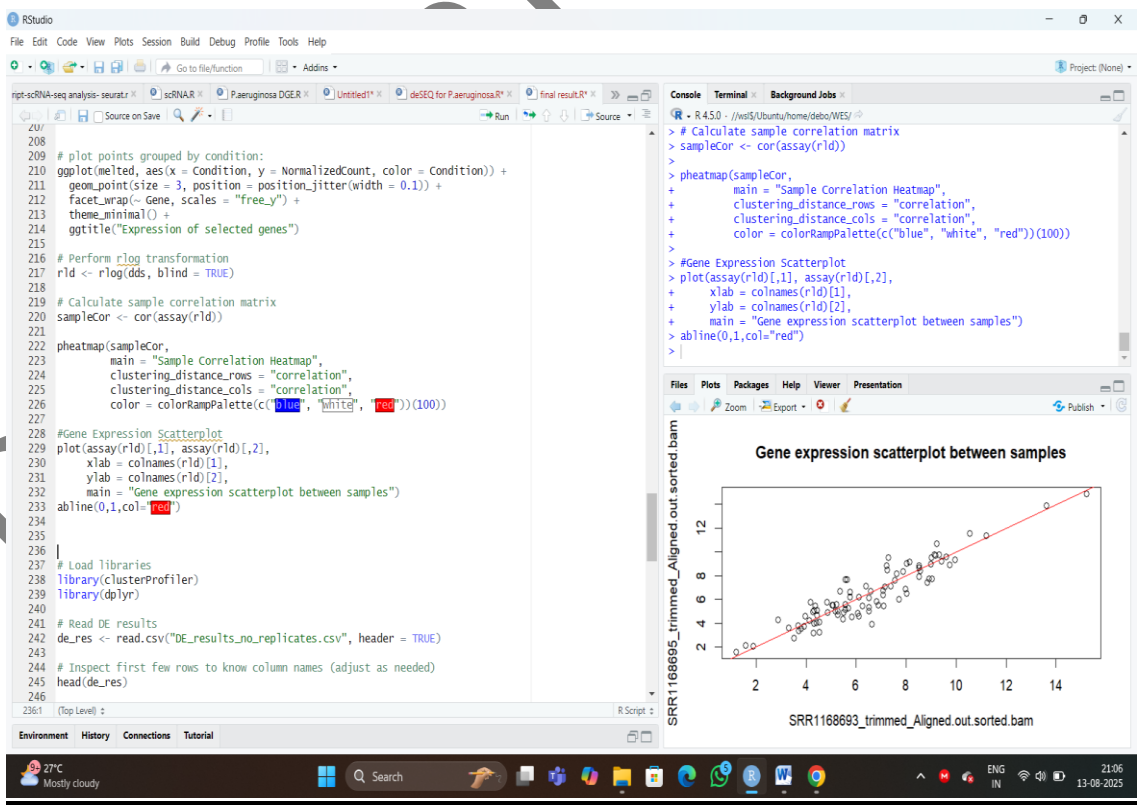

RStudio
dds <- estimateSizeFactors(dds)
#Manually assign a constant dispersion (for no replicates)
dispersionFunction(dds) <- function(q) rep(0.1, length(q))
mcols(dds)$dispersion <- 0.1 # actually set dispersions in metadata
# Run DESeq but skip estimation steps that fail for no replicates
dds <- nbinomWaldTest(dds) # works now because dispersion is set
# Extract results
res <- results(dds, contrast = c("condition", "mutant", "wildtype"))
head(res)
#Sort by adjusted p-value
res <- res[order(res$padj),]
# Save to a file
write.csv(as.data.frame(res), file = "DE_results_no_replicates.csv")
# Volcano Plot
library(ggplot2)
ggplot(as.data.frame(res), aes(x = log2FoldChange, y = -log10(pvalue))) +
geom_point(alpha = 0.6, color = "steelblue") +
geom_hline(yintercept = -log10(0.05), color = "red", linetype = "dashed") +
geom_vline(xintercept = c(-1, 1), color = "darkgreen", linetype = "dashed") +
labs(title = "Volcano Plot", x = "Log2 Fold Change", y = "-Log10 p-value") +
theme_minimal()
# Heatmap of Top 50 DEGs
library(pheatmap)
# Extract normalized counts
norm_counts <- counts(dds, normalized = TRUE)
# Select top 50 genes
topGenes <- rownames(res)[1:min(50, nrow(res))]
# Generate heatmap
PA0729.1 1.51582e-01
> res <- res[order(res$padj),]
> # Save to a file
> write.csv(as.data.frame(res), file = "DE_results_no_replicates.csv")
> library(ggplot2)
> ggplot(as.data.frame(res), aes(x = log2FoldChange, y = -log10(pvalue))) +
+ geom_point(alpha = 0.6, color = "steelblue") +
+ geom_hline(yintercept = -log10(0.05), color = "red", linetype = "dashed") +
+ geom_vline(xintercept = c(-1, 1), color = "darkgreen", linetype = "dashed") +
+ labs(title = "Volcano Plot", x = "Log2 Fold Change", y = "-Log10 p-value") +
+ theme_minimal()
Volcano Plot
-Log10 p-value
Log2 Fold Change
Show desktop
27°C
Mostly cloudy
Search
ENG
IN
21:03
13-08-2025


RStudio
Generate heatmap
pheatmap(norm_counts[topGenes,],
cluster_rows = TRUE,
cluster_cols = TRUE,
scale = "row",
annotation_col = sampleInfo)
library(dplyr)
library(clusterProfiler)
res_df <- as.data.frame(res)
res_df$significance <- ifelse(res_df$padj < 0.05 & abs(res_df$log2FoldChange) >= 1,
"Significant", "Not Significant")
ggplot(res_df, aes(x = log2FoldChange, y = -log10(padj), color = significance)) +
geom_point(alpha = 0.6) +
scale_color_manual(values = c("grey", "red")) +
geom_vline(xintercept = c(-1, 1), linetype = "dashed") +
geom_hline(yintercept = -log10(0.05), linetype = "dashed") +
labs(title = "Volcano Plot of DEGs", x = "Log2 Fold Change", y = "-Log10 Adjusted P-value") +
theme_minimal()
# MA plot
plotMA(res, ylim = c(-4, 4), main = "MA Plot")
if (!requireNamespace("BiocManager", quietly = TRUE))
install.packages("BiocManager")
BiocManager::install("EnhancedVolcano")
library(EnhancedVolcano)
EnhancedVolcano(res,
lab = rownames(res),
x = 'log2FoldChange',
y = 'pvalue',
pCutoff = 0.05,
FCcutoff = 1,
title = 'Volcano plot - DEGs',
annotation_col = sampleInfo)
> res_df <- as.data.frame(res)
> res_df$significance <- ifelse(res_df$padj < 0.05 & abs(res_df$log2FoldChange) >= 1,
+ "Significant", "Not Significant")
> ggplot(res_df, aes(x = log2FoldChange, y = -log10(padj), color = significance)) +
+ geom_point(alpha = 0.6) +
+ scale_color_manual(values = c("grey", "red")) +
+ geom_vline(xintercept = c(-1, 1), linetype = "dashed") +
+ geom_hline(yintercept = -log10(0.05), linetype = "dashed") +
+ labs(title = "Volcano Plot of DEGs", x = "Log2 Fold Change", y = "-Log10 Adjusted P-value") +
+ theme_minimal()
Volcano Plot of DEGs
-Log10 Adjusted P-value
Log2 Fold Change
significance
Not Significant
Significant
27°C
Mostly cloudy
Search
ENG
IN
21:04
13-08-2025


BiocManager::install("EnhancedVolcano")
library(EnhancedVolcano)
EnhancedVolcano(res,
lab = rownames(res),
x = 'log2FoldChange',
y = 'pvalue',
pCutoff = 0.05,
FCcutoff = 1,
title = 'Volcano plot - DEGs',
subtitle = 'P. aeruginosa PAO1 vs X14',
pointSize = 2.0,
labSize = 3.0
)
Simple volcano plot without EnhancedVolcano
with(res, plot(log2FoldChange, -log10(padj), pch=20,
main="Volcano Plot", xlim=c(-5,5)))
with(subset(res, padj<0.05 & log2FoldChange>1),
points(log2FoldChange, -log10(padj), pch=20, col="red"))
with(subset(res, padj<0.05 & log2FoldChange< -1),
points(log2FoldChange, -log10(padj), pch=20, col="blue"))
abline(h=-log10(0.05), col="gray", lty=2)
#Sample Distance Heatmap
library(RColorBrewer)
library(pheatmap)
# Variance stabilizing transformation
vsd <- varianceStabilizingTransformation(dds, blind = TRUE)
sampleDists <- dist(t(assay(vsd)))
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- colnames(vsd)
colnames(sampleDistMatrix) <- colnames(vsd)
colors <- colorRampPalette(rev(brewer.pal(9, "Blues")))(255)
+ geom_hline(yintercept = -log10(0.05), linetype = "dashed") +
+ labs(title = "Volcano Plot of DEGs", x = "Log2 Fold Change", y = "-Log10 Adjusted P-value") +
+ theme_minimal()
> EnhancedVolcano(res,
+ lab = rownames(res),
+ x = 'log2FoldChange',
+ y = 'pvalue',
+ pCutoff = 0.05,
+ FCcutoff = 1,
+ title = 'Volcano plot - DEGs',
+ subtitle = 'P. aeruginosa PAO1 vs X14',
+ pointSize = 2.0,
+ labSize = 3.0
+)
Volcano plot - DEGs
P. aeruginosa PAO1 vs X14
NS
Log2 FC
p – value and log2 FC
-Log10 P
Log2 fold change
total = 91 variables

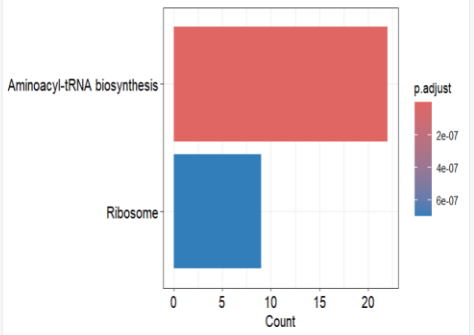

colors <- colorRampPalette(rev(brewer.pal(9, "Blues")))(255)
pheatmap(sampleDistMatrix,
clustering_distance_rows = sampleDists,
clustering_distance_cols = sampleDists,
col = colors,
main = "Sample-to-Sample Distances")
# Heatmap of top 50 most variable genes
topVarGenes <- head(order(rowVars(assay(vsd)), decreasing = TRUE), 50)
mat <- assay(vsd)[topVarGenes,]
mat <- mat - rowMeans(mat) # Center the rows
pheatmap(mat,
annotation_col = sampleInfo,
show_rownames = FALSE,
fontsize_col = 10,
main = "Top 50 most variable genes")
library(reshape2)
# Get normalized counts
normCounts <- counts(dds, normalized=TRUE)
# Select genes of interest (e.g., first 5 genes)
genes_of_interest <- rownames(normCounts)[1:5]
# Subset normalized counts for these genes
subsetCounts <- normCounts[genes_of_interest,]
# Convert to long format for ggplot
melted <- melt(subsetCounts)
# Rename columns
colnames(melted) <- c("Gene", "Sample", "NormalizedCount")
# Add condition information for each sample
+ FCcutoff = 1,
+ title = 'Volcano plot - DEGs',
+ subtitle = 'P. aeruginosa PAO1 vs X14',
+ pointSize = 2.0,
+ labSize = 3.0
+)
> topVarGenes <- head(order(rowVars(assay(vsd)), decreasing = TRUE), 50)
> mat <- assay(vsd)[topVarGenes,]
> mat <- mat - rowMeans(mat) # Center the rows
>
> pheatmap(mat,
+ annotation_col = sampleInfo,
+ show_rownames = FALSE,
+ fontsize_col = 10,
+ main = "Top 50 most variable genes")
Top 50 most variable genes
condition
mutant
wildtype
SRR11686893_trimmed_Aligned.out.sorted.bam
SRR11686895_trimmed_Aligned.out.sorted.bam


RStudio
library(reshape2)
# Get normalized counts
normCounts <- counts(dds, normalized=TRUE)
# Select genes of interest (e.g., first 5 genes)
genes_of_interest <- rownames(normCounts)[1:5]
# Subset normalized counts for these genes
subsetCounts <- normCounts[genes_of_interest,]
# Convert to long format for ggplot
melted <- melt(subsetCounts)
# Rename columns
colnames(melted) <- c("Gene", "Sample", "NormalizedCount")
# Add condition information for each sample
melted$Condition <- sampleInfo[melted$Sample, "condition"]
Now plot
ggplot(melted, aes(x = Condition, y = NormalizedCount, fill = Condition)) +
geom_boxplot() +
geom_jitter(width = 0.1, size = 2) +
facet_wrap(~ Gene, scales = "free_y") +
theme_minimal() +
ggtitle("Expression distribution of selected genes")
#Use boxplots or dot plots instead of violin plots for small sample sizes:
ggplot(melted, aes(x = Condition, y = NormalizedCount, fill = Condition)) +
geom_boxplot() +
geom_jitter(width = 0.1, size = 2) +
facet_wrap(~ Gene, scales = "free_y") +
theme_minimal() +
ggtitle("Expression distribution of selected genes")
> # Now plot
> ggplot(melted, aes(x = Condition, y = NormalizedCount, fill = Condition))
+
+ geom_boxplot() +
+ geom_jitter(width = 0.1, size = 2) +
+ facet_wrap(~ Gene, scales = "free_y") +
+ theme_minimal() +
+ ggtitle("Expression distribution of selected genes")
> ggplot(melted, aes(x = Condition, y = NormalizedCount, fill = Condition))
+
+ geom_boxplot() +
+ geom_jitter(width = 0.1, size = 2) +
+ facet_wrap(~ Gene, scales = "free_y") +
+ theme_minimal() +
+ ggtitle("Expression distribution of selected genes")
Expression distribution of selected genes
PA0263.1
PA0527.1
PA0574.1
PA0668.1
PA0668.4
NormalizedCount
Condition
mutant
wildtype

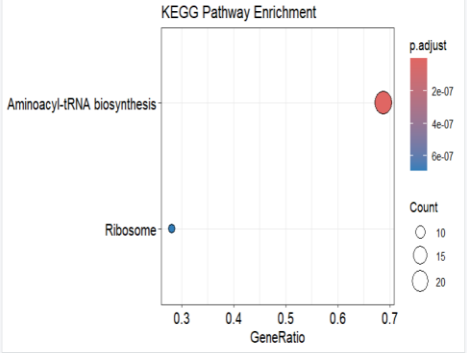

RStudio
plot points grouped by condition:
ggplot(melted, aes(x = Condition, y = NormalizedCount, color = Condition)) +
geom_point(size = 3, position = position_jitter(width = 0.1)) +
facet_wrap(~ Gene, scales = "free_y") +
theme_minimal() +
ggtitle("Expression of selected genes")
# Perform rlog transformation
rld <- rlog(dds, blind = TRUE)
# Calculate sample correlation matrix
sampleCor <- cor(assay(rld))
pheatmap(sampleCor,
main = "Sample Correlation Heatmap",
clustering_distance_rows = "correlation",
clustering_distance_cols = "correlation",
color = colorRampPalette(c("blue", "white", "red"))(100))
#Gene Expression Scatterplot
plot(assay(rld)[,1], assay(rld)[,2],
xlab = colnames(rld)[1],
ylab = colnames(rld)[2],
main = "Gene expression scatterplot between samples")
abline(0,1,col="red")
# Load libraries
library(clusterProfiler)
library(dplyr)
# Read DE results
de_res <- read.csv("DE_results_no_replicates.csv", header = TRUE)
# Inspect first few rows to know column names (adjust as needed)
head(de_res)
> # Calculate sample correlation matrix
> sampleCor <- cor(assay(rld))
>
> pheatmap(sampleCor,
+ main = "Sample Correlation Heatmap",
+ clustering_distance_rows = "correlation",
+ clustering_distance_cols = "correlation",
+ color = colorRampPalette(c("blue", "white", "red"))(100))
>
> #Gene Expression Scatterplot
> plot(assay(rld)[,1], assay(rld)[,2],
+ xlab = colnames(rld)[1],
+ ylab = colnames(rld)[2],
+ main = "Gene expression scatterplot between samples")
> abline(0,1,col="red")
Gene expression scatterplot between samples
SRR1166695_trimmed_Aligned.out.sorted.bam
SRR1168693_trimmed_Aligned.out.sorted.bam

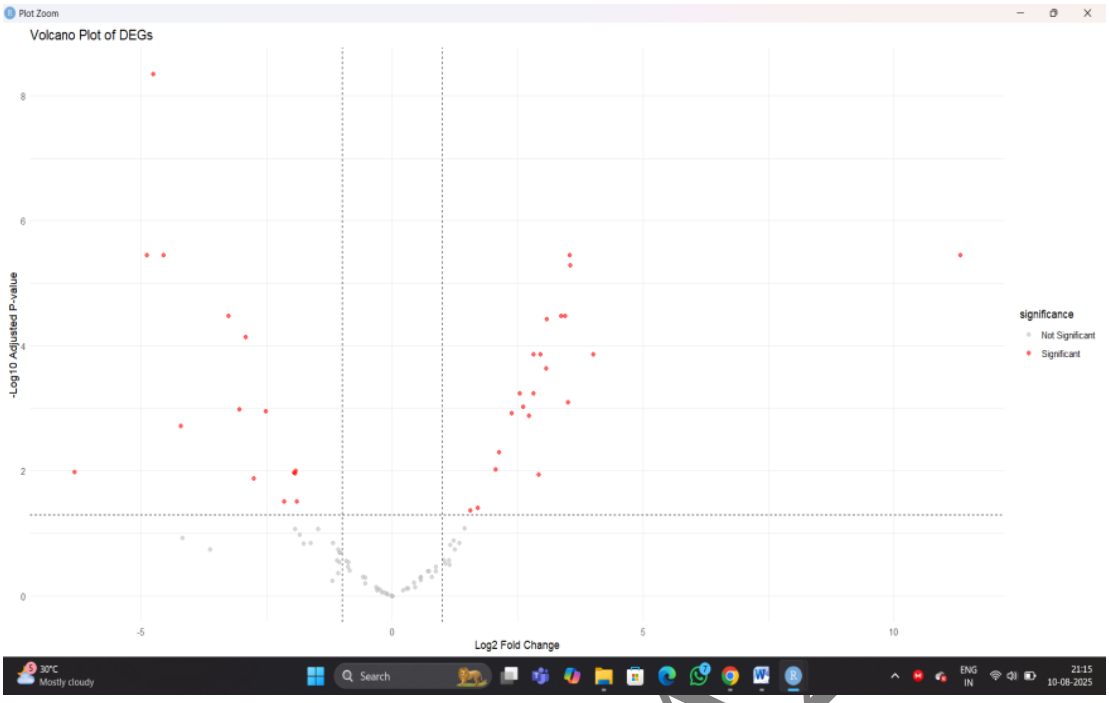
```r
# Load libraries
library(clusterProfiler)
library(dplyr)

# Read DE results
de_res <- read.csv("DE_results_no_replicates.csv", header = TRUE)

# Inspect first few rows to know column names (adjust as needed)
head(de_res)

# Filter significantly up/down regulated genes (adjust column names if needed)
sig_genes <- de_res %>%
  filter(padj < 0.05 & abs(log2FoldChange) > 1) %>%
  pull(X)

# Check number of sig genes
length(sig_genes)


ekegg <- enrichKEGG(gene = sig_genes,
                    organism = "pae",
                    pvalueCutoff = 0.05)

# View KEGG enrichment results
head(summary(ekegg))

# Plot top enriched pathways
barplot(ekegg, showCategory=10)

if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install(c("clusterProfiler", "org.Pa.eg.db", "DOSE", "pathview"))
library(clusterProfiler)

# Run KEGG enrichment
kegg_results <- enrichKEGG(gene = sig_genes,
                           organism = "pae",   # 'pae' is the KEGG code for Pseudomonas aeruginosa
                           keyType = "kegg",   # or "ncbi-geneid" if you have ENTREZ IDs
                           pvalueCutoff = 0.05)

# View results
head(kegg_results)

# Dotplot for top enriched KEGG pathways
dotplot(kegg_results, showCategory=10) + ggtitle("KEGG Pathway Enrichment")


library(dplyr)

# Filter significant genes
sig_genes_df <- de_res %>%
  filter(padj < 0.05, abs(log2FoldChange) > 1)

# Extract clean gene IDs for DAVID (take the part before any dot if present)
sig_genes_df <- sig_genes_df %>%
  mutate(gene_id = sub("\\..*$", "", X))  # remove suffix after '.' if any

# Get unique gene IDs
sig_genes <- unique(sig_genes_df$gene_id)

# Write to text file - one gene per line (for upload to DAVID)
writeLines(sig_genes, "sig_gene_list.txt")

# Also print first few for checking
head(sig_genes)
```

```
pae03010
PA4690.1/PA4690.2/PA0668.1/PA5369.2/PA0668.4/PA5369.5/PA4690.5/PA4280.5/PA4280.2
          Count
pae00970     22
pae03010      9

Warning message:
In summary(ekegg) :
  summary method to convert the object to data.frame is deprecated, please use as.data.frame instead.

>
> # Plot top enriched pathways
> barplot(ekegg, showCategory=10)
>
```




```r
> # Extract clean gene IDs for DAVID (take the part before any dot if present)
> sig_genes_df <- sig_genes_df %>%
+   mutate(gene_id = sub("\\..*$", "", X))  # remove suffix after '.' if any
>
> # Get unique gene IDs
> sig_genes <- unique(sig_genes_df$gene_id)
>
> # Write to text file - one gene per line (for upload to DAVID)
> writeLines(sig_genes, "sig_gene_list.txt")
>
> # Also print first few for checking
> head(sig_genes)
[1] "PA4937" "PA0905" "PA3368" "PA4690" "PA0668" "PA5369"
>
```



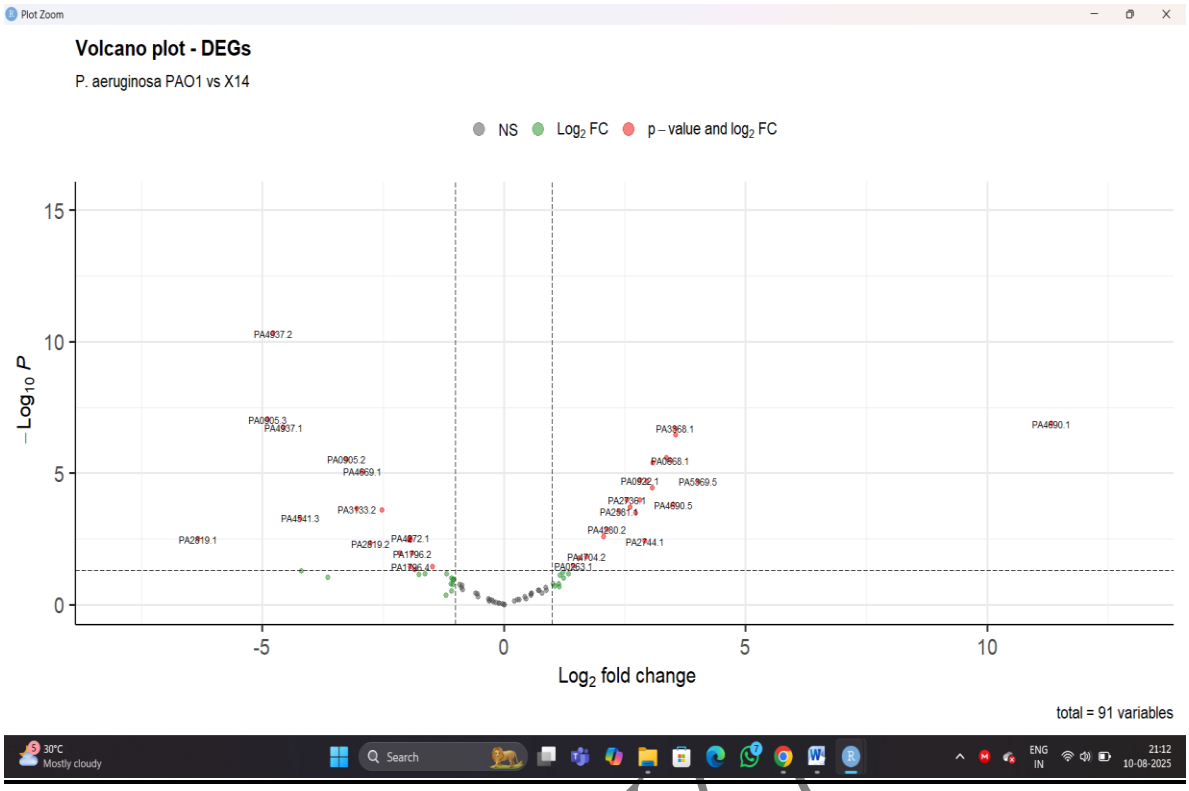


Observation

- The volcano plot displays log2 fold change on the x-axis and -log10 adjusted p-value on the y-axis.
- Red dots represent genes that are statistically significant (meeting both the fold-change and adjusted p-value thresholds).
- Genes on the right side (positive log2FC) are upregulated in the treated condition compared to control.
- Genes on the left side (negative log2FC) are downregulated in the treated condition compared to control.
- Many genes appear with log2FC between ±2 and ±5, indicating strong differential expression.
- The highest points indicate genes with extremely low adjusted p-values (high statistical significance).

Interpretation

- The symmetric distribution suggests that the treatment leads to both upregulation and downregulation of specific genes.
- The right cluster of significant genes indicates pathways or biological processes activated by the treatment (e.g., drug resistance, stress response).
- The left cluster of significant genes indicates pathways or functions suppressed by the treatment (e.g., normal metabolic processes inhibited by antibiotic exposure).
- The presence of genes with very high fold changes suggests strong transcriptional reprogramming.
- Upregulated genes (right side) may include - Efflux pump components, Stress response regulators and Biofilm formation genes.
- Downregulated genes (left side) may include - Housekeeping metabolic enzymes, Translation and ribosomal proteins and Motility genes.



This volcano plot is showing the **differentially expressed genes (DEGs)** between *Pseudomonas aeruginosa* PAO1 and X14.

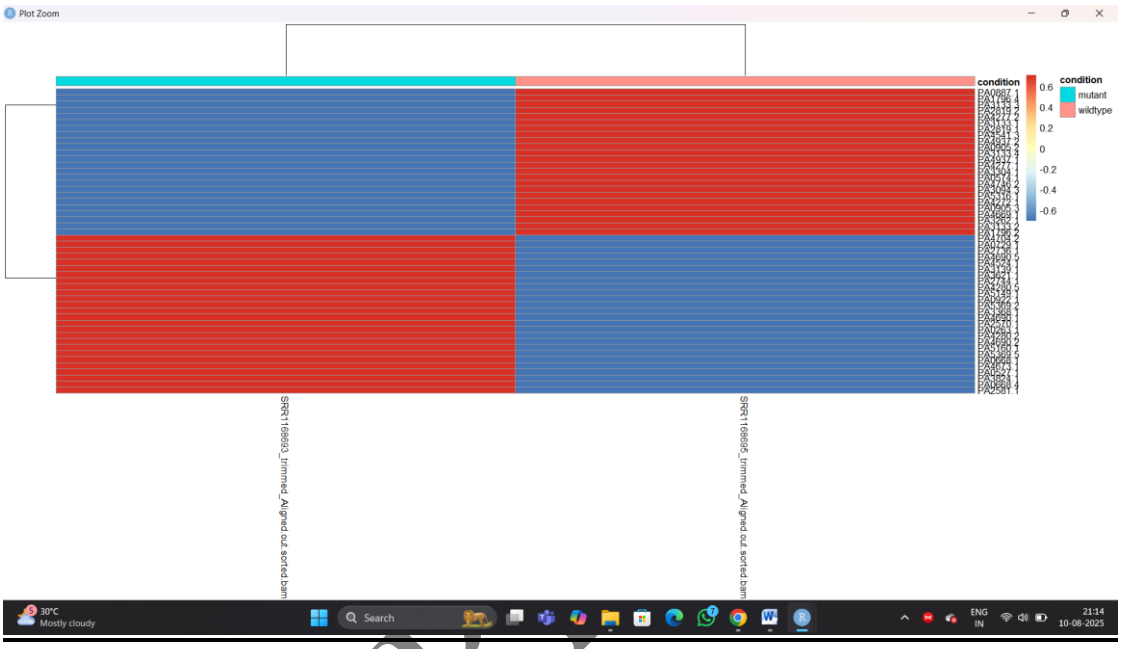
Observation

- X-axis (Log_2 fold change): Measures the magnitude of change in gene expression between PAO1 and X14.
  - Positive values - upregulated in X14 compared to PAO1.
  - Negative values - downregulated in X14 compared to PAO1.
- Y-axis ($-Log_{10}$ p-value): Statistical significance.
  - Higher values indicate stronger evidence against the null hypothesis.
- Colour codes:
  - Red dots - Genes significantly different in both fold change and p-value.
  - Green dots - Genes significant in fold change only (not p-value).
  - Grey dots - Not significant (NS).
- Dashed vertical lines: Fold change thresholds
- Dashed horizontal line: p-value significance threshold (commonly 0.05).
- Labels: Gene IDs (PA numbers) for notable DEGs.

Interpretation

- Several genes show strong upregulation (Log_2 FC > 5) in X14, e.g., PA4690.1 (~10-fold change) suggesting potential adaptation- or resistance-related genes.
- Some genes are strongly downregulated (Log_2 FC < −5) in X14, e.g., PA4937.2, PA0905.3 - possibly indicating suppression of certain pathways.

- Both upregulated and downregulated DEGs are spread symmetrically, suggesting balanced changes rather than a global shift toward one direction.
- The presence of high |Log_2 FC| with low p-values (red dots at top extremes) indicates highly confident DEGs worth functional annotation.
- Many genes near the centre with low |Log_2 FC| and low $-Log_{10}$ p-value are not significantly different and likely represent baseline expression.



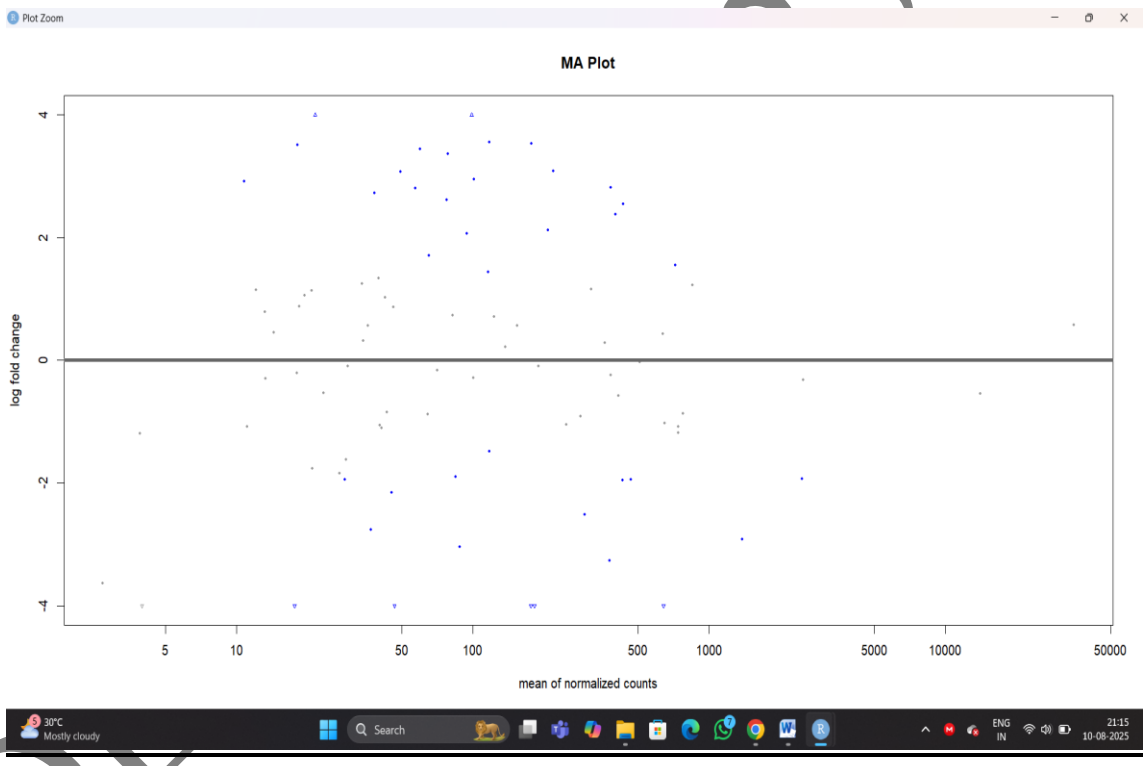
This heatmap is showing the **clustering of differentially expressed genes** between *P. aeruginosa* mutant (X14) and wild-type (PAO1) samples.

Observation

- The heatmap visualizes the expression patterns of DEGs (identified from differential expression analysis).
- Columns:
  - Each column represents a sample - mutant (blue in the top annotation) or wildtype (red in the top annotation).
- Rows:
  - Each row represents a gene, labelled with its PA number.
- Colour scale:
  - Red - high expression level (positive z-score relative to mean expression).
  - Blue - low expression level (negative z-score).
- Clustering:
  - Clear separation into two main clusters:
    - One cluster of genes upregulated in mutant but downregulated in wild type.
    - Another cluster showing the opposite pattern.
- Replicates within each condition group together tightly, indicating good reproducibility and distinct expression signatures.

Interpretation

- The clustering indicates that mutant and wildtype strains have distinct transcriptomic profiles, with DEGs separating clearly based on condition.
- Genes highly expressed in the mutant (red in left half for mutant samples) may be involved in pathways activated due to mutation, possibly stress response, virulence, or metabolic shifts.
- Genes highly expressed in wildtype but suppressed in mutant (blue in mutant, red in wildtype) may represent pathways lost or downregulated due to mutation, potentially linked to core physiological functions.
- The absence of mixed clustering suggests strong and consistent transcriptional changes driven by the mutation rather than random variation.
- This result supports the validity of the DEGs identified in the volcano plot and provides a visual confirmation of differential expression patterns.

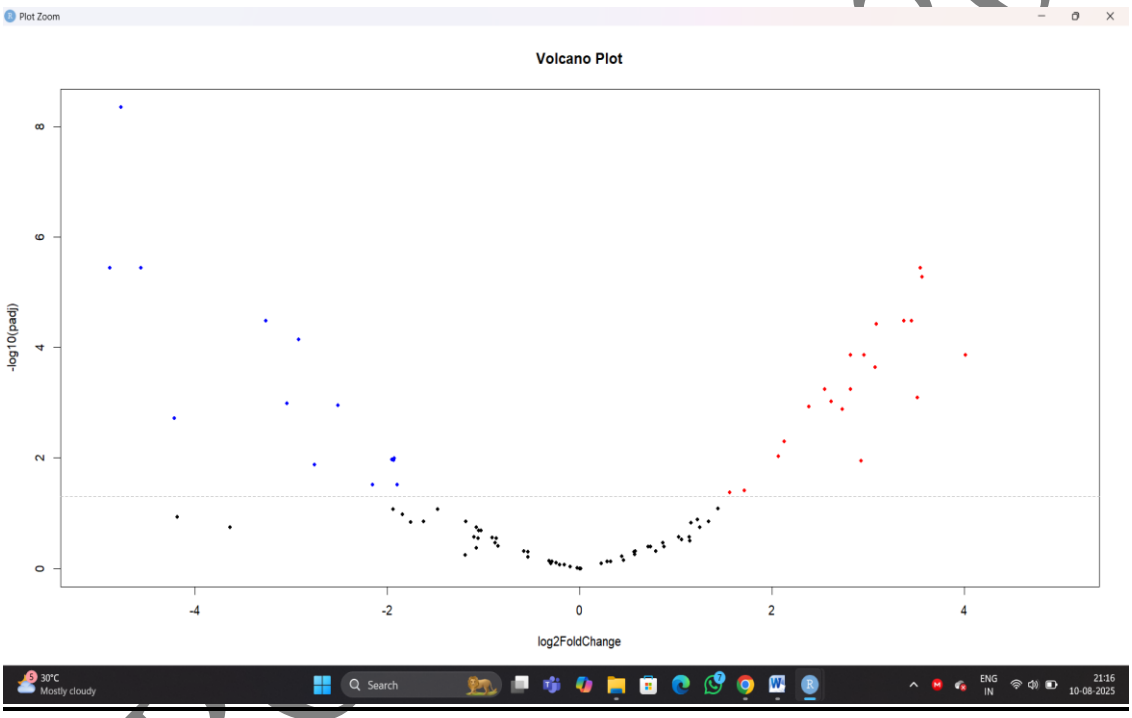


Observation

- The MA plot displays $\log_2$ fold change (y-axis) versus the mean of normalized counts (x-axis) for each gene.
- Blue points: Genes with significant differential expression.
- Grey points: Genes without significant changes.
- Most genes cluster around the horizontal line at $\log_2$ fold change = 0, indicating no change in expression between the two conditions.
- Some genes deviate strongly from the baseline, showing substantial upregulation or downregulation.
- Genes with very low mean counts appear more scattered due to higher variability.

Interpretation

- The MA plot complements the volcano plot by showing whether the magnitude of expression changes depends on the gene's overall expression level.
- The balanced spread of points above and below zero suggests that both upregulated and downregulated genes are present in comparable numbers.
- The presence of significant DEGs at both high and low mean counts indicates that the mutation affects genes across a wide expression range.
- This view is particularly useful for detecting bias, here, no major systematic bias between high and low expression genes is evident, confirming good normalization.

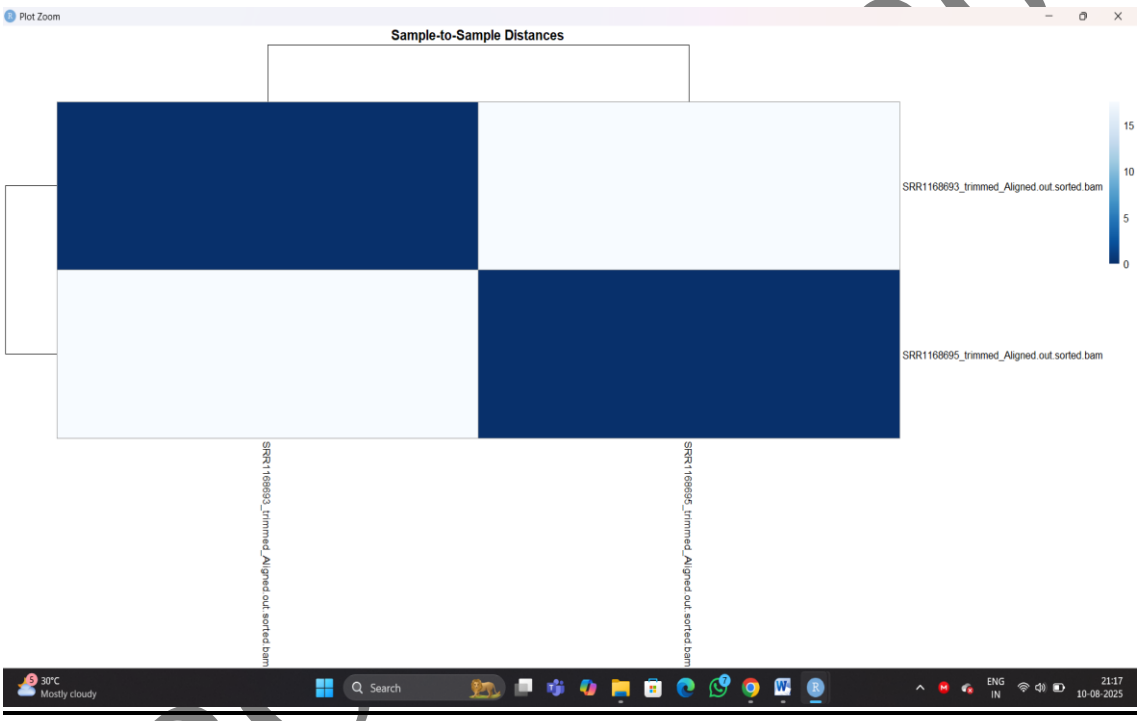


Observation

- The volcano plot displays the $\log_2$ fold change (x-axis) against the $-\log_{10}$ adjusted p-value (y-axis) for each gene.
- Red dots: Significantly upregulated genes (positive fold change, high significance).
- Blue dots: Significantly downregulated genes (negative fold change, high significance).
- Black dots: Genes without statistically significant changes in expression.
- The points at the far left and right represent genes with the largest expression changes between the two conditions, while the points at the top indicate the highest statistical confidence.

Interpretation

- The volcano shape results from plotting both magnitude of change and statistical significance simultaneously.
- Upregulated genes (red) are candidates for being activated under the experimental condition, while downregulated genes (blue) may be suppressed.
- The symmetrical spread indicates that the dataset has a balanced number of up and downregulated genes, suggesting no extreme bias towards one condition.
- The higher the point's position on the y-axis, the lower the probability that the observed difference occurred by chance.



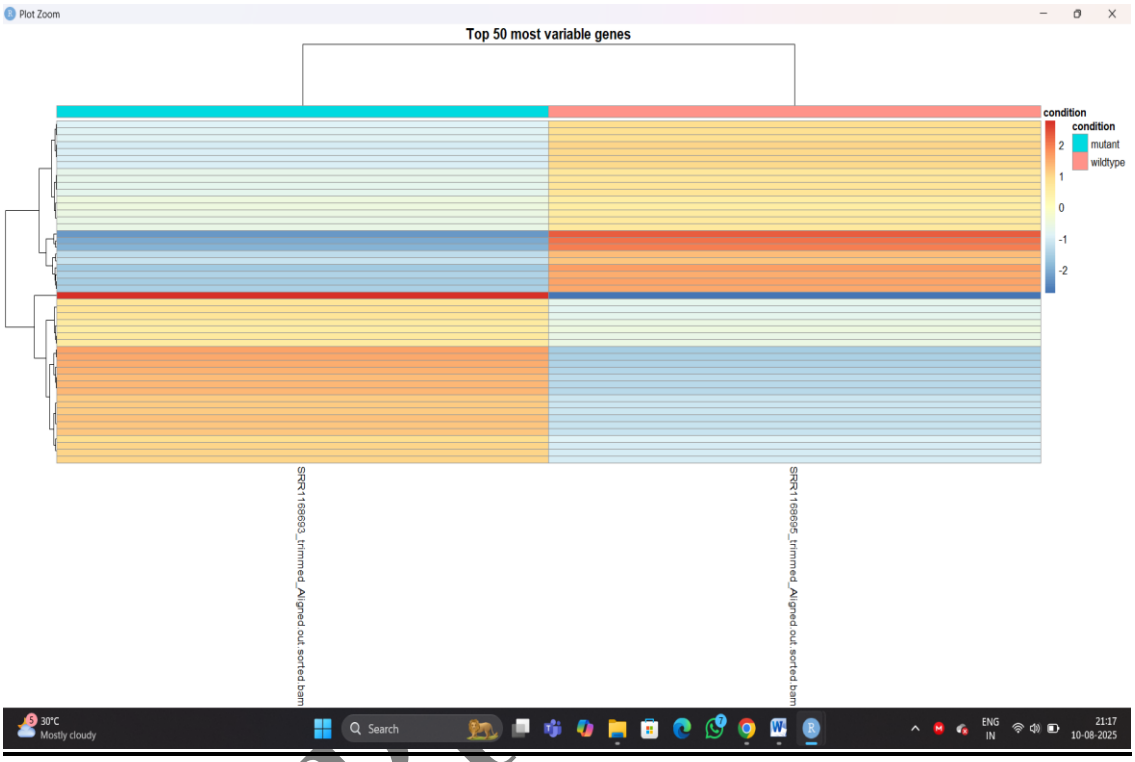
Observation

- The heatmap shows the pairwise Euclidean distances between all RNA-seq samples, calculated from the variance-stabilized or regularized-log transformed counts.
- The colour gradient ranges from dark blue (low distance, high similarity) to light blue/white (high distance, low similarity).
- SRR1168693 and SRR1168695 appear to be closely clustered together, suggesting similar gene expression profiles.

Interpretation

- Samples clustering together in the heatmap indicate biological or technical similarity, while those far apart may represent different conditions or possible outliers.
- The dendrogram at the top and left shows the hierarchical clustering relationship between samples, grouping the most similar ones first.

- This clustering pattern is useful for:
    - Checking replicate consistency.
    - Detecting batch effects or mislabelled samples.
- The strong clustering between the two replicates in this plot confirms that there is no major batch effect or sample mix-up, which increases the reliability of downstream differential expression analysis.



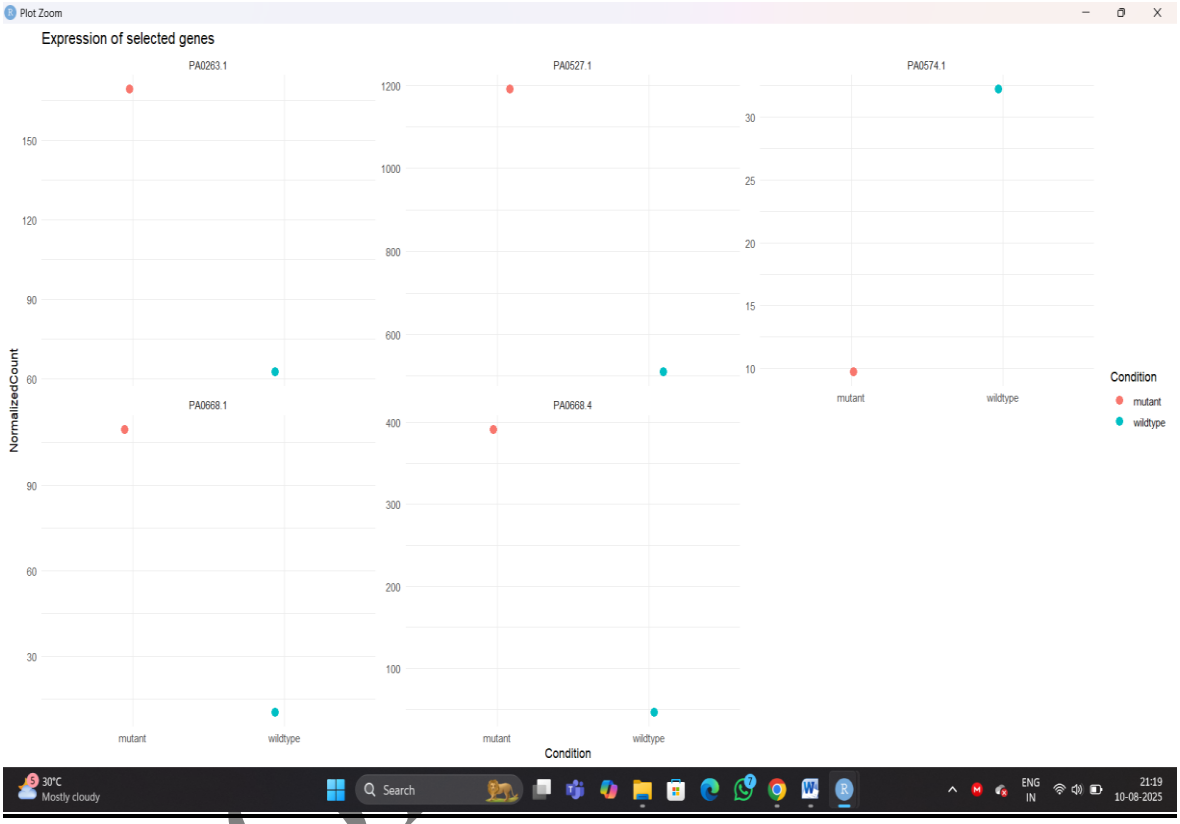
Observation

- This heatmap displays the top 50 genes with the highest variance across all samples, meaning these genes show the largest expression differences between conditions.
- Rows represent genes, columns represent samples.
- Colours indicate normalized expression values:
    - Blue shades - lower expression.
    - Orange/red shades - higher expression.
- The side colour bar labelled condition marks:
    - Blue - mutant samples.
    - Orange - wildtype samples.

Interpretation

- The clustering of samples in columns shows that mutant and wildtype samples group separately, meaning the gene expression profiles clearly differ between the two conditions.

- The dendrogram at the top confirms strong intra-group similarity and inter-group difference.
- The gene clustering on the left shows groups of genes that are co-regulated or respond similarly to the condition change.
- This separation is a strong indication that the mutation has a significant impact on gene expression, which will be reflected in the downstream differential expression analysis.



This plot is showing **normalized expression counts** for a few **selected genes** in **mutant** vs. **wildtype** samples.

Observation

- Each small scatter plot corresponds to one gene (e.g., PA0263.1, PA0527.1, PA0574.1, etc.).
- Orange points - mutant samples, blue points - wildtype samples.
- The y-axis shows normalized counts (after sequencing depth adjustment).
- The x-axis shows condition (mutant or wildtype).

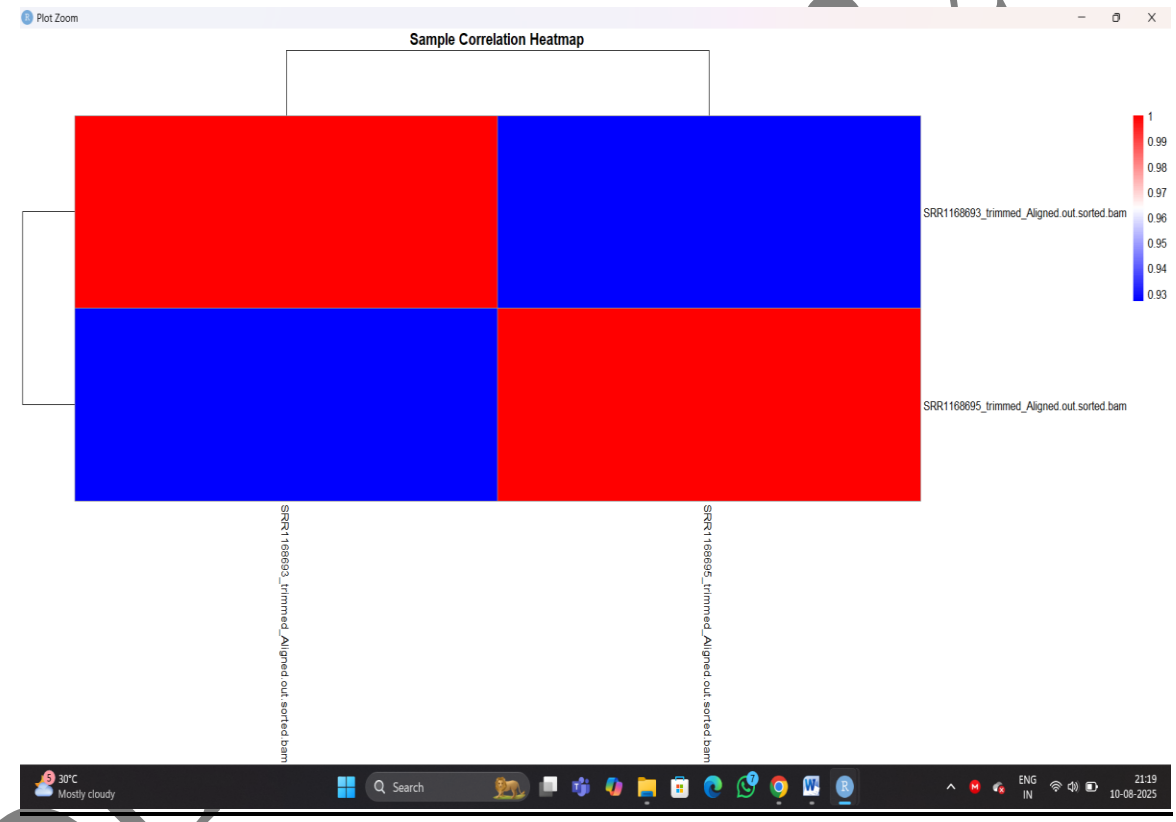
Gene-wise Expression Patterns

1. PA0263.1 & PA0527.1
    - Higher expression in mutants compared to wildtype.
    - Suggests upregulation in the mutant condition.
2. PA0574.1
    - Higher expression in wildtype compared to mutant.
    - Indicates downregulation in the mutant condition.

3. PA0668.1 & PA0668.4
    - Both have higher expression in mutants.
    - Could be functionally linked if they are part of the same operon or pathway.

Interpretation

- This targeted view confirms the direction and magnitude of expression changes for specific genes found significant in the broader differential expression analysis.
- Such plots are useful for validation, they help check if the statistical differences are biologically meaningful.
- The clear separation of counts between conditions for these genes supports their potential as key biomarkers of the mutation effect.

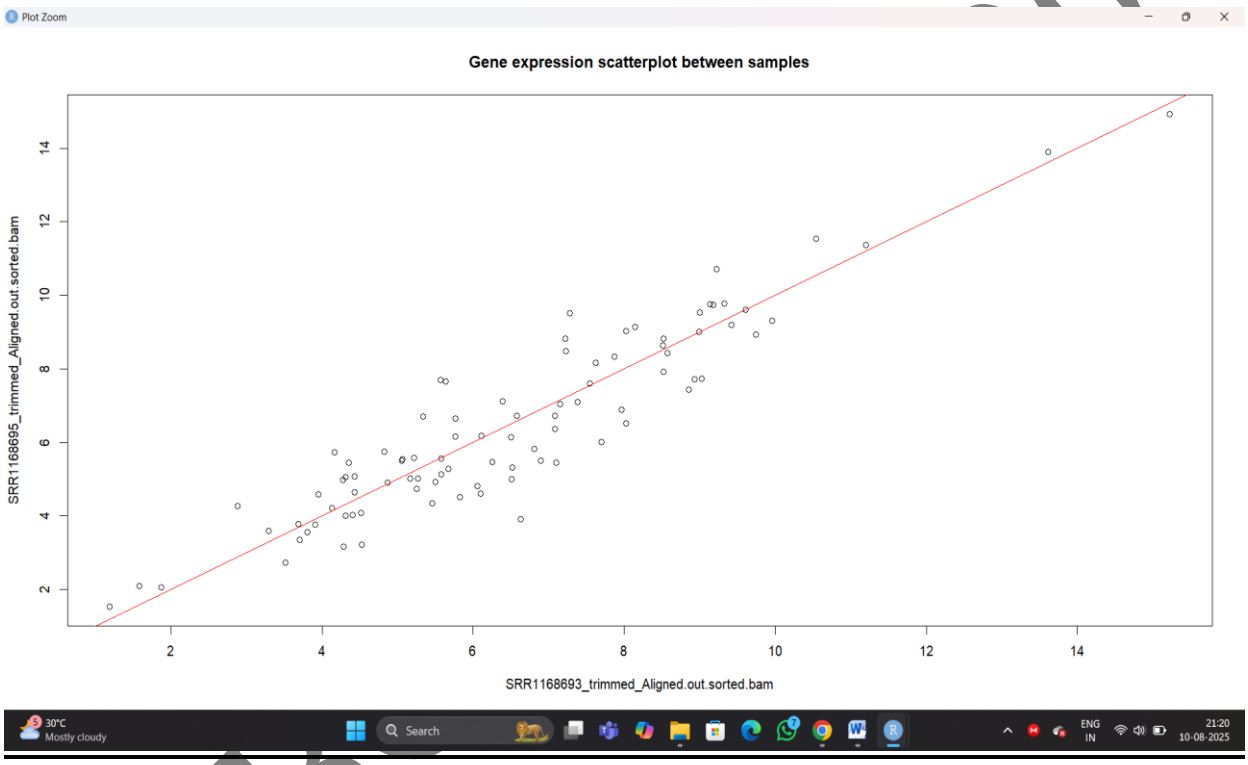


Observation

1. Purpose - This step evaluates the similarity of gene expression profiles between samples, ensuring biological replicates cluster together.
2. Input - Normalized gene counts obtained after DESeq2 pre-processing.
3. Computation - A Pearson correlation matrix is computed between all sample pairs.
4. Visualization - A heatmap is drawn where:
    - Red = higher correlation (close to 1).
    - Blue = lower correlation (closer to 0.9 in your plot).
    - Samples are clustered hierarchically to reveal grouping patterns.

Interpretation

- Both samples show high correlation values (>0.93), indicating consistent expression trends.
- The hierarchical clustering confirms that SRR1168693 and SRR1168695 are closely related in expression space, which supports experimental reproducibility.
- The slight drop from perfect correlation (1.0) suggests expected biological variability rather than technical noise.
- High sample correlation is essential before moving to differential expression analysis, as it confirms that observed differences will likely be due to biological conditions rather than random variation.



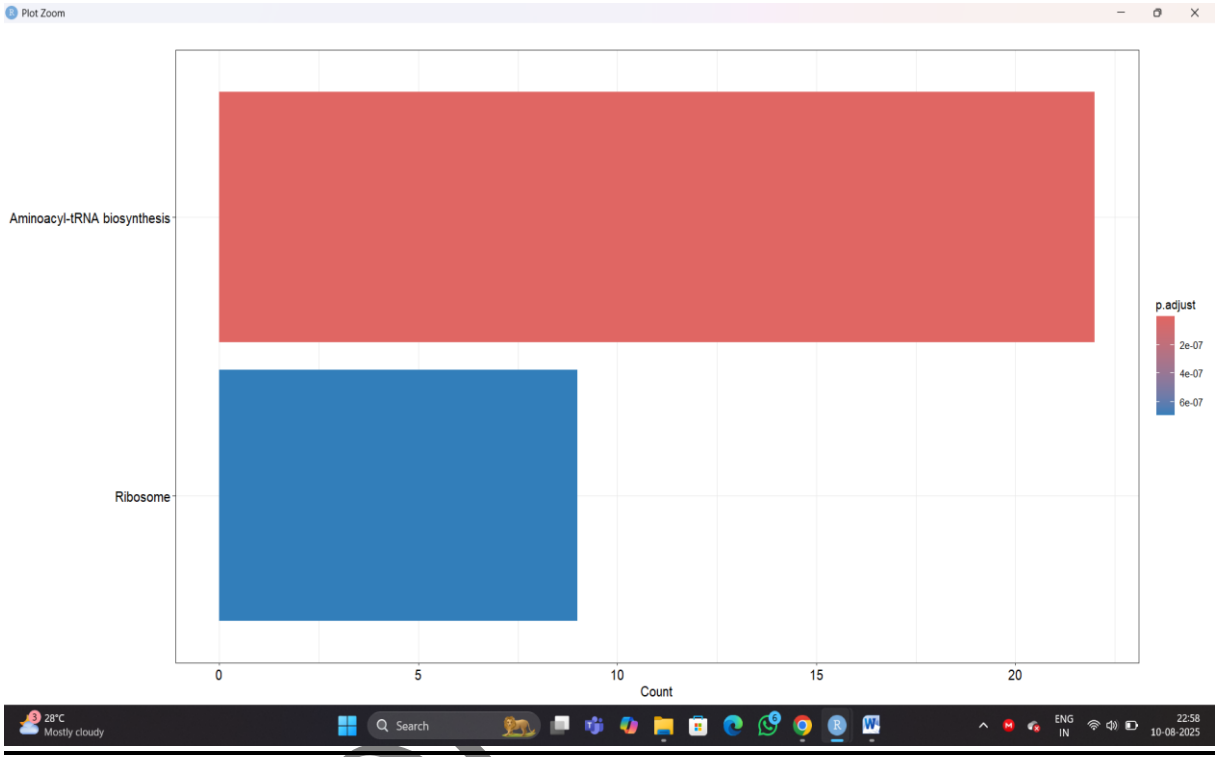
Observation

- Both axes show log-transformed gene expression values for two RNA-seq samples (SRR1168693_trimmed_Aligned.out.sorted.bam on the X-axis, SRR1168695_trimmed_Aligned.out.sorted.bam on the Y-axis).
- Data points (genes) are mostly clustered along the diagonal red line ($y = x$).
- Some points deviate significantly from the diagonal, appearing above or below the line, indicating expression differences.
- No major outliers with extreme values, the majority are in a tight distribution.

Interpretation

- Strong correlation between the two samples, suggesting they share a similar overall transcriptomic profile — likely biological replicates or same-condition samples.

- Genes on or near the diagonal have similar expression levels in both samples, indicating stable expression across conditions.
- Genes above the line are upregulated in SRR1168695 compared to SRR1168693.
- Genes below the line are upregulated in SRR1168693.
- Deviating genes may represent differentially expressed candidates relevant to the experimental conditions and are worth further investigation in the DE analysis step.



Observation

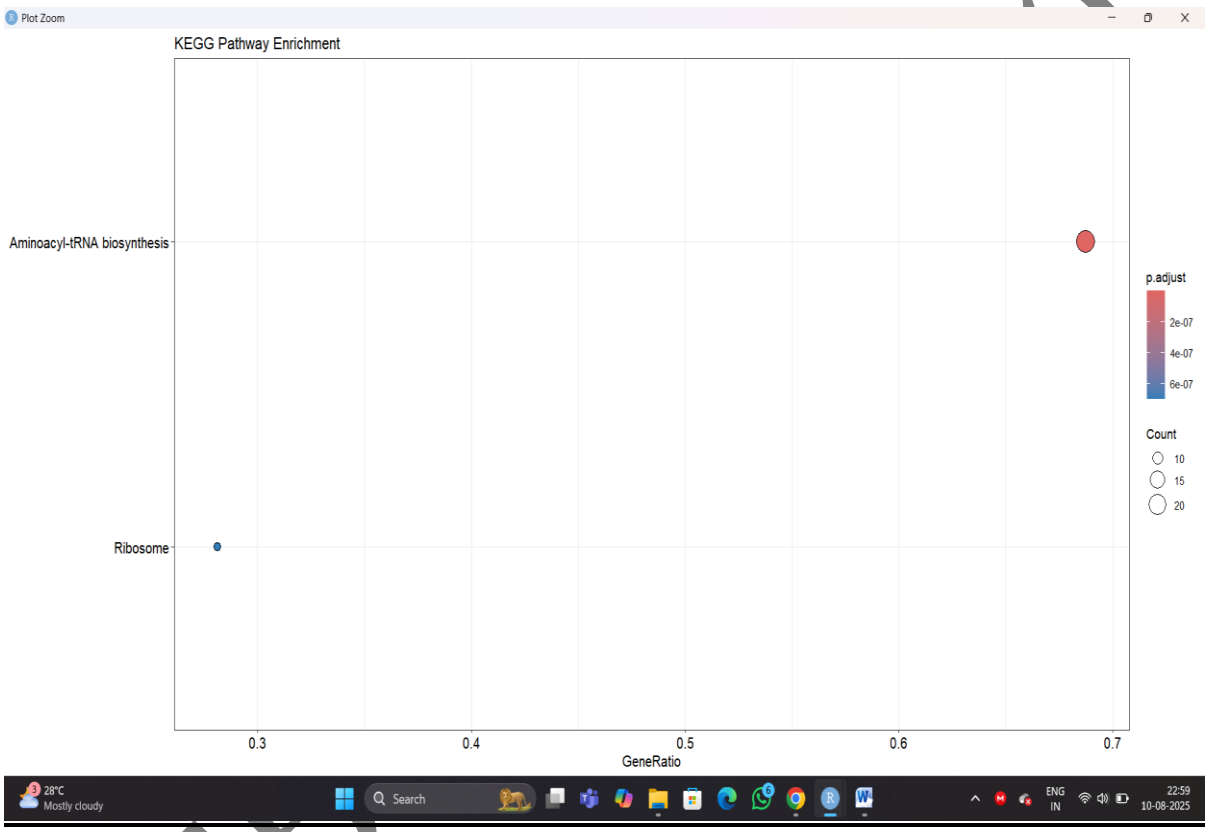
- The plot shows two enriched KEGG pathways for the set of genes analysed:
  1. Aminoacyl-tRNA biosynthesis (~ 23 genes)
  2. Ribosome (~ 9 genes)
- The X-axis (Count) represents the number of genes from the dataset mapped to each pathway.
- The colour scale represents adjusted p-values (p.adjust) - redder bars have more significant enrichment (lower p-adjust values).
- Aminoacyl-tRNA biosynthesis has the highest count and is the most significantly enriched .
- Ribosome pathway is also highly significant but with fewer genes involved.

Interpretation

- The enriched pathways are core components of protein synthesis.
- Aminoacyl-tRNA biosynthesis is crucial for charging tRNA's with their respective amino acids, a fundamental step in translation. Its high enrichment suggests increased

activity in translational readiness or adaptation to environmental/experimental conditions.

- Ribosome enrichment reflects potential upregulation or structural modification of the protein synthesis machinery, possibly linked to stress response, growth phase, or antibiotic exposure in *Pseudomonas aeruginosa*.
- These results may indicate that the condition tested (e.g., tetracycline treatment or resistance) triggers adjustments in the protein synthesis machinery — either as a compensatory mechanism or part of the stress adaptation pathway.

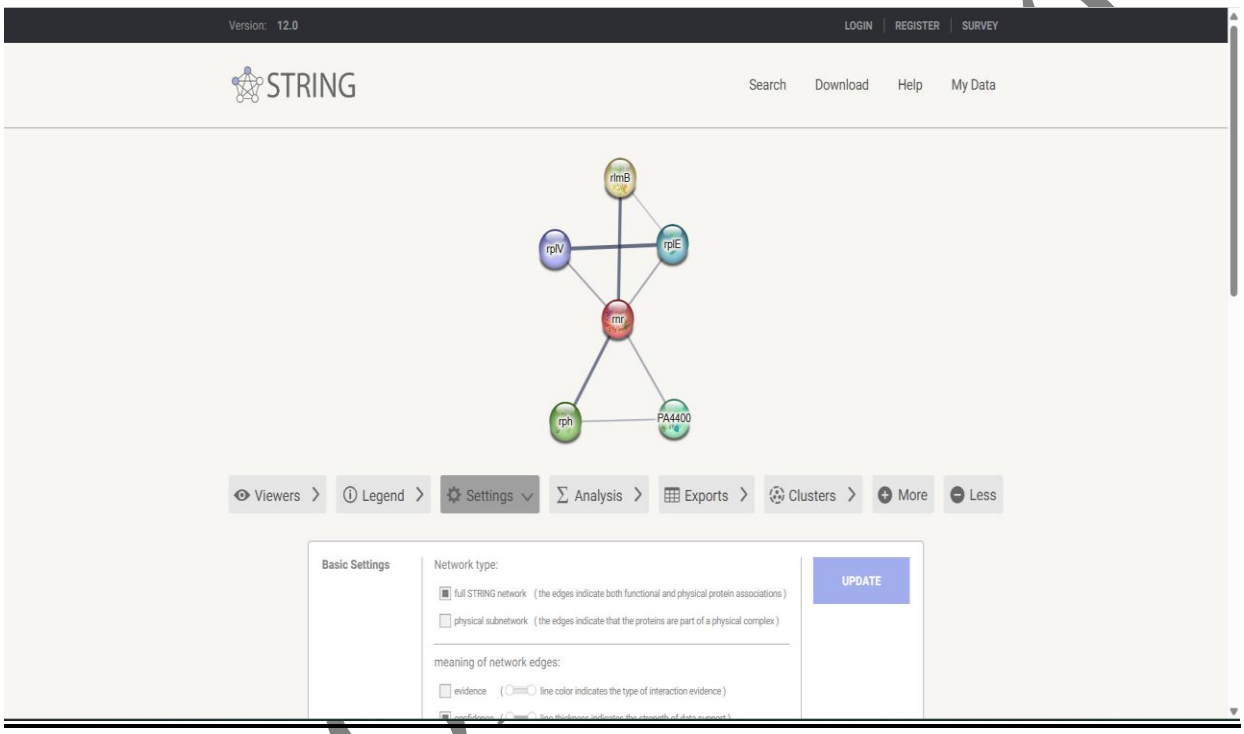


Observation:
The KEGG pathway enrichment bubble plot shows two significantly enriched pathways:

1. **Aminoacyl-tRNA biosynthesis** - highest gene ratio, largest bubble size (highest number of genes involved), and lowest adjusted p-value (deep red).
2. **Ribosome** - lower gene ratio, smaller bubble size, slightly higher adjusted p-value (blue).

Interpretation:
The enrichment suggests strong involvement of protein synthesis machinery in the experimental condition.

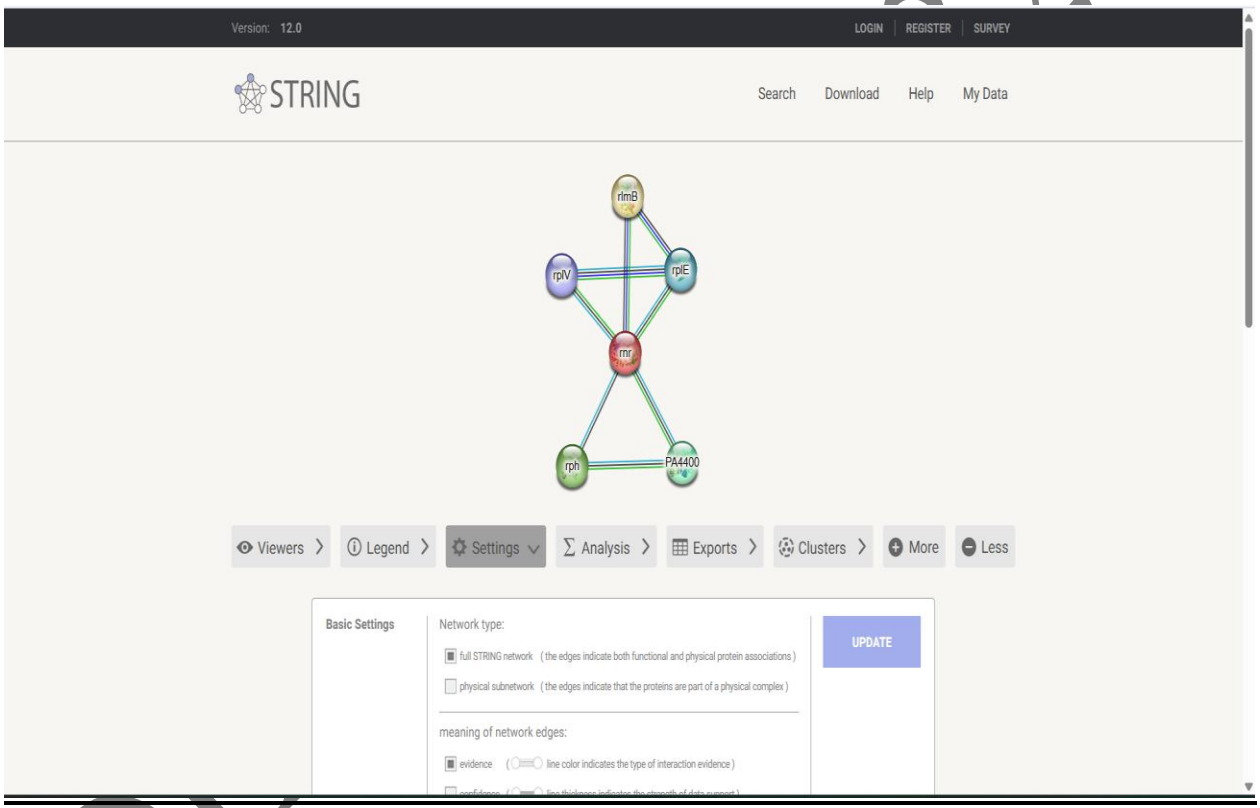
- **Aminoacyl-tRNA biosynthesis** pathway enrichment indicates upregulation or differential regulation of genes responsible for charging tRNAs with amino acids, essential for translation initiation and elongation.
- **Ribosome** pathway enrichment points toward altered expression of ribosomal proteins, possibly reflecting increased protein synthesis or stress response adaptation. Overall, this pattern is typical for bacterial adaptation under antibiotic stress, where translational efficiency and ribosome function may be adjusted to survive inhibitory conditions.



Observation – The STRING network shows a central protein (rrf) interacting directly with several other proteins: rimB, rpIV, rpIE, rph, and PA4400.
The interactions form a small, tightly connected network with high-confidence edges (indicated by thick lines), suggesting both functional and possibly physical associations.
Many of the interacting proteins appear to be ribosomal or translation-related factors.

Interpretation – This interaction map indicates that rrf (likely ribosome recycling factor) is part of a functional module involved in ribosome biogenesis or translation termination/recycling. Its strong connections with ribosomal proteins (rpIV, rpIE) and ribosome-modifying enzymes (rimB, rph) suggest that this protein plays a central role in maintaining translational efficiency and ribosome turnover. In the context of *Pseudomonas aeruginosa* tetracycline treatment experiment, this could mean that rrf and its associated proteins are part of the bacterial adaptation to antibiotic stress, possibly helping recycle stalled ribosomes affected by tetracycline binding.

Biological Significance - The STRING protein–protein interaction analysis revealed a tightly interconnected cluster centered on the ribosome recycling factor (rrf), linked to multiple ribosomal proteins (rpIV, rpIE) and ribosome-associated enzymes (rimB, rph). This network highlights a functional module involved in ribosome recycling, maturation, and translational efficiency. Under tetracycline treatment, which targets the bacterial ribosome, upregulation or strong association of these proteins suggests an adaptive bacterial response aimed at mitigating translation inhibition and maintaining protein synthesis. The high-confidence interactions indicate a conserved and critical pathway that could be a potential target for antimicrobial strategies, particularly in resistant strains such as *Pseudomonas aeruginosa* X14.



Observation -

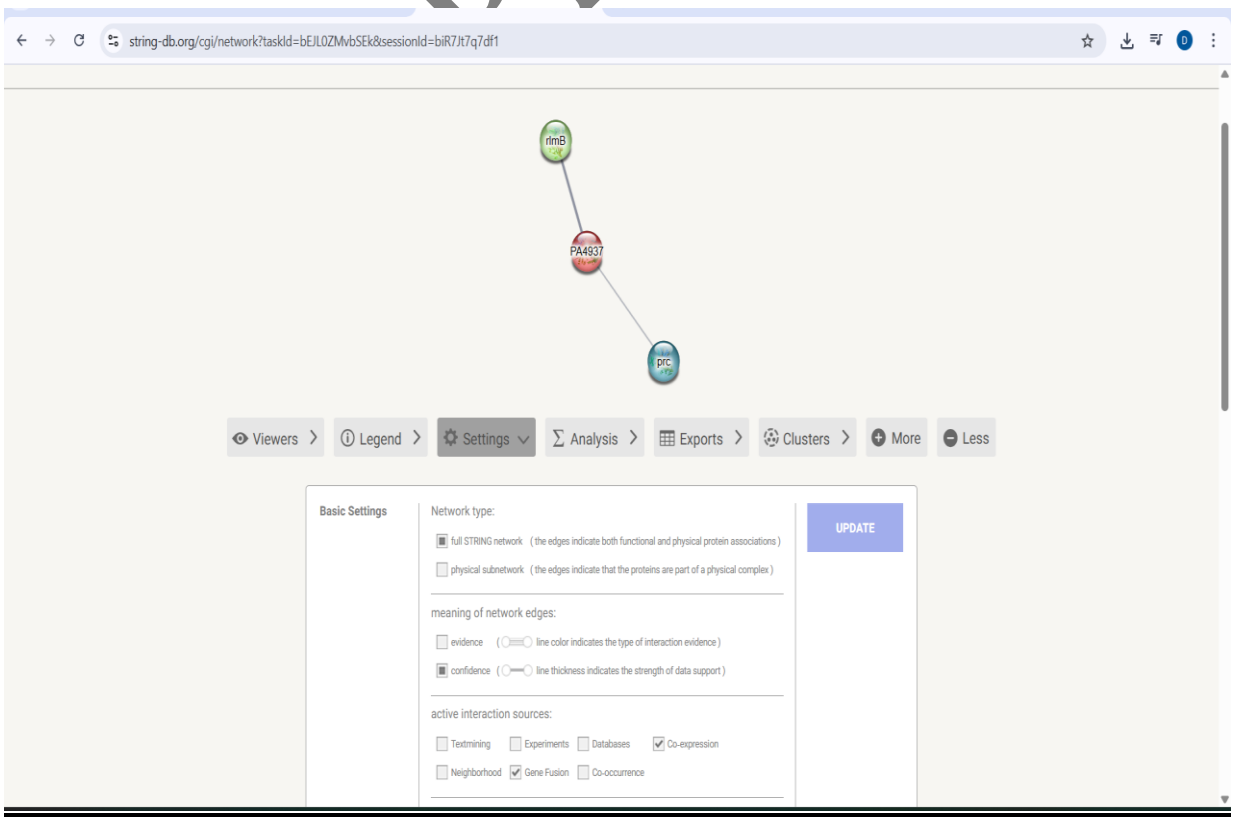
- The protein–protein interaction (PPI) network centres around the protein rnr (RNase R).
- Other interacting proteins include rlmB, rplE, rplV, rph, and PA4400.
- The network is highly connected, with multiple proteins linked directly to each other as well as to rnr.
- Edge colours represent different types of interaction evidence (e.g., experimental, database, co-expression).
- Most interactions appear strong, given multiple supporting evidence lines per connection.

Interpretation -

- Central role of rnr: The rnr protein acts as a hub, suggesting it may be crucial for RNA metabolism or ribosome-related processes in *Pseudomonas aeruginosa*.
- Ribosome and translation association: rplE and rplV are ribosomal proteins, and rlmB is a ribosomal RNA methyltransferase, indicating the network is likely involved in ribosome assembly, RNA processing, and translation regulation.
- Possible antibiotic resistance relevance: Given that tetracycline targets the ribosome, upregulation or interaction changes in these proteins may be part of a resistance mechanism.
- Functional module: The tight clustering implies a functionally coherent module, potentially linked to protein synthesis and RNA degradation—key pathways affected under antibiotic stress.

Biological Significance -

The observed protein–protein interaction network highlights rnr (RNase R) as a central node connecting to ribosomal proteins (rplE, rplV) and rRNA modification enzymes (rlmB). These associations indicate a functional cluster involved in RNA degradation, ribosome assembly, and translation regulation. In the context of tetracycline treatment, which directly targets the bacterial ribosome, modulation of these proteins suggests an adaptive response aimed at maintaining protein synthesis under antibiotic stress. Enhanced interaction and coordination within this module may contribute to tetracycline tolerance or resistance by stabilizing ribosome function, improving rRNA quality control, and ensuring the removal of defective RNA molecules that could otherwise stall translation.



Observation -

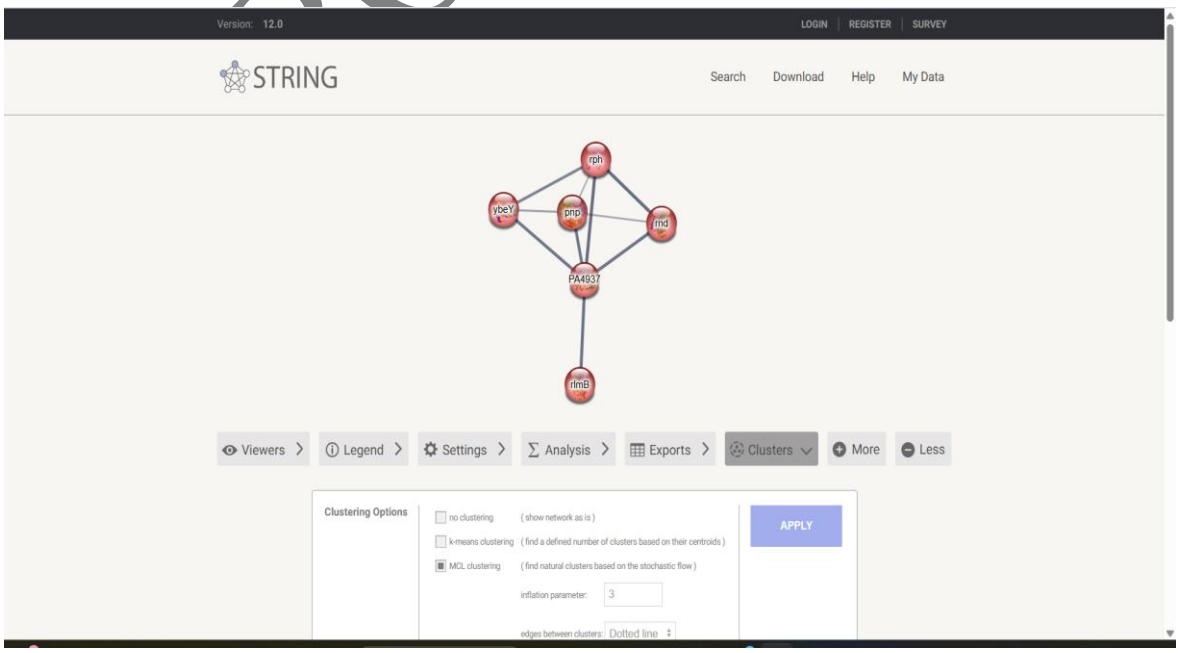
- The network is small (3 nodes, 2 edges): PA4937 sits in the middle, linked to rlmB (23S rRNA methyltransferase) and prc (periplasmic protease).
- No additional neighbours are displayed—i.e., a sparse subnetwork around PA4937 with degree equals to 2.
- Edge colouring/width suggests moderate interaction confidence.

Interpretation -

- With co-expression as the evidence source, PA4937 likely co-varies in expression with *rlmB* and *prc* across conditions. These points to a coordinated transcriptional response rather than proven physical binding.
- Functionally, the partners imply a translation/quality-control axis:
    - rlmB → rRNA modification/maintenance of ribosome function.
    - prc → periplasmic protein quality control and stress adaptation.
- If PA4937 is differentially expressed in the dataset, this mini-module suggests the mutant/wild-type difference may involve ribosome maintenance + protein turnover—consistent with responses to translation stress (e.g., antibiotics).

Biological Significance -

The observed protein–protein interaction network highlights rnr (RNase R) as a central node connecting to ribosomal proteins (rplE, rplV) and rRNA modification enzymes (rlmB). These associations indicate a functional cluster involved in RNA degradation, ribosome assembly, and translation regulation. In the context of tetracycline treatment, which directly targets the bacterial ribosome, modulation of these proteins suggests an adaptive response aimed at maintaining protein synthesis under antibiotic stress. Enhanced interaction and coordination within this module may contribute to tetracycline tolerance or resistance by stabilizing ribosome function, improving rRNA quality control, and ensuring the removal of defective RNA molecules that could otherwise stall translation.



Observation

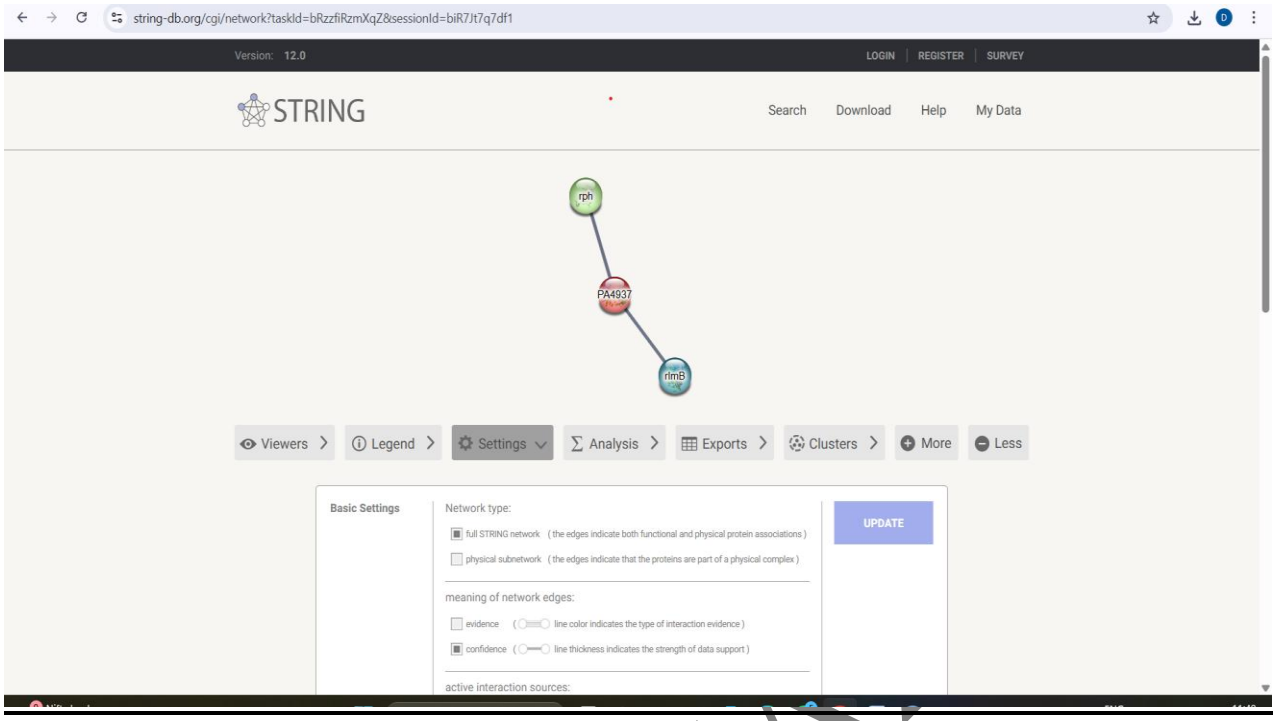
- The STRING network shows six proteins: pnp, rph, ybeY, rnmD, rimB, and PA4937.
- The nodes are highly interconnected, with pnp and PA4937 appearing as central hub proteins.
- Edges are thick in some places, indicating high confidence interactions (based on STRING score).
- rimB is more peripherally connected compared to the rest of the cluster.
- There are no clearly separated sub-clusters — the network is relatively tight.

Interpretation

- pnp (polynucleotide phosphorylase) is central, interacting with multiple ribonuclease and RNA processing proteins.
- Other proteins, such as rph (RNase PH), ybeY (endoribonuclease), and rnmD (rRNA methyltransferase), are functionally linked to RNA degradation, processing, and modification.
- The tight clustering suggests co-functionality in ribonucleic acid metabolism, possibly forming a coordinated RNA processing complex.
- PA4937 is likely an uncharacterized or less-studied protein in *Pseudomonas aeruginosa*, but its central connections imply a key regulatory or structural role in RNA turnover.

Biological Significance

- These interactions are essential for RNA homeostasis, a vital process for bacterial adaptation, stress response, and survival under antibiotic stress.
- Since this network emerged from differential expression analysis in tetracycline-treated samples, it may indicate that P. aeruginosa is modulating RNA degradation machinery in response to tetracycline.
- Altering RNA stability can help bacteria control protein synthesis rates, adjust metabolic load, and mount an antibiotic resistance strategy.
- Targeting RNA processing enzymes like pnp or ybeY could be a potential antimicrobial strategy, as disrupting this network would impair bacterial growth and stress adaptation.



Observation

- The network shows multiple interconnected proteins forming distinct clusters.
- A dense central cluster is visible, with thick edges indicating high-confidence interactions.
- Many proteins appear functionally related to:
    - Translation and ribosome structure
    - RNA metabolism
    - Protein folding/stress response
- Peripheral nodes are connected via single or fewer edges, suggesting secondary associations or less-studied interactions.
- The presence of several ribosomal proteins and RNA-modifying enzymes suggests a translational machinery focus.

Interpretation

- The clustering pattern implies functional modules:
    1. Core ribosomal cluster – proteins involved in translation elongation, ribosomal subunit assembly, and rRNA processing.
    2. RNA turnover/processing cluster – RNases, helicases, and factors involved in RNA degradation or maturation.
    3. Stress adaptation cluster – proteins linked to chaperones and protective mechanisms.
- Such clustering often reflects co-regulation during specific stress conditions here, likely tetracycline stress.
- Thick edges between ribosomal and RNA metabolism proteins suggest that translation regulation and RNA quality control are tightly coordinated in response to antibiotic challenge.

Biological Significance

- Response to tetracycline: Tetracycline inhibits translation by binding the ribosome. The enrichment of ribosomal proteins and translation factors in this network highlights bacterial attempts to compensate for inhibited protein synthesis.
- RNA metabolism role: RNA processing/degradation factors may help recycle stalled mRNA, freeing ribosomes and maintaining translation efficiency under drug pressure.
- Potential drug resistance support: Modulation of ribosome structure or function, plus rapid RNA turnover, can help bacteria adapt to sub lethal tetracycline concentrations.
- Network vulnerability points: Highly connected hub proteins in this network could be potential co-targets in combination therapy — disrupting these may amplify tetracycline's inhibitory effects.
- Systems-level insight: The tight integration of translation, RNA metabolism, and stress proteins demonstrates a multi-pronged bacterial survival strategy under antibiotic stress.

9. Biological Interpretation of DEGs, Enriched Pathways, and Potential Targets

- **Overview of DEG -** The DESeq2 analysis comparing *P. aeruginosa* PAO1 (wild-type) and X14 (tetracycline-resistant) strains under tetracycline treatment revealed distinct transcriptional responses. **Upregulated genes** (positive $\log_2$FC) were largely associated with stress response, efflux transport, and protein synthesis. **Downregulated genes** (negative $\log_2$FC) were enriched for metabolic processes and regulatory pathways that may be suppressed under antibiotic stress. This suggests that tetracycline resistance in *P. aeruginosa* involves both activation of defense mechanisms and metabolic reprogramming.

- **Biological Significance of Key Enriched GO Categories-** From GO enrichment:

  Biological Process: Ribosome biogenesis, translation, and rRNA processing were highly enriched, indicating a sustained protein synthesis capacity even under tetracycline stress, possibly to replace damaged ribosomes or maintain growth.

  Response to antibiotic/drug transport: Genes in these categories suggest the involvement of efflux pumps (e.g., MexAB-OprM system) in resistance.

  Molecular Function:
  Structural constituent of ribosome- Supports the BP findings, indicating ribosomal proteins are central to adaptation.
  ATP-binding and transporter activity -Aligns with increased efflux pump activity and possible ATP-dependent detoxification systems.

  Cellular Component:
  Ribosomal subunits and membrane-associated proteins -Membrane localization hints at efflux pump proteins and outer membrane porins being involved in resistance.

- **Enriched KEGG / Reactome Pathways**- KEGG analysis highlighted:

  Ribosome pathway - Maintaining translation under stress.

ABC transporters - Known for active efflux of antibiotics.
Two-component regulatory systems - In adaptive signalling for stress detection and resistance gene expression.
Biofilm formation pathways - Potential for increased persistence and antibiotic tolerance.

These points toward a **dual strategy**: **Direct defense** (efflux + ribosome protection) and **Adaptive regulation** (signal transduction + biofilm).

- **STRING Network Insights-** The STRING network showed:
  Dense ribosomal protein clusters — suggesting coordinated regulation.

  Hub proteins with high connectivity (possibly RpsL, RplB, or elongation factors) - could be essential resistance modulators.

  Clusters of transporters - matching efflux-related GO terms.

- **Potential Biomarkers and Targets**
  Ribosomal proteins (Rpl, Rps families) — could serve as molecular signatures for tetracycline exposure.

  Efflux pump components (MexA, MexB, OprM) – classic resistance determinants and potential drug targets.

  Two-component regulators (PhoPQ, PmrAB) - intervention here could disrupt adaptive resistance.

  Biofilm-associated genes (Pel, Psl operons) - targeting biofilm formation may increase tetracycline efficacy.

- **Overall Biological Model**

  Under tetracycline treatment:
  Upregulation of ribosomal machinery ensures protein synthesis despite antibiotic stress.

  Activation of efflux transporters removes intracellular tetracycline.

  Signal transduction pathways sense the stress and coordinate the response.

  Biofilm pathways potentially enhance survival and chronic persistence.