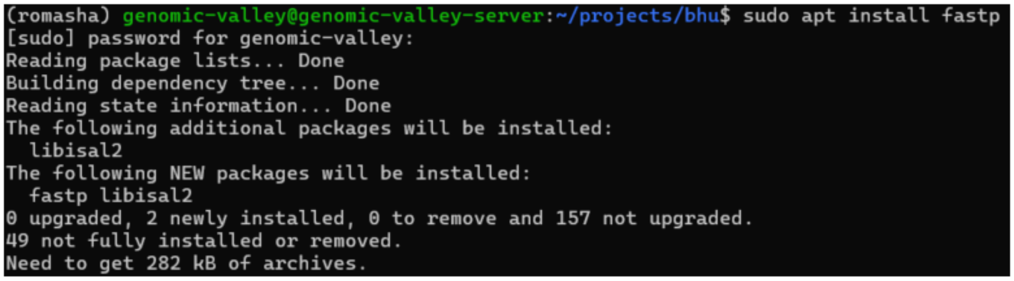
Step 1: Installation of Fastp

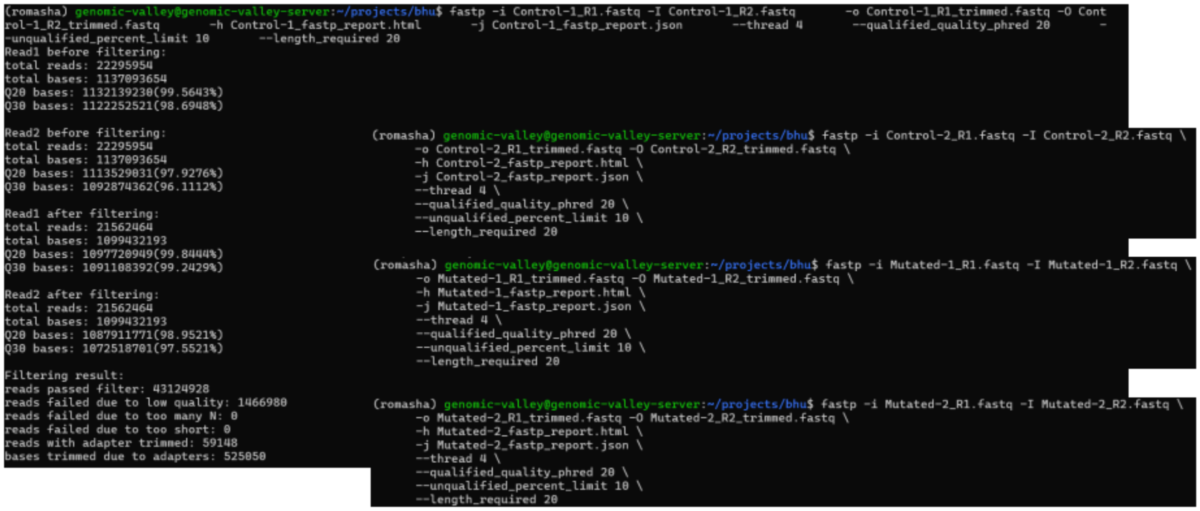


```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ sudo apt install fastp
[sudo] password for genomic-valley:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libisal2
The following NEW packages will be installed:
  fastp libisal2
0 upgraded, 2 newly installed, 0 to remove and 157 not upgraded.
49 not fully installed or removed.
Need to get 282 kB of archives.
```

Tool: Fastp
Command Used: sudo apt install fastp

Fastp is a fast, all-in-one preprocessing tool used for quality control and filtering of raw sequencing reads. It performs adapter trimming, quality filtering and base correction in a single step. The tool was installed using the apt package manager which ensures proper dependency handling. Successful installation confirms that the system is ready for read preprocessing.

Step 2: Quality Control and Read Trimming



```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ fastp -i Control-1_R1.fastq -I Control-1_R2.fastq       -o Control-1_R1_trimmed.fastq -O Cont
rol-1_R2_trimmed.fastq       -h Control-1_fastp_report.html       -j Control-1_fastp_report.json       --thread 4       --qualified_quality_phred 20       -
-unqualified_percent_limit 10       --length_required 20
Read1 before filtering:
total reads: 22295954
total bases: 1137093654
Q20 bases: 1132139230(99.5643%)
Q30 bases: 1122252521(98.6948%)

Read2 before filtering:
total reads: 22295954
total bases: 1137093654
Q20 bases: 1113529031(97.9276%)
Q30 bases: 1092874362(96.1112%)

Read1 after filtering:
total reads: 21562464
total bases: 1099432193
Q20 bases: 1097720949(99.8444%)
Q30 bases: 1091108392(99.2429%)

Read2 after filtering:
total reads: 21562464
total bases: 1099432193
Q20 bases: 1087911771(98.9521%)
Q30 bases: 1072518701(97.5521%)

Filtering result:
reads passed filter: 43124928
reads failed due to low quality: 1466980
reads failed due to too many N: 0
reads failed due to too short: 0
reads with adapter trimmed: 59148
bases trimmed due to adapters: 525050
```

```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ fastp -i Control-2_R1.fastq -I Control-2_R2.fastq \
      -o Control-2_R1_trimmed.fastq -O Control-2_R2_trimmed.fastq \
      -h Control-2_fastp_report.html \
      -j Control-2_fastp_report.json \
      --thread 4 \
      --qualified_quality_phred 20 \
      --unqualified_percent_limit 10 \
      --length_required 20
```

```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ fastp -i Mutated-1_R1.fastq -I Mutated-1_R2.fastq \
      -o Mutated-1_R1_trimmed.fastq -O Mutated-1_R2_trimmed.fastq \
      -h Mutated-1_fastp_report.html \
      -j Mutated-1_fastp_report.json \
      --thread 4 \
      --qualified_quality_phred 20 \
      --unqualified_percent_limit 10 \
      --length_required 20
```

```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ fastp -i Mutated-2_R1.fastq -I Mutated-2_R2.fastq \
      -o Mutated-2_R1_trimmed.fastq -O Mutated-2_R2_trimmed.fastq \
      -h Mutated-2_fastp_report.html \
      -j Mutated-2_fastp_report.json \
      --thread 4 \
      --qualified_quality_phred 20 \
      --unqualified_percent_limit 10 \
      --length_required 20
```

Tool: Fastp

Fastp was used to process paired-end FASTQ files. For example: Control_1_R1.fastq and Control_1_R2.fastq. It filters low-quality reads, trims adapters and removes bases below the quality threshold. A minimum read length of 20 and a quality score cutoff of 20 (Q20) were applied to ensure high-quality reads for downstream analysis.

Output generated:

- Trimmed FASTQ files
- HTML quality report
- JSON summary report
- Fastp provides a detailed summary of read quality both before and after filtering. The statistics include total reads, base quality (Q20/Q30), and GC content. After filtering, a higher percentage of high-quality bases was observed indicating successful removal of low-quality and adapter-contaminated sequences.

Here are the individual **fastp commands** for each sample pair:

Control-1:

```
fastp -i Control-1_R1.fastq -I Control-1_R2.fastq \
    -o Control-1_R1_trimmed.fastq -O Control-1_R2_trimmed.fastq \
    -h Control-1_fastp_report.html \
    -j Control-1_fastp_report.json \
    --thread 4 \
    --qualified_quality_phred 20 \
    --unqualified_percent_limit 10 \
    --length_required 20
```

Control-2:

```
fastp -i Control-2_R1.fastq -I Control-2_R2.fastq \
    -o Control-2_R1_trimmed.fastq -O Control-2_R2_trimmed.fastq \
    -h Control-2_fastp_report.html \
    -j Control-2_fastp_report.json \
    --thread 4 \
    --qualified_quality_phred 20 \
    --unqualified_percent_limit 10 \
    --length_required 20
```

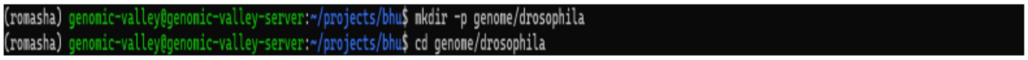
Mutated-1:

```
fastp -i Mutated-1_R1.fastq -I Mutated-1_R2.fastq \
    -o Mutated-1_R1_trimmed.fastq -O Mutated-1_R2_trimmed.fastq \
    -h Mutated-1_fastp_report.html \
    -j Mutated-1_fastp_report.json \
    --thread 4 \
    --qualified_quality_phred 20 \
    --unqualified_percent_limit 10 \
    --length_required 20
```

Mutated-2:

```
fastp -i Mutated-2_R1.fastq -I Mutated-2_R2.fastq \
    -o Mutated-2_R1_trimmed.fastq -O Mutated-2_R2_trimmed.fastq \
    -h Mutated-2_fastp_report.html \
    -j Mutated-2_fastp_report.json \
    --thread 4 \
    --qualified_quality_phred 20 \
    --unqualified_percent_limit 10 \
    --length_required 20
```

Step 3: Directory Organization

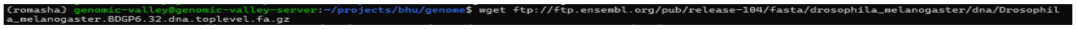


```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ mkdir -p genome/drosophila
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ cd genome/drosophila
```

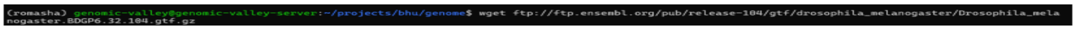
Command Used: mkdir -p genome/drosophila

A dedicated directory structure was created to organize genome analysis files systematically. Proper directory management helps maintain reproducibility and avoids file misplacement. All downstream analysis files related to Drosophila were stored in this directory.

Step 4: Downloaded Reference Files



```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu/genome$ wget ftp://ftp.ensembl.org/pub/release-104/fasta/drosophila_melanogaster/dna/Drosophil
a_melanogaster.BDGP6.32.dna.toplevel.fa.gz
```



```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu/genome$ wget ftp://ftp.ensembl.org/pub/release-104/gtf/drosophila_melanogaster/Drosophila_mela
nogaster.BDGP6.32.104.gtf.gz
```

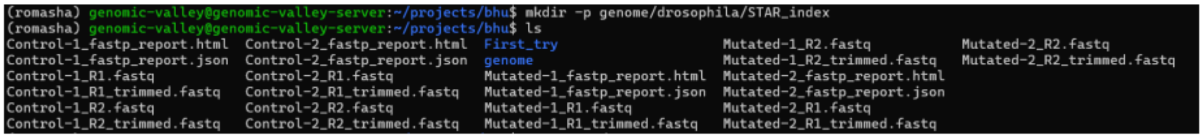
Downloaded: From Ensembl

Command used:
#Download the genome FASTA file
wget ftp://ftp.ensembl.org/pub/release-104/fasta/drosophila_melanogaster/dna/Drosophila_melanogaster.BDGP6.32.dna.toplevel.fa.gz

Download GTF annotation
wget ftp://ftp.ensembl.org/pub/release-104/gtf/drosophila_melanogaster/Drosophila_melanogaster.BDGP6.32.104.gtf.gz

Decompress the files
gunzip *.gz

Step 5: STAR (Genome Indexing)



```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ mkdir -p genome/drosophila/STAR_index
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ ls
Control-1_fastp_report.html  Control-2_fastp_report.html  First_try                    Mutated-1_R2.fastq           Mutated-2_R2.fastq
Control-1_fastp_report.json  Control-2_fastp_report.json  genome                       Mutated-1_R2_trimmed.fastq   Mutated-2_R2_trimmed.fastq
Control-1_R1.fastq           Control-2_R1.fastq           Mutated-1_fastp_report.html  Mutated-2_fastp_report.html
Control-1_R1_trimmed.fastq   Control-2_R1_trimmed.fastq   Mutated-1_fastp_report.json  Mutated-2_fastp_report.json
Control-1_R2.fastq           Control-2_R2.fastq           Mutated-1_R1.fastq           Mutated-2_R1.fastq
Control-1_R2_trimmed.fastq   Control-2_R2_trimmed.fastq   Mutated-1_R1_trimmed.fastq   Mutated-2_R1_trimmed.fastq
```

For Creating the STAR index directory
Command used:
mkdir -p genome/drosophila/STAR_index

Tool: STAR (Genome Indexing)

STAR genome indexing prepares the reference genome for fast and accurate RNA-seq read alignment. It builds suffix arrays and splice junction databases from the genome FASTA and annotation (GTF). Indexing is a mandatory pre-alignment step and is done only once per genome.

For Building STAR index:

Command used:

```
STAR --runMode genomeGenerate \
    --genomeDir genome/drosophila/STAR_index \
    --genomeFastaFiles genome/drosophila/Drosophila_melanogaster.BDGP6.32.dna.toplevel.fa\
    --sjdbGTFfile genome/drosophila/Drosophila_melanogaster.BDGP6.32.104.gtf \
    --sjdbOverhang 50 \
    --runThreadN 4
```

Step 6: STAR (RNA-seq Read Alignment & Gene Counting)

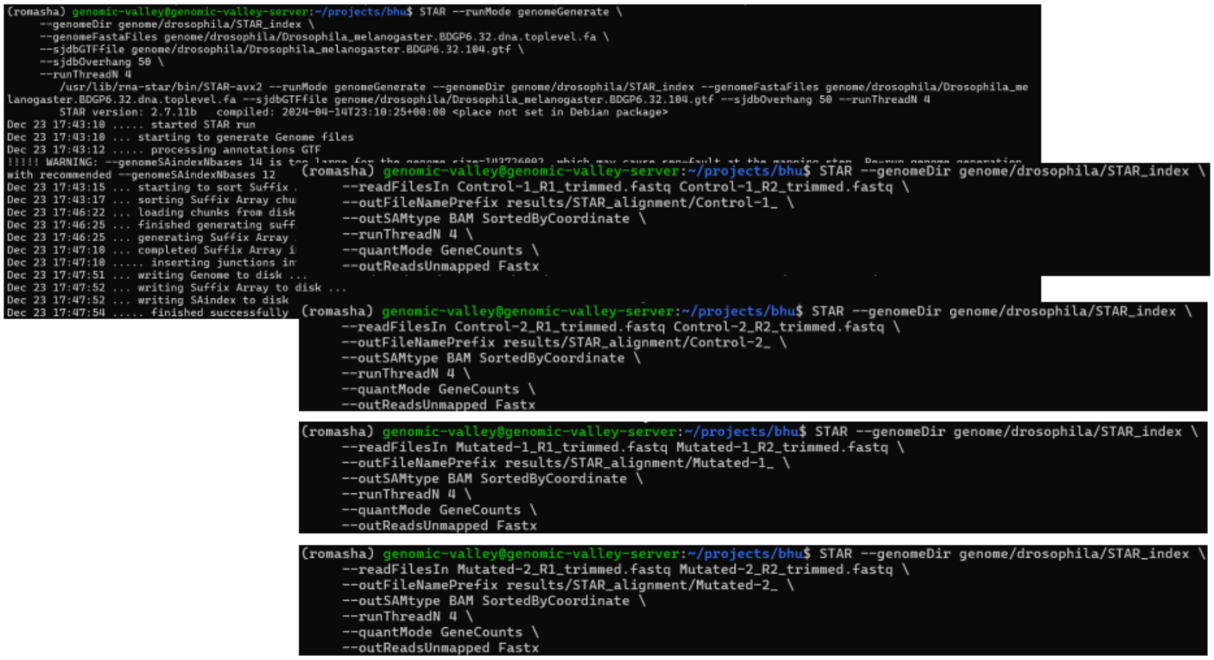
Tool: STAR (Alignment & Gene Counting)

STAR is used to align paired-end RNA-seq reads to the indexed reference genome. It performs splice-aware alignment, essential for eukaryotic transcriptome analysis. The **--runMode** GeneCounts option directly generates gene-level read counts. The output BAM file is sorted and ready for downstream analysis.

Key outputs:

Aligned.sortedByCoord.out.bam → aligned reads

ReadsPerGene.out.tab → raw gene counts for expression analysis



```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ STAR --runMode genomeGenerate \
     --genomeDir genome/drosophila/STAR_index \
     --genomeFastaFiles genome/drosophila/Drosophila_melanogaster.BDGP6.32.dna.toplevel.fa \
     --sjdbGTFfile genome/drosophila/Drosophila_melanogaster.BDGP6.32.104.gtf \
     --sjdbOverhang 50 \
     --runThreadN 4
        /usr/lib/rna-star/bin/STAR-avx2 --runMode genomeGenerate --genomeDir genome/drosophila/STAR_index --genomeFastaFiles genome/drosophila/Drosophila_me
lanogaster.BDGP6.32.dna.toplevel.fa --sjdbGTFfile genome/drosophila/Drosophila_melanogaster.BDGP6.32.104.gtf --sjdbOverhang 50 --runThreadN 4
        STAR version: 2.7.11b   compiled: 2024-04-14T23:10:25+00:00 <place not set in Debian package>
Dec 23 17:43:10 ..... started STAR run
Dec 23 17:43:10 ... starting to generate Genome files
Dec 23 17:43:12 ..... processing annotations GTF
!!!!! WARNING: --genomeSAindexNbases 14 is too large for the genome size=143726002, which may cause seg-fault at the mapping step. Re-run genome generation
with recommended --genomeSAindexNbases 12
Dec 23 17:43:15 ... starting to sort Suffix .
Dec 23 17:43:17 ... sorting Suffix Array chu
Dec 23 17:46:22 ... loading chunks from disk
Dec 23 17:46:25 ... finished generating suff
Dec 23 17:46:25 ... generating Suffix Array .
Dec 23 17:47:10 ... completed Suffix Array i
Dec 23 17:47:10 ..... inserting junctions in
Dec 23 17:47:51 ... writing Genome to disk ...
Dec 23 17:47:52 ... writing Suffix Array to disk ...
Dec 23 17:47:52 ... writing SAindex to disk
Dec 23 17:47:54 ..... finished successfully
```

```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ STAR --genomeDir genome/drosophila/STAR_index \
     --readFilesIn Control-1_R1_trimmed.fastq Control-1_R2_trimmed.fastq \
     --outFileNamePrefix results/STAR_alignment/Control-1_ \
     --outSAMtype BAM SortedByCoordinate \
     --runThreadN 4 \
     --quantMode GeneCounts \
     --outReadsUnmapped Fastx
```

```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ STAR --genomeDir genome/drosophila/STAR_index \
     --readFilesIn Control-2_R1_trimmed.fastq Control-2_R2_trimmed.fastq \
     --outFileNamePrefix results/STAR_alignment/Control-2_ \
     --outSAMtype BAM SortedByCoordinate \
     --runThreadN 4 \
     --quantMode GeneCounts \
     --outReadsUnmapped Fastx
```

```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ STAR --genomeDir genome/drosophila/STAR_index \
     --readFilesIn Mutated-1_R1_trimmed.fastq Mutated-1_R2_trimmed.fastq \
     --outFileNamePrefix results/STAR_alignment/Mutated-1_ \
     --outSAMtype BAM SortedByCoordinate \
     --runThreadN 4 \
     --quantMode GeneCounts \
     --outReadsUnmapped Fastx
```

```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ STAR --genomeDir genome/drosophila/STAR_index \
     --readFilesIn Mutated-2_R1_trimmed.fastq Mutated-2_R2_trimmed.fastq \
     --outFileNamePrefix results/STAR_alignment/Mutated-2_ \
     --outSAMtype BAM SortedByCoordinate \
     --runThreadN 4 \
     --quantMode GeneCounts \
     --outReadsUnmapped Fastx
```

Here's the **STAR alignment command** or each sample pair:

Control-1:

```
STAR --genomeDir genome/drosophila/STAR_index \
   --readFilesIn Control-1_R1_trimmed.fastq Control-1_R2_trimmed.fastq \
   --outFileNamePrefix results/STAR_alignment/Control-1_ \
   --outSAMtype BAM SortedByCoordinate \
   --runThreadN 4 \
   --quantMode GeneCounts \
   --outReadsUnmapped Fastx
```

Control-2:

```
STAR --genomeDir genome/drosophila/STAR_index \
   --readFilesIn Control-2_R1_trimmed.fastq Control-2_R2_trimmed.fastq \
   --outFileNamePrefix results/STAR_alignment/Control-2_ \
   --outSAMtype BAM SortedByCoordinate \
   --runThreadN 4 \
   --quantMode GeneCounts \
   --outReadsUnmapped Fastx
```

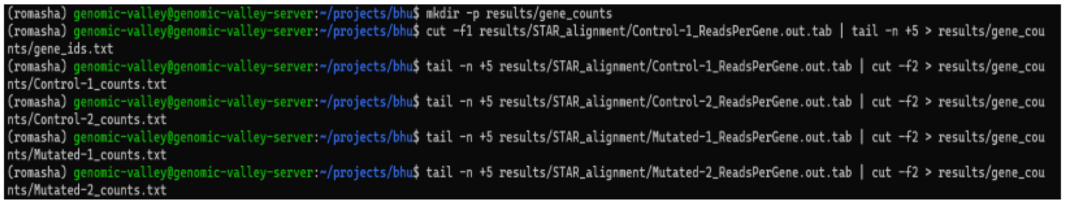
Mutated-1:

```
STAR --genomeDir genome/drosophila/STAR_index \
    --readFilesIn Mutated-1_R1_trimmed.fastq Mutated-1_R2_trimmed.fastq \
    --outFileNamePrefix results/STAR_alignment/Mutated-1_ \
    --outSAMtype BAM SortedByCoordinate \
    --runThreadN 4 \
    --quantMode GeneCounts \
    --outReadsUnmapped Fastx
```

Mutated-2:

```
STAR --genomeDir genome/drosophila/STAR_index \
    --readFilesIn Mutated-2_R1_trimmed.fastq Mutated-2_R2_trimmed.fastq \
    --outFileNamePrefix results/STAR_alignment/Mutated-2_ \
    --outSAMtype BAM SortedByCoordinate \
    --runThreadN 4 \
    --quantMode GeneCounts \
    --outReadsUnmapped Fastx
```

Step 7: Generating Count File



```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ mkdir -p results/gene_counts
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ cut -f1 results/STAR_alignment/Control-1_ReadsPerGene.out.tab | tail -n +5 > results/gene_cou
nts/gene_ids.txt
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ tail -n +5 results/STAR_alignment/Control-1_ReadsPerGene.out.tab | cut -f2 > results/gene_cou
nts/Control-1_counts.txt
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ tail -n +5 results/STAR_alignment/Control-2_ReadsPerGene.out.tab | cut -f2 > results/gene_cou
nts/Control-2_counts.txt
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ tail -n +5 results/STAR_alignment/Mutated-1_ReadsPerGene.out.tab | cut -f2 > results/gene_cou
nts/Mutated-1_counts.txt
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ tail -n +5 results/STAR_alignment/Mutated-2_ReadsPerGene.out.tab | cut -f2 > results/gene_cou
nts/Mutated-2_counts.txt
```

Tool: Linux Command-line Utilities (mkdir, cut, tail, paste, echo)

Linux text-processing tools are used to clean, format, and combine gene count files. They help remove unwanted rows, extract specific columns, and merge multiple samples. This step converts STAR outputs into a single count matrix required for downstream tools (DESeq2/edgeR). They are fast, reproducible, and ideal for large RNA-seq datasets.

Create results directory for counts

Command used:

```
mkdir -p results/gene_counts
```

First, get gene IDs from file

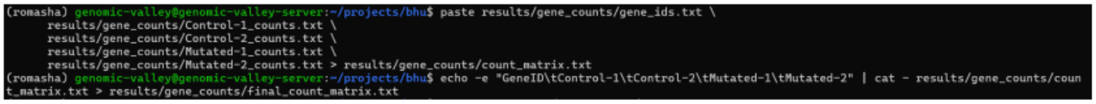
Command used:

```
cut -f1 results/STAR_alignment/Control-1_ReadsPerGene.out.tab | tail -n +5 > results/gene_counts/gene_ids.txt
```

Extract counts for each sample -

Command used:

```
tail -n +5 results/STAR_alignment/Control-1_ReadsPerGene.out.tab | cut -f2 > results/gene_counts/Control-1_counts.txt
tail -n +5 results/STAR_alignment/Control-2_ReadsPerGene.out.tab | cut -f2 > results/gene_counts/Control-2_counts.txt
tail -n +5 results/STAR_alignment/Mutated-1_ReadsPerGene.out.tab | cut -f2 > results/gene_counts/Mutated-1_counts.txt
tail -n +5 results/STAR_alignment/Mutated-2_ReadsPerGene.out.tab | cut -f2 > results/gene_counts/Mutated-2_counts.txt
```



```
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ paste results/gene_counts/gene_ids.txt \
      results/gene_counts/Control-1_counts.txt \
      results/gene_counts/Control-2_counts.txt \
      results/gene_counts/Mutated-1_counts.txt \
      results/gene_counts/Mutated-2_counts.txt > results/gene_counts/count_matrix.txt
(romasha) genomic-valley@genomic-valley-server:~/projects/bhu$ echo -e "GeneID\tControl-1\tControl-2\tMutated-1\tMutated-2" | cat - results/gene_counts/coun
t_matrix.txt > results/gene_counts/final_count_matrix.txt
```

This step combines individual gene count files from all samples into a single count matrix. Each row represents a gene, and each column corresponds to a biological condition or replicate. Adding a proper header ensures compatibility with downstream differential expression tools. The final matrix serves as the primary input for RNA-seq statistical analysis (DESeq2/edgeR).

Combine into final count matrix
Command used:

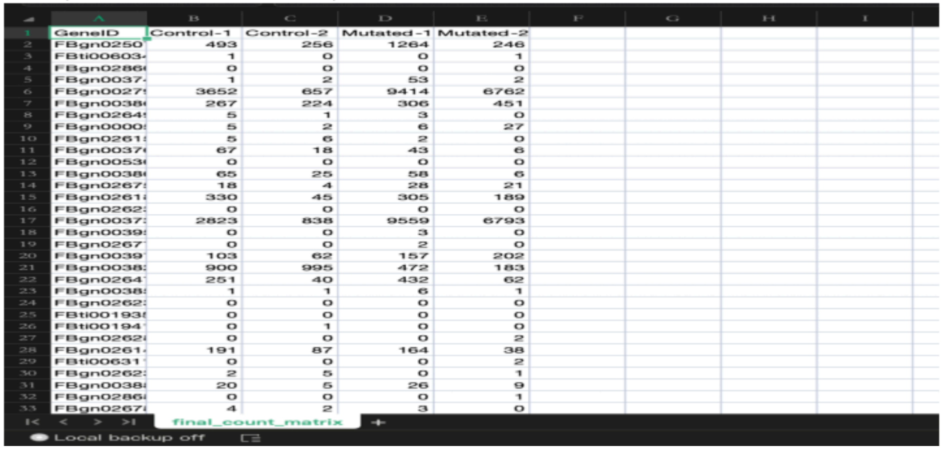
```
paste results/gene_counts/gene_ids.txt \
    results/gene_counts/Control-1_counts.txt \
    results/gene_counts/Control-2_counts.txt \
    results/gene_counts/Mutated-1_counts.txt \
    results/gene_counts/Mutated-2_counts.txt > results/gene_counts/count_matrix.txt
```

Add header
Command used:

```
echo -e "GeneID\tControl-1\tControl-2\tMutated-1\tMutated-2" | cat - results/gene_counts/count_matrix.txt > 
results/gene_counts/final_count_matrix.txt
```

Step 8 : Edit & finalize the count matrix

The merged gene count data were curated to generate a final count matrix suitable for differential expression analysis. Unnecessary columns were removed, sample names were standardized and a unified header was added.
Genes with zero counts across all samples were optionally filtered out to improve statistical robustness. The finalized count matrix was used as input for DESeq2-based differential gene expression analysis.
The final count matrix includes gene identifiers in the first column followed by raw read counts for each biological sample. It contains four sample columns representing two control replicates (Control-1 and Control-2) and two mutated replicates (Mutated-1 and Mutated-2).



GeneID	Control-1	Control-2	Mutated-1	Mutated-2
FBgn0250	493	256	1264	246
FBti00603	1	0	0	1
FBgn0286	0	0	0	0
FBgn0037	1	2	53	2
FBgn0027	3652	657	9414	6762
FBgn0038	267	224	306	451
FBgn0264	5	1	3	0
FBgn0000	5	2	6	27
FBgn0261	5	6	2	0
FBgn0037	67	18	43	6
FBgn0053	0	0	0	0
FBgn0038	65	25	58	6
FBgn0267	18	4	28	21
FBgn0261	330	45	305	189
FBgn0262	0	0	0	0
FBgn0037	2823	838	9559	6793
FBgn0039	0	0	3	0
FBgn0267	0	0	2	0
FBgn0039	103	62	157	202
FBgn0038	900	995	472	183
FBgn0264	251	40	432	62
FBgn0038	1	1	6	1
FBgn0262	0	0	0	0
FBti00193	0	0	0	0
FBti00194	0	1	0	0
FBgn0262	0	0	0	2
FBgn0261	191	87	164	38
FBti00631	0	0	0	2
FBgn0262	2	5	0	1
FBgn0038	20	5	26	9
FBgn0286	0	0	0	1
FBgn0267	4	2	3	0