Rna –seq analysis (upstream analysis) plant samples (Arabidopsis thaline) from fastq file to feature count

Mohamad sharawy Ibrahim

Bachelor in agriculture biotechnology ,Cairo university

EGCOMBIO DIPLOM STUDENT

-agenda

- Upstream analysis !!
- Rna –seq !
- Samples and methods
- Pipeline (tools)
- Script
- Result
- Questions !?

Upstream analysis

refers to the initial stages of data processing and analysis, particularly in contexts like bioinformatics

Data Acquisition: Collecting raw data from various sources, such as sequencing machines in genomics, sensors, or databases.

Data Cleaning: Removing noise, errors, or irrelevant information from the raw data. This could involve filtering out low-quality data, handling missing values, or correcting errors.

Data Preprocessing: Transforming the raw data into a format suitable for analysis. In bioinformatics, this might involve aligning sequences to a reference genome, normalizing data, or converting data into a usable format (e.g., FASTQ to BAM files in different fields in computational biology)

RNA-seq experiment

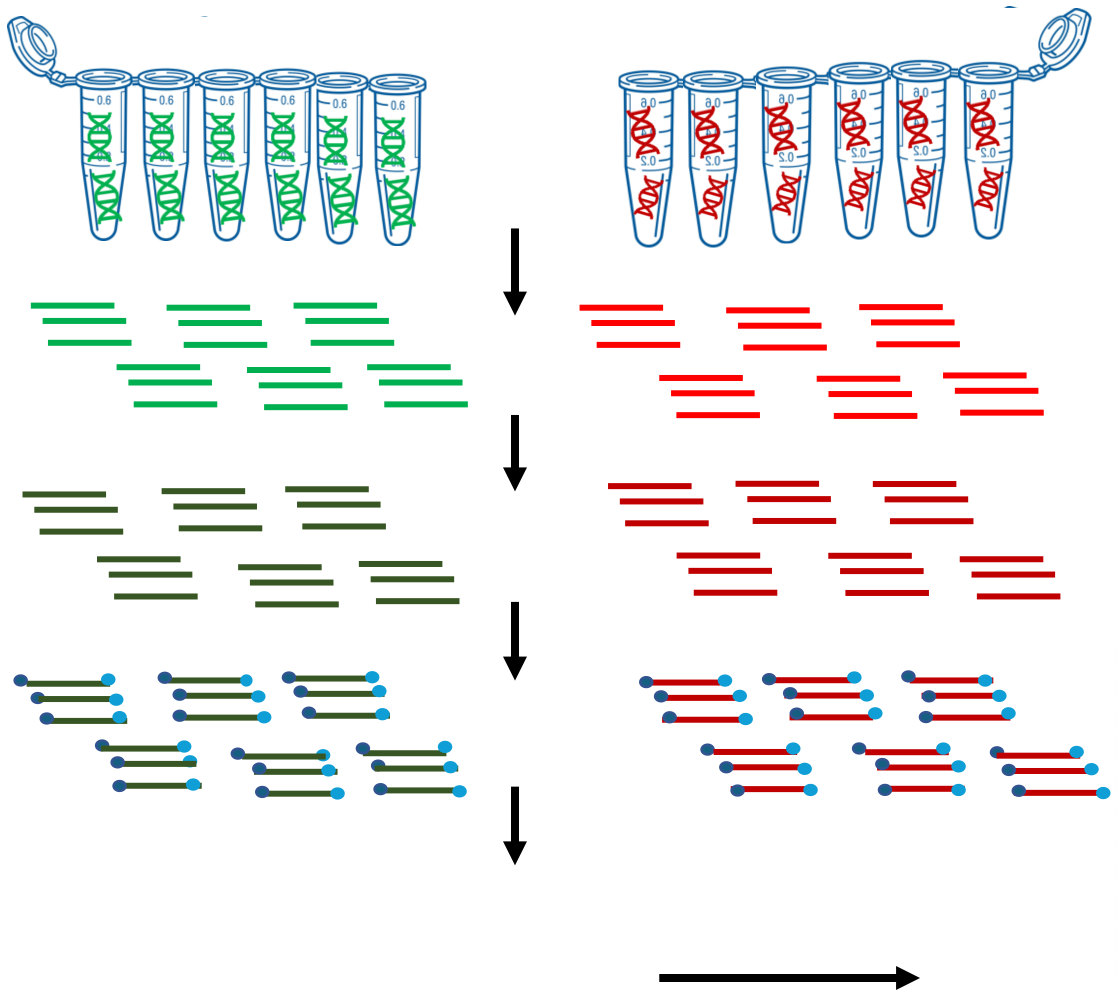
Library preparation

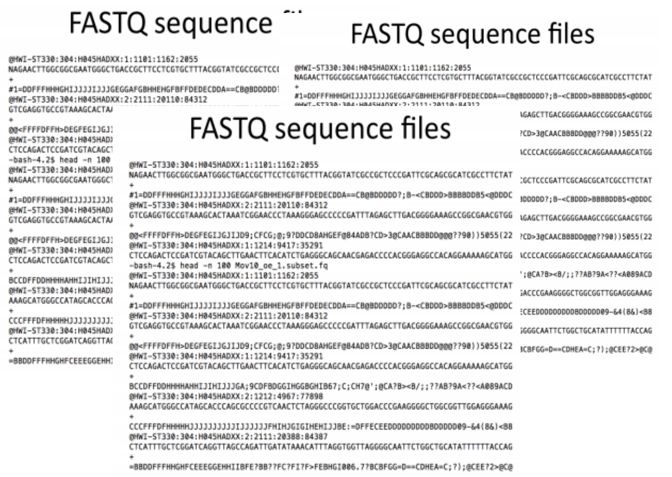
mRNA

mRNA fragmentation

Adaptors ligation

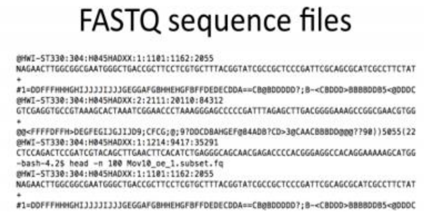
High throughput sequencer




FASTQ sequence files

```
@HWI-ST330:304:H045HADXX:1:1101:1162:2055
NAGAACTTGGCGGCGAATGGGCTGACCGCTTCCTCGTGCTTTACGGTATCGCCGCTCCCGATTCGCAGCGCATCGCCTTCTAT
+
#1=DDFFFHHHGHIJJJJIJJJGEGGAFGBHHEHGFBFFDEDECDDA==CB@BDDDDD?;B-<CBDDD>BBBBDDB5<@DDDC
@HWI-ST330:304:H045HADXX:2:2111:20110:84312
GTCGAGGTGCCGTAAAGCACTAAATCGGAACCCTAAAGGGAGCCCCCGATTTAGAGCTTGACGGGGAAAGCCGGCGAACGTGG
+
@@<FFFFDFFH>DEGFEGIJGJIJD9;CFCG;@;9?DDCDBAHGEF@B4ADB?CD>3@CAACBBBDD@@@??90))5055(22
@HWI-ST330:304:H045HADXX:1:1214:9417:35291
CTCCAGACTCCGATCGTACAGCTTGAACTTCACATCTGAGGGCAGCAACGAGACCCCACGGGAGGCCACAGGAAAAAGCATGG
-bash-4.2$ head -n 100 Mov10_oe_1.subset.fq
@HWI-ST330:304:H045HADXX:1:1101:1162:2055
NAGAACTTGGCGGCGAATGGGCTGACCGCTTCCTCGTGCTTTACGGTATCGCCGCTCCCGATTCGCAGCGCATCGCCTTCTAT
+
#1=DDFFFHHHGHIJJJJIJJJGEGGAFGBHHEHGFBFFDEDECDDA==CB@BDDDDD?;B-<CBDDD>BBBBDDB5<@DDDC
@HWI-ST330:304:H045HADXX:2:2111:20110:84312
GTCGAGGTGCCGTAAAGCACTAAATCGGAACCCTAAAGGGAGCCCCCGATTTAGAGCTTGACGGGGAAAGCCGGCGAACGTGG
+
@@<FFFFDFFH>DEGFEGIJGJIJD9;CFCG;@;9?DDCDBAHGEF@B4ADB?CD>3@CAACBBBDD@@@??90))5055(22
@HWI-ST330:304:H045HADXX:1:1214:9417:35291
CTCCAGACTCCGATCGTACAGCTTGAACTTCACATCTGAGGGCAGCAACGAGACCCCACGGGAGGCCACAGGAAAAAGCATGG
+
BCCDFFDDHHHHAHHIJIHIJJJGA;9CDFBDGGIHGGBGHIB67;C;CH7@';@CA?B><B/;;??AB?9A<??<A089ACD
@HWI-ST330:304:H045HADXX:2:1212:4967:77898
AAAGCATGGGCCATAGCACCCAGCGCCCCGTCAACTCTAGGGCCCGGTGCTGGACCCGAAGGGGCTGGCGGTTGGAGGGAAAG
+
CCCFFFDFHHHHHJJJJJJJJJJIJJJJJJFHIHJGIGIHEHIJJBE:=DFFECEEDDDDDDDDDBDDDDD09-&4(8&)<BB
@HWI-ST330:304:H045HADXX:2:2111:20388:84387
CTCATTTGCTCGGATCAGGTTAGCCAGATTGATATAAACATTTAGGTGGTTAGGGGCAATTCTGGCTGCATATTTTTTACCAG
+
=BBDDFFFHHGHFCEEEGGEHHIIBFE?BB??FC?FI?F>FEBHGI006.7?BCBFGG=D==CDHEA=C;?);@CEE?2>@C@
```

RNA-seq differential expression workflow



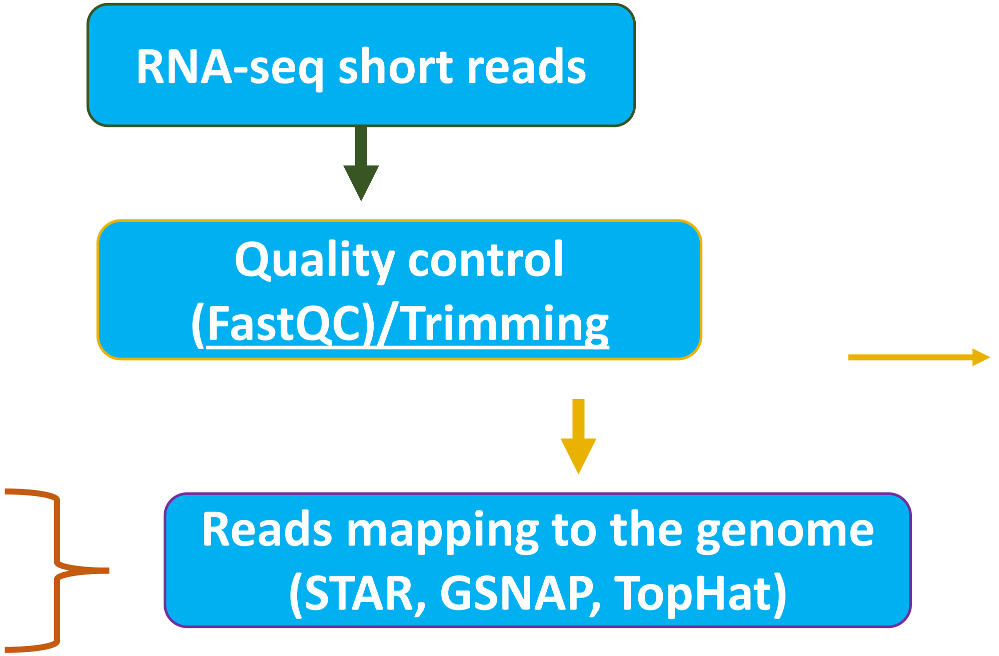
FASTQ sequence files

```
@HWI-ST330:304:H045HADXX:1:1101:1162:2055
NAGAACTTGGCGGCGAATGGGCTGACCGCTTCCTCGTGCTTTACGGTATCGCCGCTCCCGATTCGCAGCGCATCGCCTTCTAT
+
#1=DDFFFHHHGHIJJJJIJJJGEGGAFGBHHEHGFBFFDEDECDDA==CB@BDDDDD?;B-<CBDDD>BBBBDDB5<@DDDC
@HWI-ST330:304:H045HADXX:2:2111:20110:84312
GTCGAGGTGCCGTAAAGCACTAAATCGGAACCCTAAAGGGAGCCCCCGATTTAGAGCTTGACGGGGAAAGCCGGCGAACGTGG
+
@@<FFFFDFFH>DEGFEGIJGJIJD9;CFCG;@;9?DDCDBAHGEF@B4ADB?CD>3@CAACBBBDD@@@??9B))5055(22
@HWI-ST330:304:H045HADXX:1:1214:9417:35291
CTCCAGACTCCGATCGTACAGCTTGAACTTCACATCTGAGGGCAGCAACGAGACCCCACGGGAGGCCACAGGAAAAAGCATGG
-bash-4.2$ head -n 100 Mov10_oe_1.subset.fq
@HWI-ST330:304:H045HADXX:1:1101:1162:2055
NAGAACTTGGCGGCGAATGGGCTGACCGCTTCCTCGTGCTTTACGGTATCGCCGCTCCCGATTCGCAGCGCATCGCCTTCTAT
+
#1=DDFFFHHHGHIJJJJIJJJGEGGAFGBHHEHGFBFFDEDECDDA==CB@BDDDDD?;B-<CBDDD>BBBBDDB5<@DDDC
```

reference genome sequences (FASTA files)

annotations (GTF file)

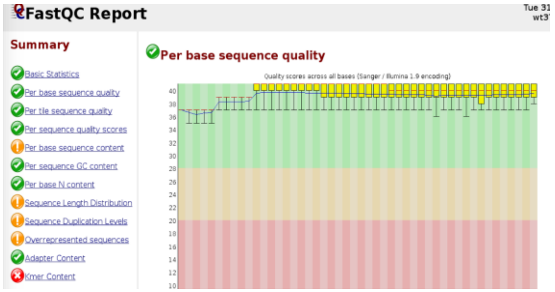
FASTQ sequence reads



RNA-seq short reads

Quality control (FastQC)/Trimming

Reads mapping to the genome (STAR, GSNAP, TopHat)



FastQC Report

Tue 31
wt3'

Summary

- ✓ Basic Statistics
- ✓ Per base sequence quality
- ✓ Per tile sequence quality
- ✓ Per sequence quality scores
- ! Per base sequence content
- ✓ Per sequence GC content
- ✓ Per base N content
- ! Sequence Length Distribution
- ! Sequence Duplication Levels
- ! Overrepresented sequences
- ✓ Adapter Content
- ✗ Kmer Content

✓ **Per base sequence quality**

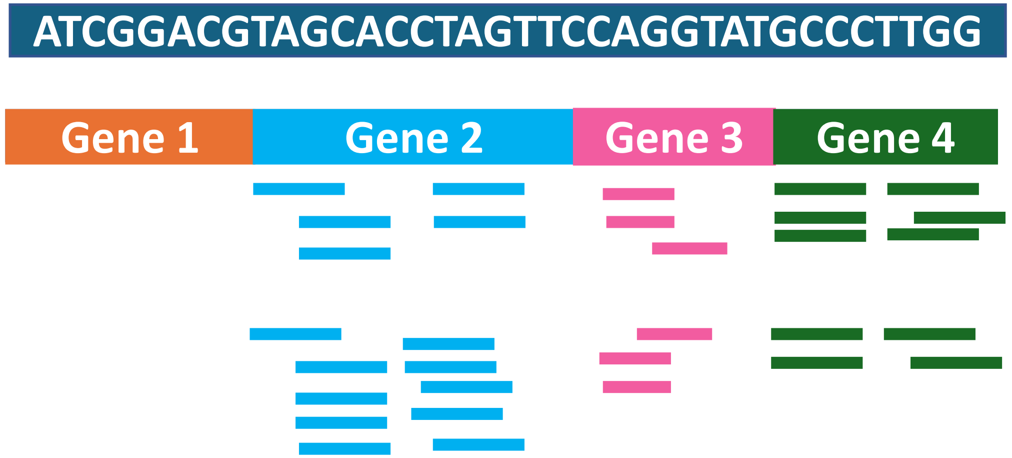
Quality scores across all bases (Sanger / Illumina 1.9 encoding)

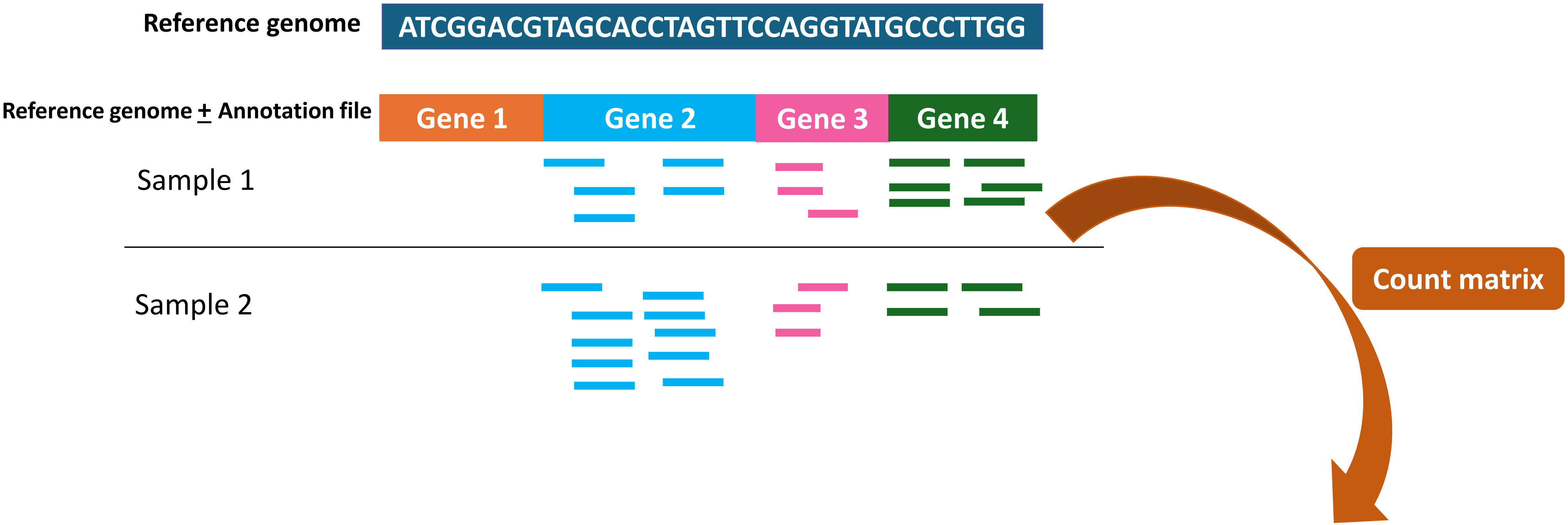
Reference genome

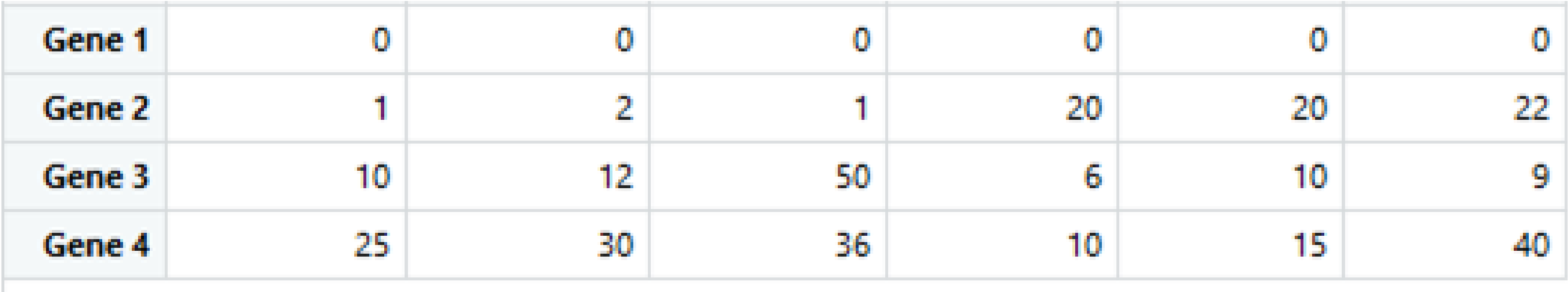
Reference genome + Annotation file

Sample 1

Sample 2




Reference genome
ATCGGACGTAGCACCTAGTTCCAGGTATGCCCTTGG
Reference genome ± Annotation file
Gene 1
Gene 2
Gene 3
Gene 4
Sample 1
Sample 2
Count matrix



	Control1	Control2	Control3	SC2_1	SC2_2	SC2_3
Gene 1	0	0	0	0	0	0
Gene 2	1	2	1	20	20	22
Gene 3	10	12	50	6	10	9
Gene 4	25	30	36	10	15	40

<u>Samples and methods</u>

Arabidopsis thaliana, a small flowering plant widely used as a model organism in plant biology and genetics. When analyzing *Arabidopsis* samples, particularly in genomics or transcriptomics, the upstream analysis might include specific steps tailored to this plant.

Upstream Analysis for *Arabidopsis* Samples

Sample Preparation:

1. Collection: Harvesting tissue samples (like leaves) from *Arabidopsis* plants.
2. RNA Extraction: Isolate high-quality total RNA from the collected tissues, ensuring the RNA is intact and free of contaminants (e.g., DNA, proteins)..

RNA Sequencing (RNA-Seq):

•**Library Preparation**: Convert RNA into a sequencing library, typically by reverse transcribing RNA into cDNA, fragmenting the cDNA, and adding sequencing adapters.

•**Sequencing**: Perform high-throughput sequencing using platforms like Illumina to generate raw RNA-Seq reads. The sequencing can be single-end or paired-end, depending on the study design.

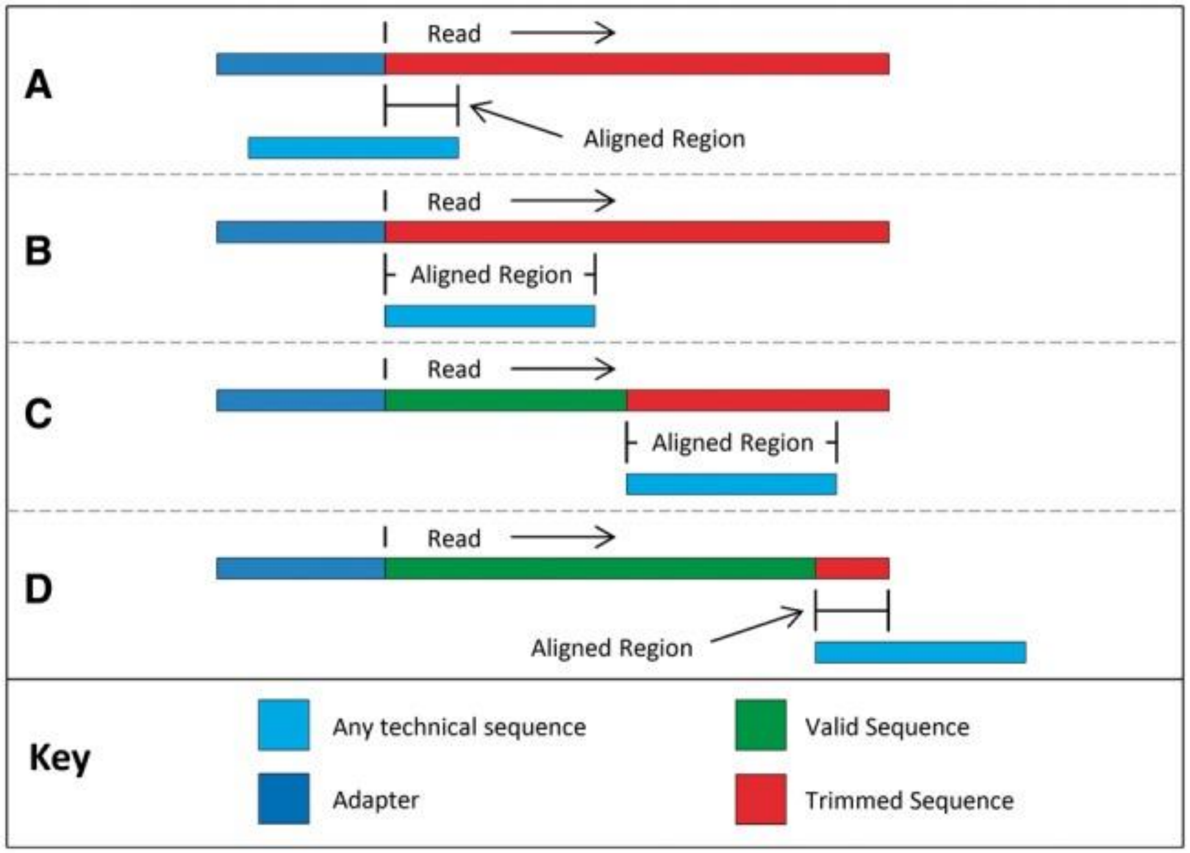
Tools

Quality Control of Raw Reads:

- **Read Quality Assessment**: Use tools like FastQC to assess the quality of raw sequencing reads. This includes checking for factors such as base quality scores, GC content, and adapter contamination.

- **Trimming and Filtering**: Trim low-quality bases and remove adapter sequences using tools like Trimmomatic or Cutadapt. Filter out low-quality reads to ensure high-quality data for subsequent analysis.

-
```
java -jar trimmomatic-0.35.jar SE -phred33 input.fq.gz output.fq.gz ILLUMINACLIP:TruSeq3-SE:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36
```



Indexing and Alignment:

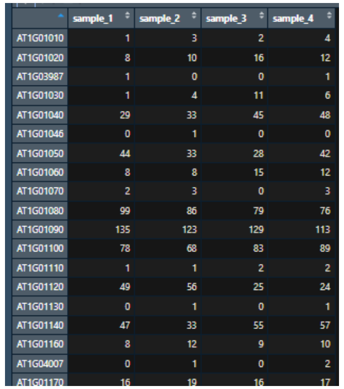
•**Mapping to the *Arabidopsis* Reference Genome**: Align the cleaned reads to the *Arabidopsis thaliana* reference genome (e.g., TAIR10) using aligners such as HISAT2, STAR, or Bowtie2.

•First indexing

STAR --runThreadN 8 \ --runMode genomeGenerate \ --genomeDir /path/to/output/genomeIndex \ --genomeFastaFiles /path/to/reference/genome.fasta \ --sjdbGTFfile /path/to/annotation/file.gtf \ --sjdbOverhang 100

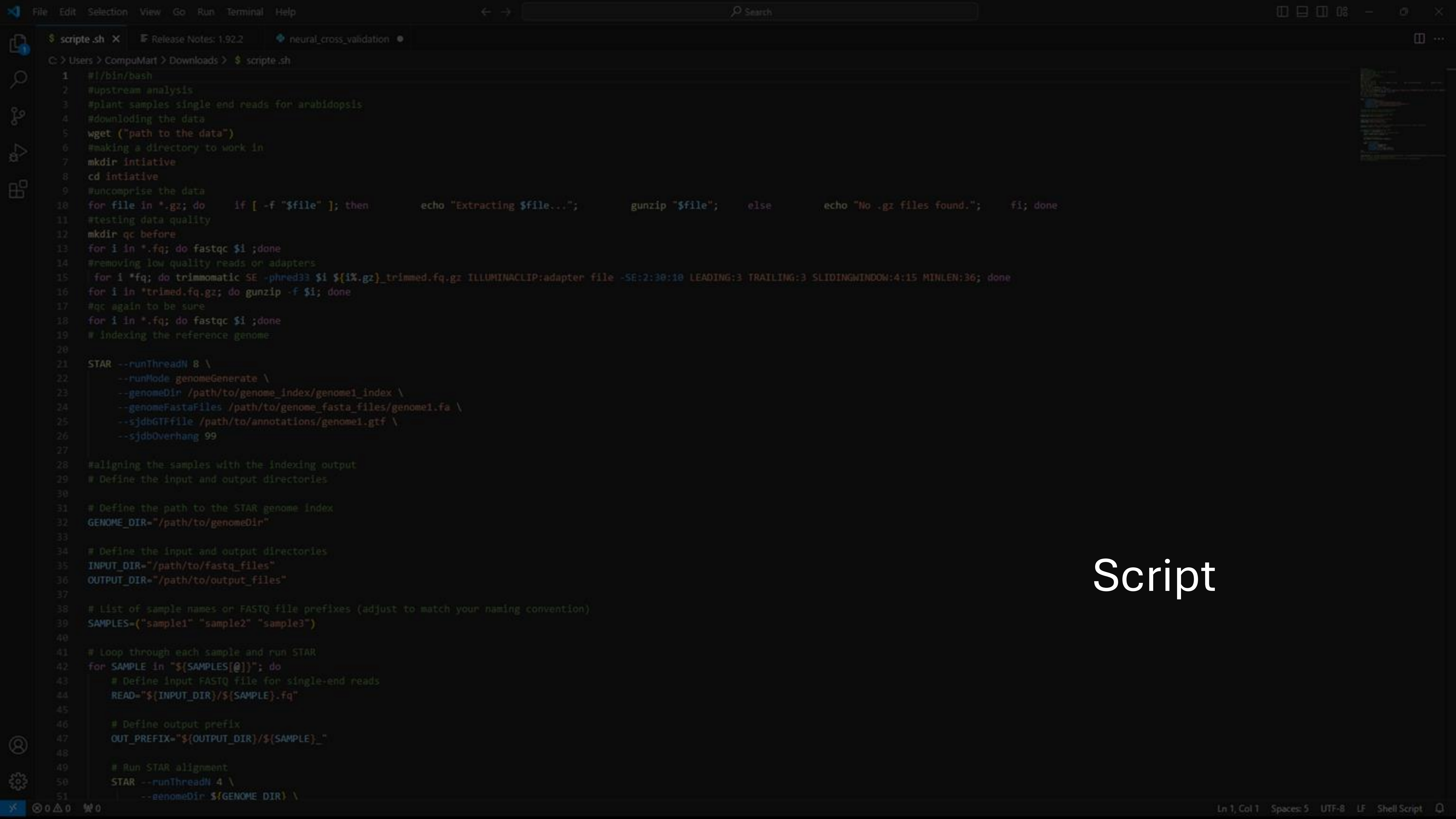
•--runThreadN 8: Specifies the number of threads to use for faster processing.
•--runMode genomeGenerate: Tells STAR to run in genome indexing mode.
•--genomeDir: Specifies the directory where the genome index files will be saved.
•--genomeFastaFiles: Points to the reference genome FASTA file.
•--sjdbGTFfile: Points to the annotation GTF file.
•--sjdbOverhang 100: Sets the length of the genomic sequence around annotated
•Second alignment
•STAR --runThreadN 8 \ --genomeDir /path/to/genomeIndex \ --readFilesIn /path/to/read.fastq \ --outFileNamePrefix /path/to/output/sample_name \ --outSAMtype BAM SortedByCoordinate \
•runThreadN 8: Number of threads for parallel processing. Adjust based on your CPU availability.
•--genomeDir: Directory containing the STAR genome index files.
•--readFilesIn: Input FASTQ files for the reads. If using paired-end reads, specify both files; otherwise, just one file for single-end reads.
•--outFileNamePrefix: Prefix for output files, including the directory path.
•--outSAMtype BAM SortedByCoordinate: Outputs the alignment in BAM format, sorted by genomic coordinates, which is useful for downstream processing.

- **Quantification**:
- **Gene/Transcript Quantification**: Quantify the abundance of each gene or transcript by counting the aligned reads. This is typically done using tools like featureCounts, HTSeq, or by using transcript abundance estimation tools like Salmon or Kallisto.
- featureCounts
- featureCounts -T 8 \ -a /path/to/annotation.gtf \ -o counts.txt \ /path/to/aligned_sample1.bam /path/to/aligned_sample2.bam ...



	sample_1	sample_2	sample_3	sample_4
AT1G01010	1	3	2	4
AT1G01020	8	10	16	12
AT1G03987	1	0	0	1
AT1G01030	1	4	11	6
AT1G01040	29	33	45	48
AT1G01046	0	1	0	0
AT1G01050	44	33	28	42
AT1G01060	8	8	15	12
AT1G01070	2	3	0	3
AT1G01080	99	86	79	76
AT1G01090	135	123	129	113
AT1G01100	78	68	83	89
AT1G01110	1	1	2	2
AT1G01120	49	56	25	24
AT1G01130	0	1	0	1
AT1G01140	47	33	55	57
AT1G01160	8	12	9	10
AT1G04007	0	1	0	2
AT1G01170	16	19	16	17

Script

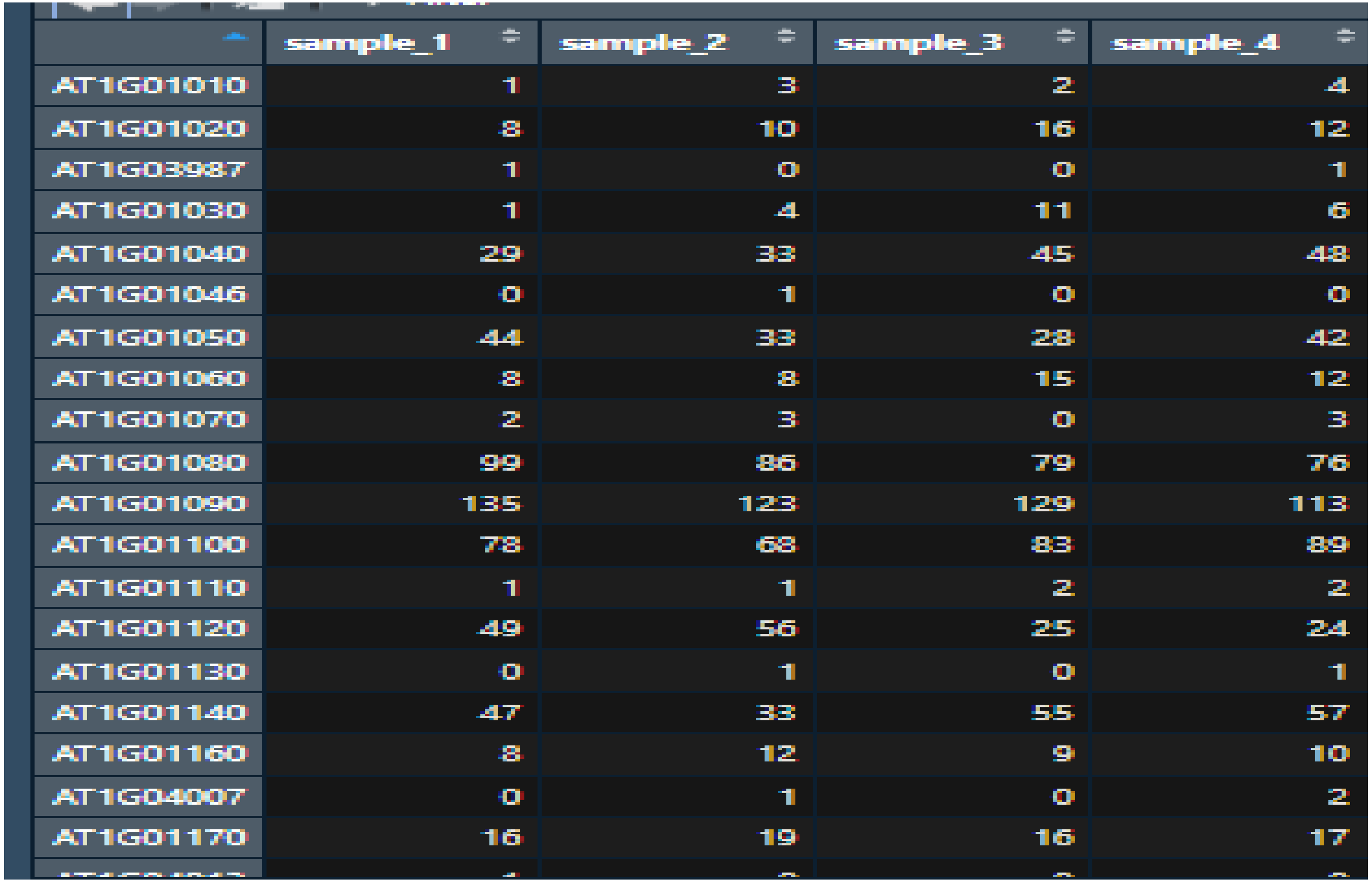


File Edit Selection View Go Run Terminal Help
Search
scripte.sh
Release Notes: 1.92.2
neural_cross_validation
C: > Users > CompuMart > Downloads > scripte.sh
⊗ 0 ⚠ 0 ⚡ 0
Ln 1, Col 1 Spaces: 5 UTF-8 LF Shell Script

```
#!/bin/bash
#upstream analysis
#plant samples single end reads for arabidopsis
#downloding the data
wget ("path to the data")
#making a directory to work in
mkdir intiative
cd intiative
#uncomprise the data
for file in *.gz; do     if [ -f "$file" ]; then         echo "Extracting $file...";         gunzip "$file";     else         echo "No .gz files found.";     fi; done
#testing data quality
mkdir qc before
for i in *.fq; do fastqc $i ;done
#removing low quality reads or adapters
 for i *fq; do trimmomatic SE -phred33 $i ${i%.gz}_trimmed.fq.gz ILLUMINACLIP:adapter file -SE:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36; done
for i in *trimed.fq.gz; do gunzip -f $i; done
#qc again to be sure
for i in *.fq; do fastqc $i ;done
# indexing the reference genome

STAR --runThreadN 8 \
     --runMode genomeGenerate \
     --genomeDir /path/to/genome_index/genome1_index \
     --genomeFastaFiles /path/to/genome_fasta_files/genome1.fa \
     --sjdbGTFfile /path/to/annotations/genome1.gtf \
     --sjdbOverhang 99

#aligning the samples with the indexing output
# Define the input and output directories

# Define the path to the STAR genome index
GENOME_DIR="/path/to/genomeDir"

# Define the input and output directories
INPUT_DIR="/path/to/fastq_files"
OUTPUT_DIR="/path/to/output_files"

# List of sample names or FASTQ file prefixes (adjust to match your naming convention)
SAMPLES=("sample1" "sample2" "sample3")

# Loop through each sample and run STAR
for SAMPLE in "${SAMPLES[@]}"; do
    # Define input FASTQ file for single-end reads
    READ="${INPUT_DIR}/${SAMPLE}.fq"

    # Define output prefix
    OUT_PREFIX="${OUTPUT_DIR}/${SAMPLE}_"

    # Run STAR alignment
    STAR --runThreadN 4 \
         --genomeDir ${GENOME_DIR} \
```

Results

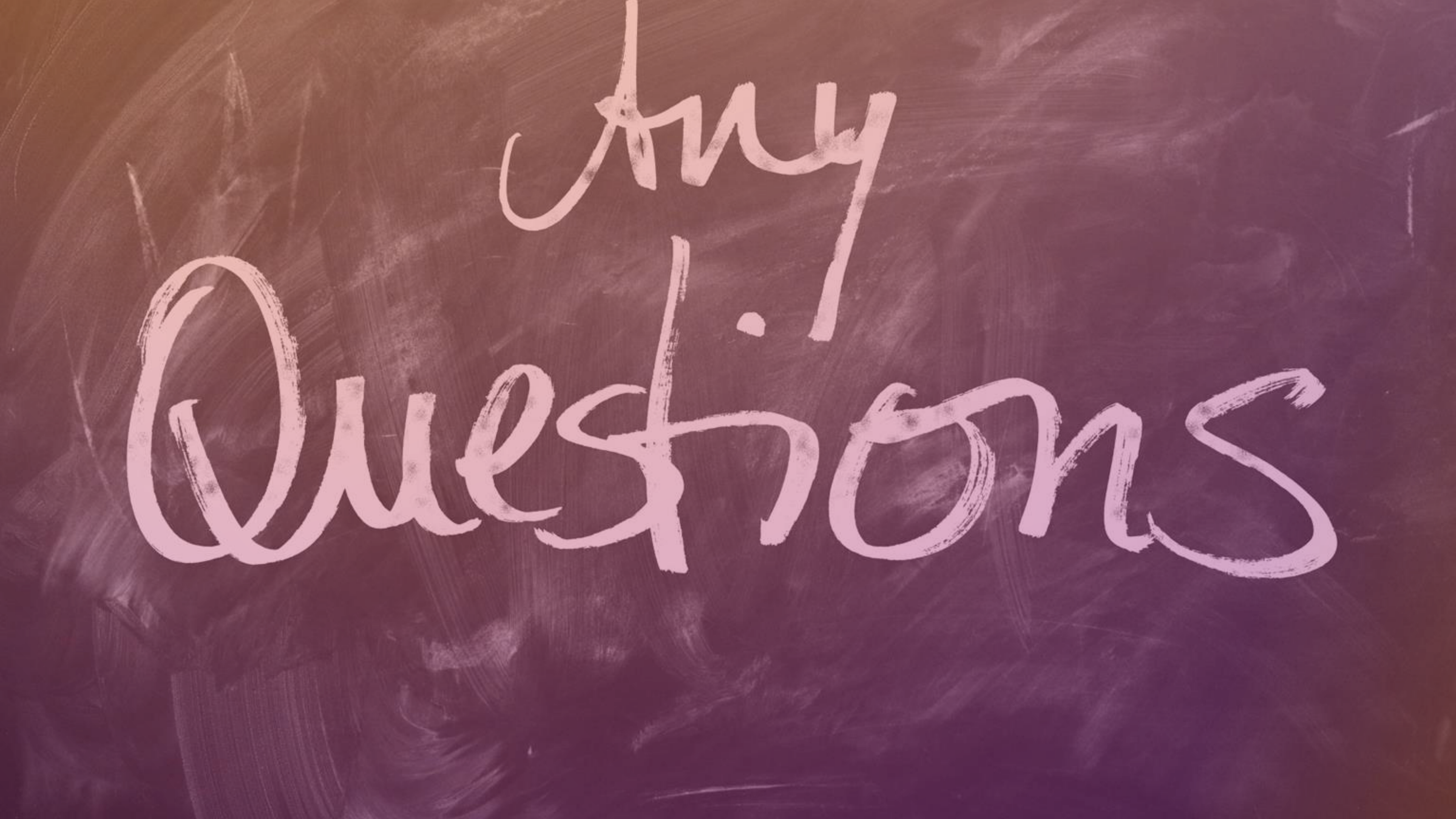
Cleaned feature matrix explain the appendance of deferent expression of genes in different samples



	sample_1	sample_2	sample_3	sample_4
AT1G01010	1	3	2	4
AT1G01020	8	10	16	12
AT1G03987	1	0	0	1
AT1G01030	1	4	11	6
AT1G01040	29	33	45	48
AT1G01046	0	1	0	0
AT1G01050	44	33	28	42
AT1G01060	8	8	15	12
AT1G01070	2	3	0	3
AT1G01080	99	86	79	76
AT1G01090	135	123	129	113
AT1G01100	78	68	83	89
AT1G01110	1	1	2	2
AT1G01120	49	56	25	24
AT1G01130	0	1	0	1
AT1G01140	47	33	55	57
AT1G01160	8	12	9	10
AT1G04007	0	1	0	2
AT1G01170	16	19	16	17

Reference

- Trimmomatic: A Flexible Trimmer for Illumina Sequence Data (bioinformaticschool.com)
- UTAP: User-friendly Transcriptome Analysis Pipeline - PubMed (nih.gov)
- Alignment with STAR | Introduction to RNA-Seq using high-performance computing - ARCHIVED (hbctraining.github.io)
- FeatureCounts - Bioinformatics Notebook (rnnh.github.io)
- UTAP: User-friendly Transcriptome Analysis Pipeline | BMC Bioinformatics | Full Text (biomedcentral.com)
- bio0202-Members of the Multinational Arabidopsis Steering Committee (nsf.gov)


Any
Questions