LINUX

Windows Subsystem for Linux (WSL)

Watch YouTube of WSL Ubuntu

Install via Microsoft Store: Ubuntu → Launch → Terminal-based.

All code/syntax/command, scripts are present in CODE.txt

https://github.com/Shouryanpatil/A-Guide-to-Basic-RNA-Seq

Download SRA Toolkit

YouTube Video - https://www.youtube.com/watch?v=E1n-Z2HDAD0

Bash code

```
mkdir SRA_TOOLKIT

cd SRA_TOOLKIT

wget --output-document sratoolkit.tar.gz https://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/current/sratoolkit.current-ubuntu64.tar.gz

tar -vxzf sratoolkit.tar.gz

export PATH=$PWD/sratoolkit.3.0.0-ubuntu64/bin

which fastq-dump

# extra export PATH=/usr/bin:/bin:$PATH

# extra source ~/.bashrc

which fastq-dump

vdb-config -i # Set directory want to data to download

# Mine was /mnt/e/SRA_TOOLKIT
```

Conda

YouTube Video - https://www.youtube.com/watch?v=AshsPB3KT-E

Bash code

```
wget https://repo.anaconda.com/archive/Anaconda3-2024.10-1-Linux-x86_64.sh
```

```
ls

chmod +X Anaconda3-2024.10-1-Linux-x86_64.sh

ls

./Anaconda3-2024.10-1-Linux-x86_64.sh

  >>> Press Enter
```

```
yes
 Enter
yes
conda config --set auto_activate_base false


conda config --show channels
conda config --add channels conda-forge
conda config --add channels bioconda
conda config --show channels


conda env list
conda create -n bioinformatics


clear
```

FastQC, Trimmomatic, HISAT2, Samtools, featureCounts

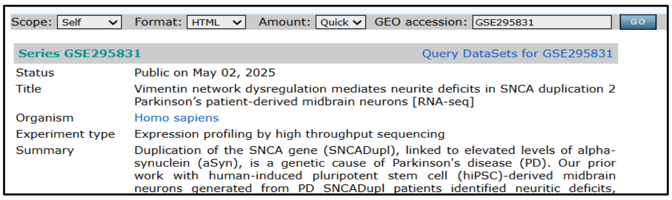
Bash

```
conda activate bioinformatics


conda --version
# verify conda is installed.
conda update conda
# update conda to latest version.


y


conda install -y -c bioconda fastqc trimmomatic hisat2 samtools subread
conda install -c conda-forge libgcc-ng
```

1. Download Sample Data

GSE295831 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE295831)

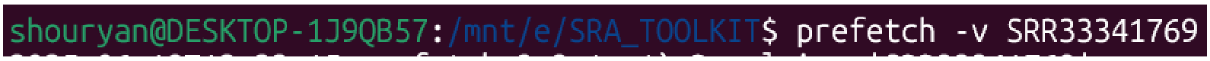
https://www.ncbi.nlm.nih.gov/Traces/study/?acc=PRJNA1256454&o=acc_s%3Aa



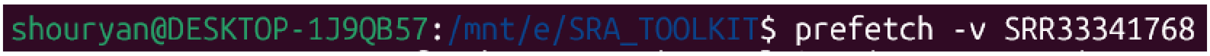
Scope: Self Format: HTML Amount: Quick GEO accession: GSE295831 GO

Series GSE295831	Query DataSets for GSE295831
Status	Public on May 02, 2025
Title	Vimentin network dysregulation mediates neurite deficits in SNCA duplication 2 Parkinson’s patient-derived midbrain neurons [RNA-seq]
Organism	Homo sapiens
Experiment type	Expression profiling by high throughput sequencing
Summary	Duplication of the SNCA gene (SNCADupl), linked to elevated levels of alpha-synuclein (aSyn), is a genetic cause of Parkinson's disease (PD). Our prior work with human-induced pluripotent stem cell (hiPSC)-derived midbrain neurons generated from PD SNCADupl patients identified neuritic deficits,

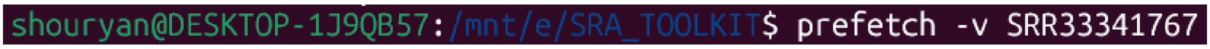
Disease Data



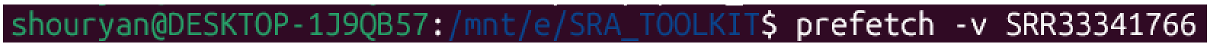
```
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ prefetch -v SRR33341769
```



```
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ prefetch -v SRR33341768
```

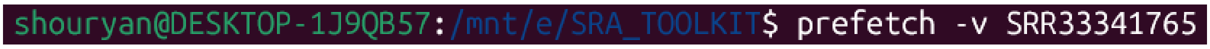


```
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ prefetch -v SRR33341767
```

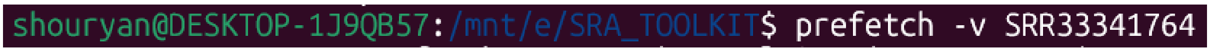


```
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ prefetch -v SRR33341766
```

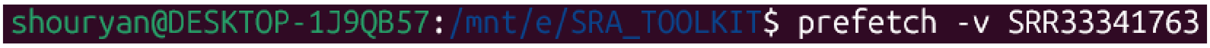
Control Data



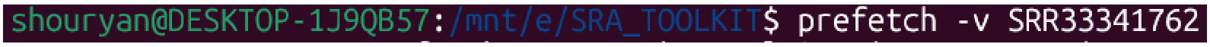
```
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ prefetch -v SRR33341765
```



```
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ prefetch -v SRR33341764
```

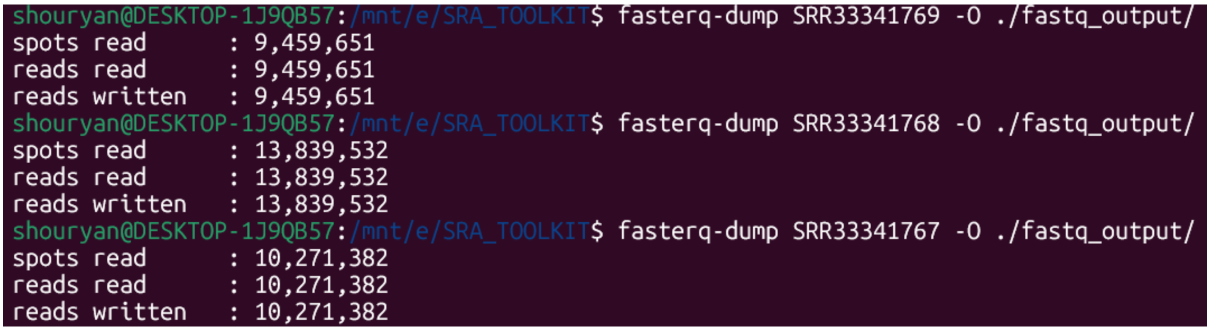


```
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ prefetch -v SRR33341763
```

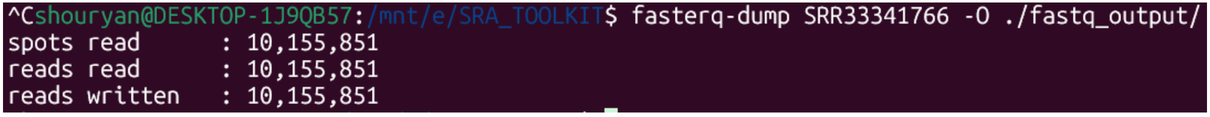


```
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ prefetch -v SRR33341762
```

2. Convert .sra to .fastq



```
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ fasterq-dump SRR33341769 -O ./fastq_output/
spots read      : 9,459,651
reads read      : 9,459,651
reads written   : 9,459,651
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ fasterq-dump SRR33341768 -O ./fastq_output/
spots read      : 13,839,532
reads read      : 13,839,532
reads written   : 13,839,532
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ fasterq-dump SRR33341767 -O ./fastq_output/
spots read      : 10,271,382
reads read      : 10,271,382
reads written   : 10,271,382
```



```
^Cshouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ fasterq-dump SRR33341766 -O ./fastq_output/
spots read      : 10,155,851
reads read      : 10,155,851
reads written   : 10,155,851
```

spots read:

Number of sequencing records (in SRA format, each "spot" is one sequencing event).

For single-end data, 1 spot = 1 read.

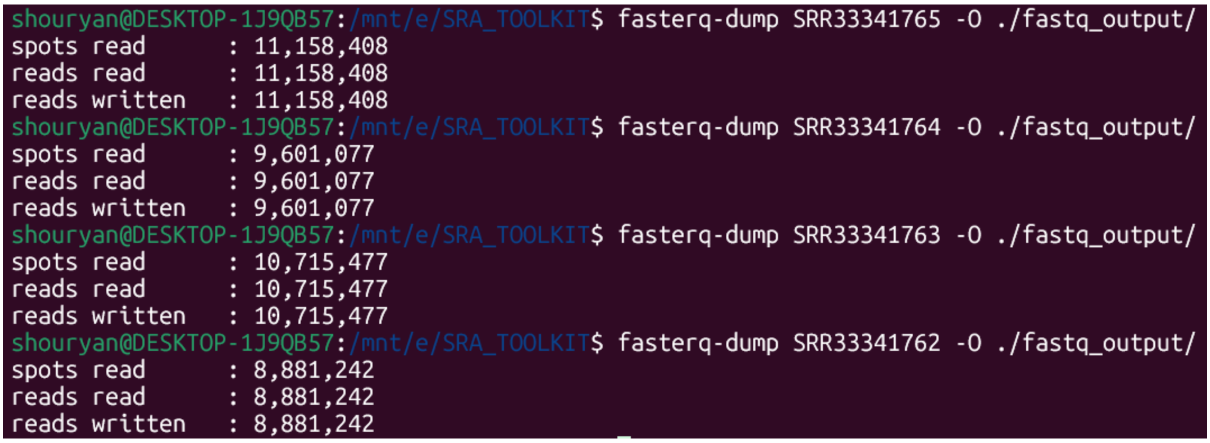
reads read:

Number of actual sequencing reads found in the .sra file.

Since it's single-end, this is equal to spots.

reads written:

Number of reads successfully written to the .fastq file.

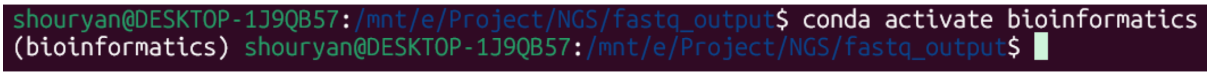


```
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ fasterq-dump SRR33341765 -O ./fastq_output/
spots read      : 11,158,408
reads read      : 11,158,408
reads written   : 11,158,408
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ fasterq-dump SRR33341764 -O ./fastq_output/
spots read      : 9,601,077
reads read      : 9,601,077
reads written   : 9,601,077
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ fasterq-dump SRR33341763 -O ./fastq_output/
spots read      : 10,715,477
reads read      : 10,715,477
reads written   : 10,715,477
shouryan@DESKTOP-1J9QB57:/mnt/e/SRA_TOOLKIT$ fasterq-dump SRR33341762 -O ./fastq_output/
spots read      : 8,881,242
reads read      : 8,881,242
reads written   : 8,881,242
```

Move fastq file to working directory

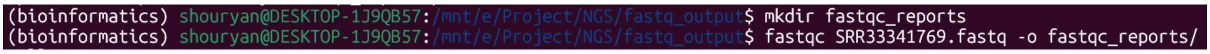
3. Quality control

Activated Conda

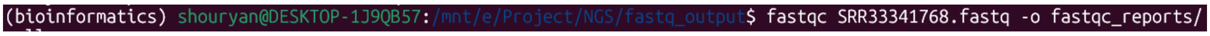


```
shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ conda activate bioinformatics
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$
```

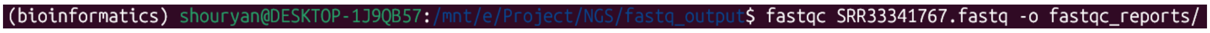
FASTQ



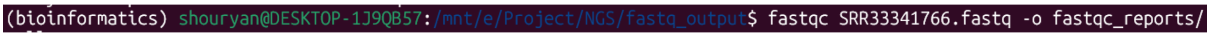
```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ mkdir fastqc_reports
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ fastqc SRR33341769.fastq -o fastqc_reports/
```



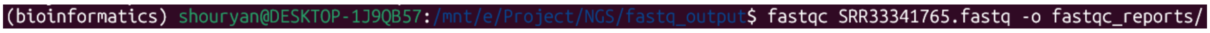
```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ fastqc SRR33341768.fastq -o fastqc_reports/
```



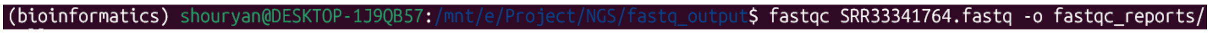
```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ fastqc SRR33341767.fastq -o fastqc_reports/
```



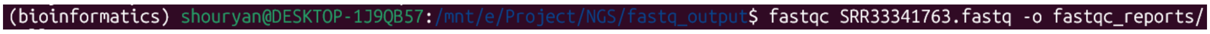
```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ fastqc SRR33341766.fastq -o fastqc_reports/
```



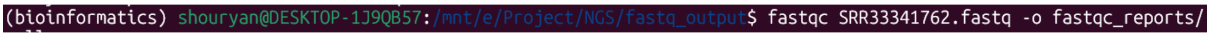
```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ fastqc SRR33341765.fastq -o fastqc_reports/
```



```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ fastqc SRR33341764.fastq -o fastqc_reports/
```



```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ fastqc SRR33341763.fastq -o fastqc_reports/
```



```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ fastqc SRR33341762.fastq -o fastqc_reports/
```

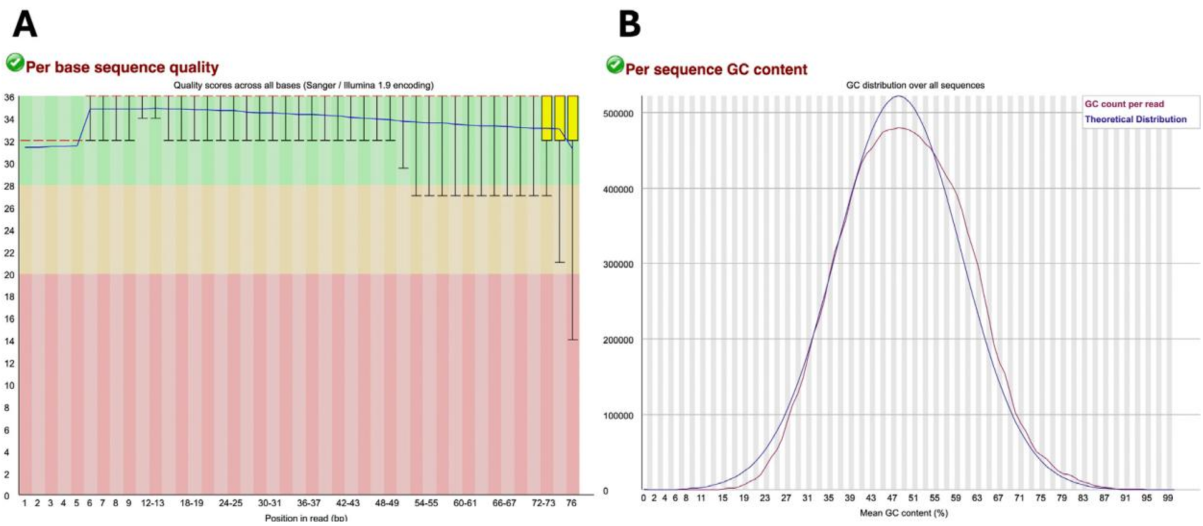


Figure 1 Explanation (Simple and Short):

A. The graph shows the quality of each base in sequencing reads.

- Green area = very good quality (Phred score > 28)
- Orange area = acceptable quality (Phred score > 20)
- Red area = poor quality (Phred score 0–20)

B. This graph shows how much GC content (Guanine + Cytosine) is in the reads.

- Blue line = expected (normal) GC content for the organism
- Red line = actual GC content in your sample

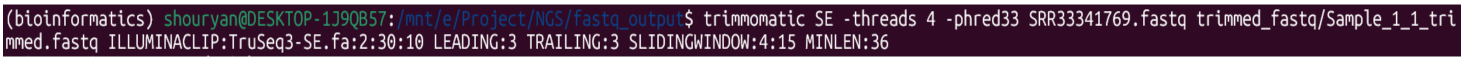
4. Trim adapters and reads of low quality using Trimmomatic

Keep the file *adapters_file.fa* in working folder

My data is single end data so I was using *TruSeq3-SE*

TruSeq3-SE file - https://github.com/usadellab/Trimmomatic/blob/main/adapters/TruSeq3-SE.fa

```
trimmomatic SE -threads 4 -phred33 \

<input_file.fastq> \

<output_file.trimmed.fastq> \

ILLUMINACLIP:<adapters_file.fa>:2:30:10 \

LEADING:3 \

TRAILING:3 \

SLIDINGWINDOW:4:15 \
MINLEN:36
```

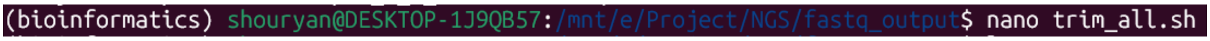


```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ trimmomatic SE -threads 4 -phred33 SRR33341769.fastq trimmed_fastq/Sample_1_1_tri
mmed.fastq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36
```

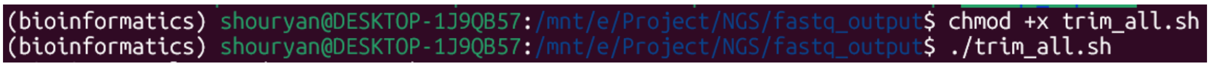
And other 7 manually code

OR

Run script



```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ nano trim_all.sh
```

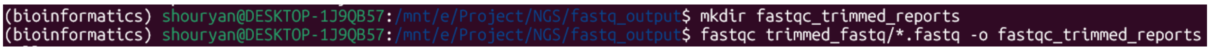


```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ chmod +x trim_all.sh
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ ./trim_all.sh
```

trim_all.sh contain

```
#!/bin/bash


# Make sure output directory exists

mkdir -p trimmed_fastq


# Array of input files and corresponding sample names

declare -a samples=(

  "SRR33341769 Sample_1_1"

  "SRR33341768 Sample_1_2"

  "SRR33341767 Sample_1_3"

  "SRR33341766 Sample_1_4"

  "SRR33341765 Sample_2_1"

  "SRR33341764 Sample_2_2"

  "SRR33341763 Sample_2_3"

  "SRR33341762 Sample_2_4"

)


# Path to adapter file (adjust if needed)

ADAPTER="TruSeq3-SE.fa"


# Loop through each sample and run Trimmomatic

for entry in "${samples[@]}"; do

  read SRR_ID SAMPLE_NAME <<< "$entry"


  echo "Trimming $SAMPLE_NAME ($SRR_ID)..."

  trimmomatic SE -threads 4 -phred33 \
    "${SRR_ID}.fastq" \
    "trimmed_fastq/${SAMPLE_NAME}_trimmed.fastq" \
    ILLUMINACLIP:$ADAPTER:2:30:10 \
    LEADING:3 \
    TRAILING:3 \
    SLIDINGWINDOW:4:15 \
    MINLEN:36

  echo "Finished $SAMPLE_NAME"
done

echo "All samples trimmed."
```

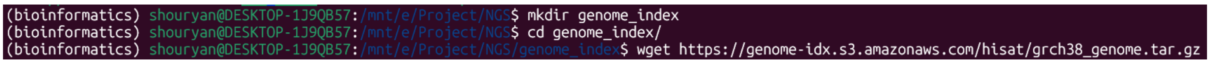
5. Run quality control again on trimmed files



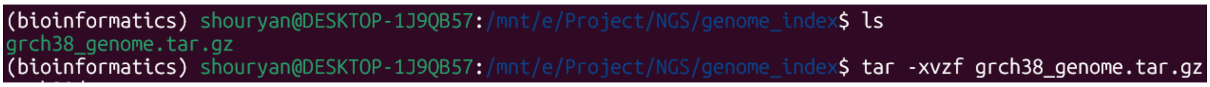
```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ mkdir fastqc_trimmed_reports
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/fastq_output$ fastqc trimmed_fastq/*.fastq -o fastqc_trimmed_reports
```

6. Align to genome using HISAT2

Create directory and download genome data of human

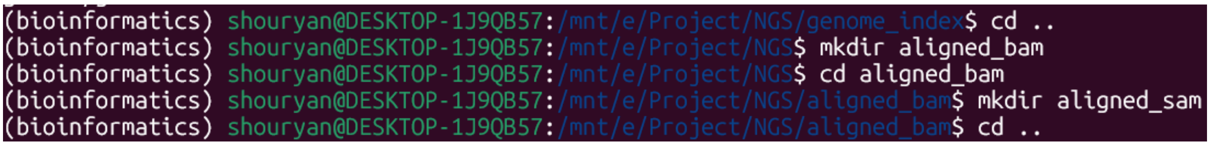


```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ mkdir genome_index
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ cd genome_index/
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/genome_index$ wget https://genome-idx.s3.amazonaws.com/hisat/grch38_genome.tar.gz
```



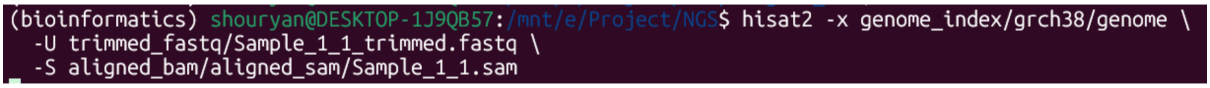
```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/genome_index$ ls
grch38_genome.tar.gz
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/genome_index$ tar -xvzf grch38_genome.tar.gz
```

Create directory to place data at desired location



```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/genome_index$ cd ..
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ mkdir aligned_bam
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ cd aligned_bam
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/aligned_bam$ mkdir aligned_sam
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/aligned_bam$ cd ..
```

Know align to genome using HISAT2

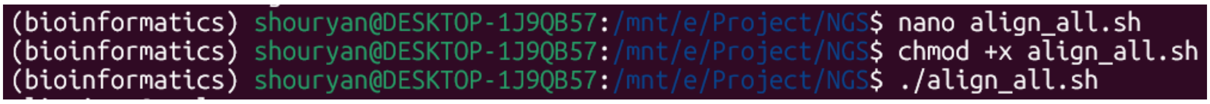


```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ hisat2 -x genome_index/grch38/genome \
  -U trimmed_fastq/Sample_1_1_trimmed.fastq \
  -S aligned_bam/aligned_sam/Sample_1_1.sam
```

do for rest of sample other 7 manually

OR

Run script

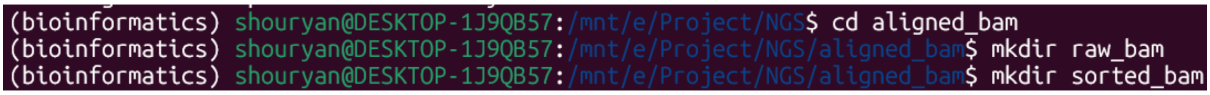


```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ nano align_all.sh
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ chmod +x align_all.sh
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ ./align_all.sh
```

7. File conversion

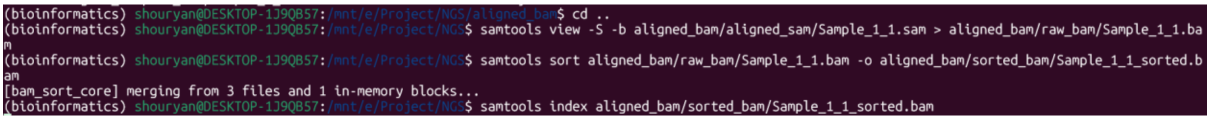
Move to folder there sam file

Create other directory to keep the data in particular folder



```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ cd aligned_bam
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/aligned_bam$ mkdir raw_bam
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/aligned_bam$ mkdir sorted_bam
```

Then convert .sam file to .bam (manually)

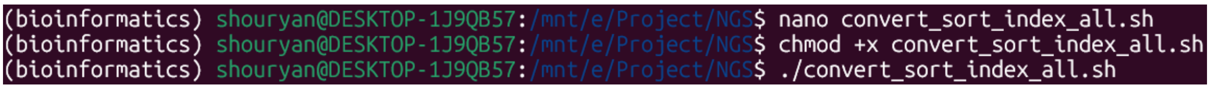


```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS/aligned_bam$ cd ..
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ samtools view -S -b aligned_bam/aligned_sam/Sample_1_1.sam > aligned_bam/raw_bam/Sample_1_1.ba
m
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ samtools sort aligned_bam/raw_bam/Sample_1_1.bam -o aligned_bam/sorted_bam/Sample_1_1_sorted.b
am
[bam_sort_core] merging from 3 files and 1 in-memory blocks...
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ samtools index aligned_bam/sorted_bam/Sample_1_1_sorted.bam
```

Step	Command	Why It's Needed
Convert	samtools view	Convert .sam to .bam (compressed)
Sort	samtools sort	Required for indexing and downstream tools
Index	samtools index	Enables fast querying & visualization

OR

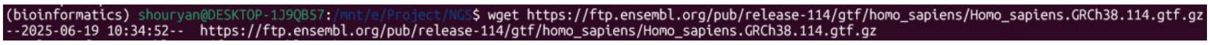
Create Script convert_sort_index_all.sh



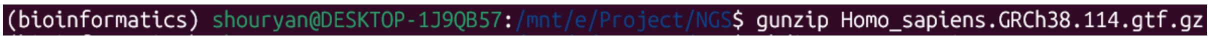
```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ nano convert_sort_index_all.sh
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ chmod +x convert_sort_index_all.sh
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ ./convert_sort_index_all.sh
```

8. Count reads per gene using featureCounts

Download file and unzip it

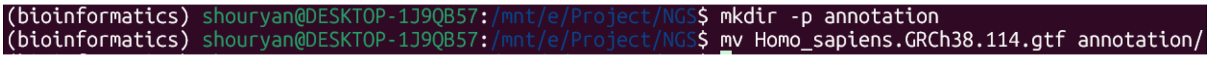


```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ wget https://ftp.ensembl.org/pub/release-114/gtf/homo_sapiens/Homo_sapiens.GRCh38.114.gtf.gz
--2025-06-19 10:34:52--  https://ftp.ensembl.org/pub/release-114/gtf/homo_sapiens/Homo_sapiens.GRCh38.114.gtf.gz
```



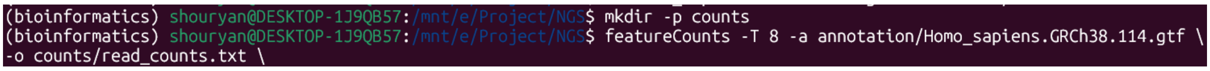
```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ gunzip Homo_sapiens.GRCh38.114.gtf.gz
```

Make folder and Move file to that folder

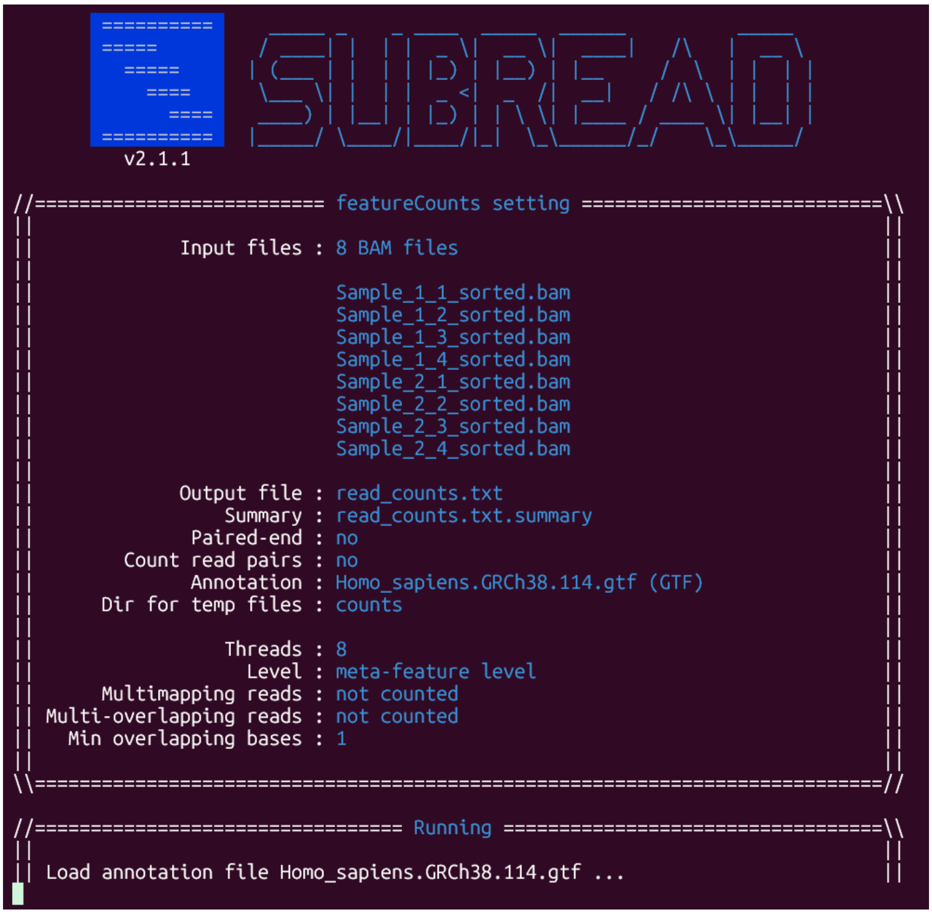


```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ mkdir -p annotation
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ mv Homo_sapiens.GRCh38.114.gtf annotation/
```

Create new folder and run featureCounts command

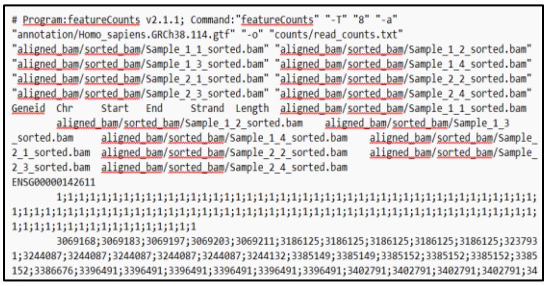


```
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ mkdir -p counts
(bioinformatics) shouryan@DESKTOP-1J9QB57:/mnt/e/Project/NGS$ featureCounts -T 8 -a annotation/Homo_sapiens.GRCh38.114.gtf \
-o counts/read_counts.txt \
```

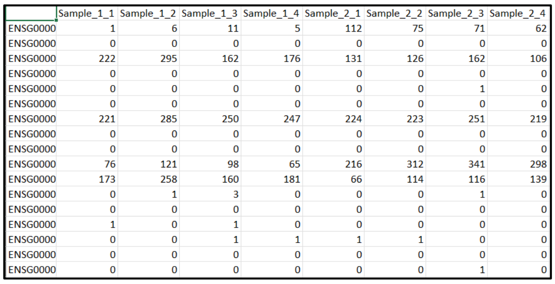


```
       v2.1.1

//========================== featureCounts setting ===========================\\
||                                                                            ||
||             Input files : 8 BAM files                                      ||
||                                                                            ||
||                           Sample_1_1_sorted.bam                            ||
||                           Sample_1_2_sorted.bam                            ||
||                           Sample_1_3_sorted.bam                            ||
||                           Sample_1_4_sorted.bam                            ||
||                           Sample_2_1_sorted.bam                            ||
||                           Sample_2_2_sorted.bam                            ||
||                           Sample_2_3_sorted.bam                            ||
||                           Sample_2_4_sorted.bam                            ||
||                                                                            ||
||             Output file : read_counts.txt                                  ||
||                 Summary : read_counts.txt.summary                          ||
||              Paired-end : no                                               ||
||        Count read pairs : no                                               ||
||              Annotation : Homo_sapiens.GRCh38.114.gtf (GTF)                ||
||      Dir for temp files : counts                                           ||
||                                                                            ||
||                 Threads : 8                                                ||
||                   Level : meta-feature level                               ||
||      Multimapping reads : not counted                                      ||
|| Multi-overlapping reads : not counted                                      ||
||   Min overlapping bases : 1                                                ||
||                                                                            ||
\\============================================================================//

//================================= Running ==================================\\
||                                                                            ||
|| Load annotation file Homo_sapiens.GRCh38.114.gtf ...                       ||
```

Next in we convert read_count.txt to proper .csv file



```
# Program:featureCounts v2.1.1; Command:"featureCounts" "-T" "8" "-a"
"annotation/Homo_sapiens.GRCh38.114.gtf" "-o" "counts/read_counts.txt"
"aligned_bam/sorted_bam/Sample_1_1_sorted.bam" "aligned_bam/sorted_bam/Sample_1_2_sorted.bam"
"aligned_bam/sorted_bam/Sample_1_3_sorted.bam" "aligned_bam/sorted_bam/Sample_1_4_sorted.bam"
"aligned_bam/sorted_bam/Sample_2_1_sorted.bam" "aligned_bam/sorted_bam/Sample_2_2_sorted.bam"
"aligned_bam/sorted_bam/Sample_2_3_sorted.bam" "aligned_bam/sorted_bam/Sample_2_4_sorted.bam"
Geneid  Chr     Start   End     Strand  Length  aligned_bam/sorted_bam/Sample_1_1_sorted.bam
        aligned_bam/sorted_bam/Sample_1_2_sorted.bam    aligned_bam/sorted_bam/Sample_1_3
_sorted.bam     aligned_bam/sorted_bam/Sample_1_4_sorted.bam    aligned_bam/sorted_bam/Sample_
2_1_sorted.bam  aligned_bam/sorted_bam/Sample_2_2_sorted.bam    aligned_bam/sorted_bam/Sample_
2_3_sorted.bam  aligned_bam/sorted_bam/Sample_2_4_sorted.bam
ENSG00000142611
        1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;
1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;
1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1
        3069168;3069183;3069197;3069203;3069211;3186125;3186125;3186125;3186125;3186125;323793
1;3244087;3244087;3244087;3244087;3244087;3244132;3385149;3385149;3385152;3385152;3385152;3385
152;3386676;3396491;3396491;3396491;3396491;3396491;3396491;3402791;3402791;3402791;3402791;34
```



	Sample_1_1	Sample_1_2	Sample_1_3	Sample_1_4	Sample_2_1	Sample_2_2	Sample_2_3	Sample_2_4
ENSG0000	1	6	11	5	112	75	71	62
ENSG0000	0	0	0	0	0	0	0	0
ENSG0000	222	295	162	176	131	126	162	106
ENSG0000	0	0	0	0	0	0	0	0
ENSG0000	0	0	0	0	0	0	1	0
ENSG0000	0	0	0	0	0	0	0	0
ENSG0000	221	285	250	247	224	223	251	219
ENSG0000	0	0	0	0	0	0	0	0
ENSG0000	0	0	0	0	0	0	0	0
ENSG0000	76	121	98	65	216	312	341	298
ENSG0000	173	258	160	181	66	114	116	139
ENSG0000	0	1	3	0	0	0	1	0
ENSG0000	0	0	0	0	0	0	0	0
ENSG0000	1	0	1	0	0	0	0	0
ENSG0000	0	0	1	1	1	1	0	0
ENSG0000	0	0	0	0	0	0	0	0
ENSG0000	0	0	0	0	0	0	1	0

RStudio

Set up

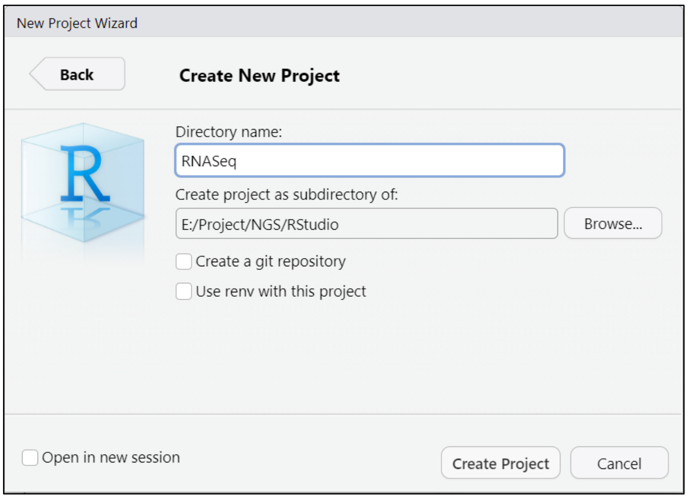
Open RStudio

- On left side *File > New project > New Directory > New Project >

(Write Directory name *RNASeq)

(Create project as subdirectory of:

In this select your working directory) > Create Project



Directory name: RNASeq

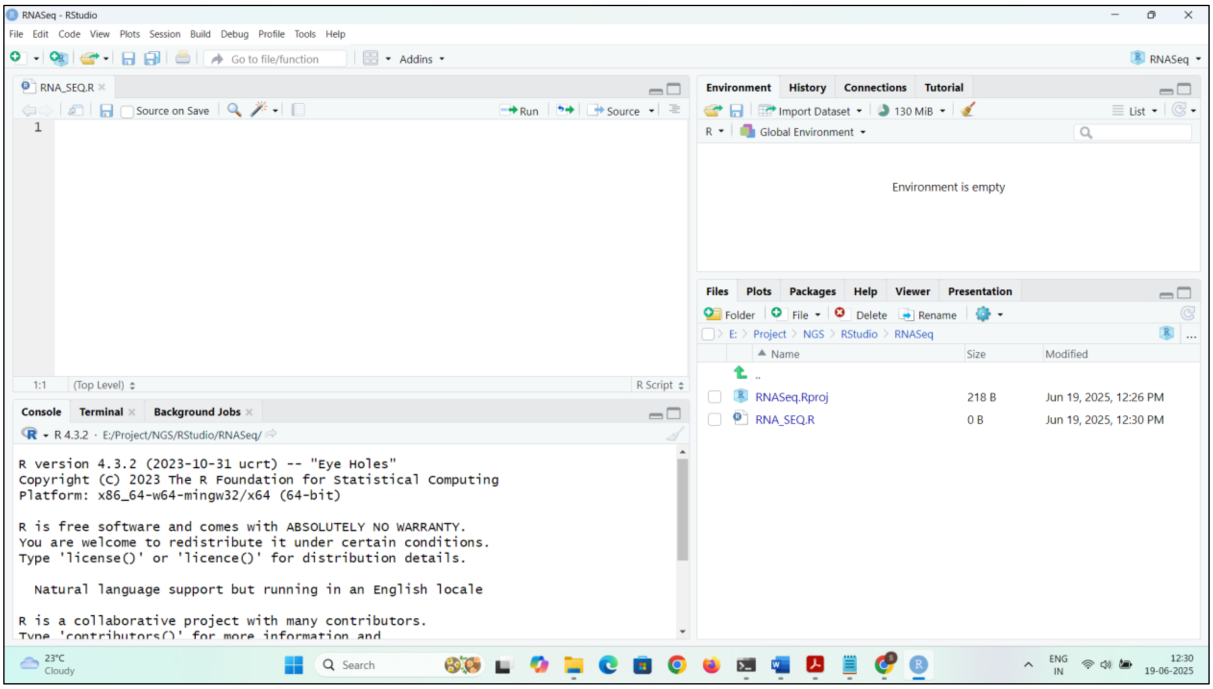
Create project as subdirectory of: E:/Project/NGS/RStudio

□ Create a git repository

□ Use renv with this project

□ Open in new session

- File > New File > R Script
- File > Save AS > (write a name of your file) > Save



R 4.3.2 · E:/Project/NGS/RStudio/RNASeq/

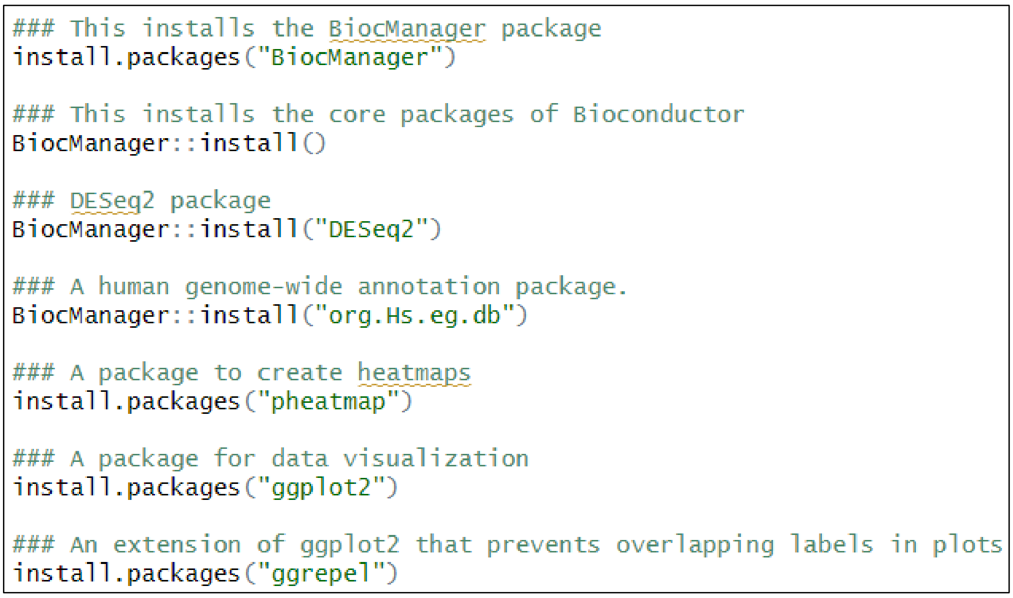
```
R version 4.3.2 (2023-10-31 ucrt) -- "Eye Holes"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
```

Environment is empty

E: > Project > NGS > RStudio > RNASeq

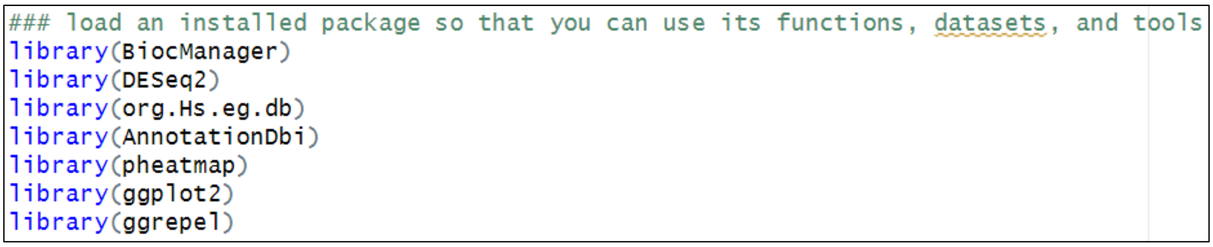
Name	Size	Modified
..		
RNASeq.Rproj	218 B	Jun 19, 2025, 12:26 PM
RNA_SEQ.R	0 B	Jun 19, 2025, 12:30 PM

1. Install Required packages



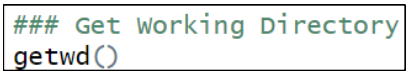
```
### This installs the BiocManager package
install.packages("BiocManager")

### This installs the core packages of Bioconductor
BiocManager::install()

### DESeq2 package
BiocManager::install("DESeq2")

### A human genome-wide annotation package.
BiocManager::install("org.Hs.eg.db")

### A package to create heatmaps
install.packages("pheatmap")

### A package for data visualization
install.packages("ggplot2")

### An extension of ggplot2 that prevents overlapping labels in plots
install.packages("ggrepel")
```

Load package



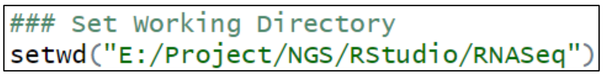
```
### load an installed package so that you can use its functions, datasets, and tools
library(BiocManager)
library(DESeq2)
library(org.Hs.eg.db)
library(AnnotationDbi)
library(pheatmap)
library(ggplot2)
library(ggrepel)
```

Get working directory



```
### Get Working Directory
getwd()
```

Set Working directory



```
### Set Working Directory
setwd("E:/Project/NGS/RStudio/RNASeq")
```

Command	**Meaning**	**Use Case**
getwd()	Get current folder path	Check where R is operating
setwd()	Change current folder	Set where to read/write files easily

2. Clean and Load data

Before loading data keep **read_counts.txt** file in working folder where RNA_SEQ.R file is

```
### Read all lines
lines <- readLines("read_counts.txt")
```

```
### Check number of columns in each line
line_lengths <- sapply(strsplit(lines, "\t"), length)
```

```
### Keep only lines with 14 columns (i.e., correct ones)
clean_lines <- lines[line_lengths == 14]
```

```
### Write cleaned data to new file
writeLines(clean_lines, "clean_read_counts.txt")
```

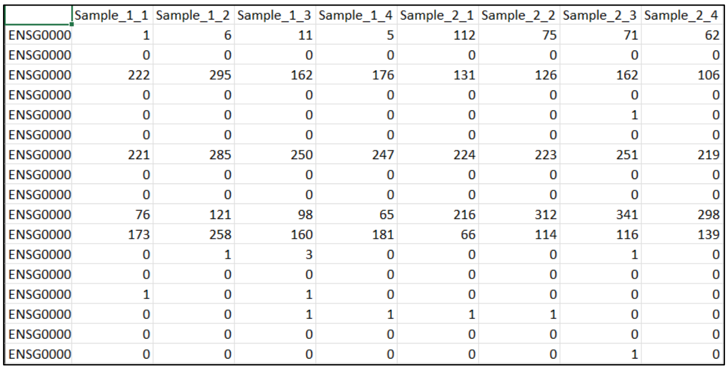
```
# Read the cleaned file
counts_txt <- read.delim("clean_read_counts.txt", header = TRUE, row.names = 1, sep = "\t")
```

```
### Remove unwanted columns by name (if they exist)
unwanted_cols <- c("Chr", "Start", "End", "Strand", "Length")
counts_txt <- counts_txt[ , !(colnames(counts_txt) %in% unwanted_cols) ]
```

```
### Clean column names
colnames(counts_txt) <- gsub(".*(Sample_\\d+_\\d+).*", "\\1", colnames(counts_txt))
```

```
### Save as CSV
write.csv(counts_txt, "read_counts.csv")
```

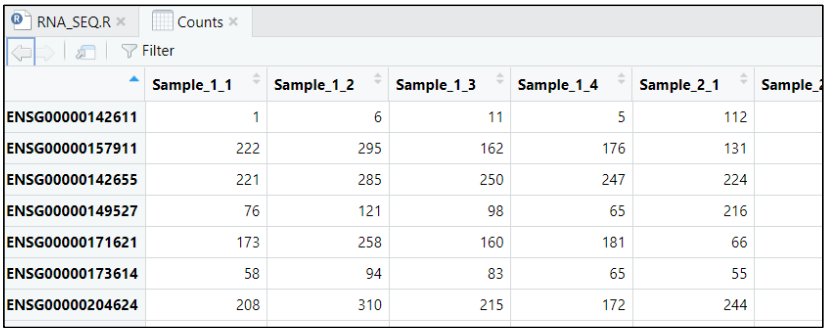
Output of this code is



	Sample_1_1	Sample_1_2	Sample_1_3	Sample_1_4	Sample_2_1	Sample_2_2	Sample_2_3	Sample_2_4
ENSG0000	1	6	11	5	112	75	71	62
ENSG0000	0	0	0	0	0	0	0	0
ENSG0000	222	295	162	176	131	126	162	106
ENSG0000	0	0	0	0	0	0	0	0
ENSG0000	0	0	0	0	0	0	1	0
ENSG0000	0	0	0	0	0	0	0	0
ENSG0000	221	285	250	247	224	223	251	219
ENSG0000	0	0	0	0	0	0	0	0
ENSG0000	0	0	0	0	0	0	0	0
ENSG0000	76	121	98	65	216	312	341	298
ENSG0000	173	258	160	181	66	114	116	139
ENSG0000	0	1	3	0	0	0	1	0
ENSG0000	0	0	0	0	0	0	0	0
ENSG0000	1	0	1	0	0	0	0	0
ENSG0000	0	0	1	1	1	1	0	0
ENSG0000	0	0	0	0	0	0	0	0
ENSG0000	0	0	0	0	0	0	1	0

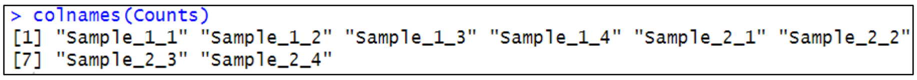
```
### Load from CSV
Counts <- read.delim("read_counts.csv", header = TRUE, row.names = 1, sep = ",")
```

```
### Keeping only genes with row sums greater than 50
Counts <- Counts[which(rowSums(Counts) > 50), ]
```

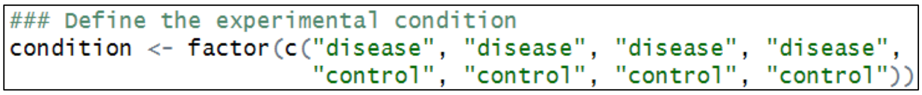


	Sample_1_1	Sample_1_2	Sample_1_3	Sample_1_4	Sample_2_1	Sample_2
ENSG00000142611	1	6	11	5	112	
ENSG00000157911	222	295	162	176	131	
ENSG00000142655	221	285	250	247	224	
ENSG00000149527	76	121	98	65	216	
ENSG00000171621	173	258	160	181	66	
ENSG00000173614	58	94	83	65	55	
ENSG00000204624	208	310	215	172	244	

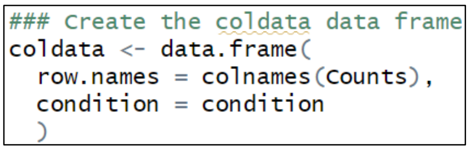
3. Run DESeq2



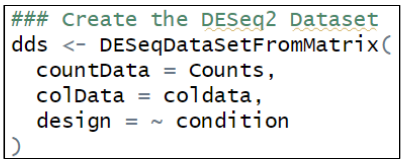
```
> colnames(Counts)
[1] "Sample_1_1" "Sample_1_2" "Sample_1_3" "Sample_1_4" "Sample_2_1" "Sample_2_2"
[7] "Sample_2_3" "Sample_2_4"
```



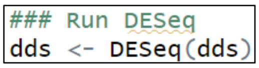
```
### Define the experimental condition
condition <- factor(c("disease", "disease", "disease", "disease",
                      "control", "control", "control", "control"))
```



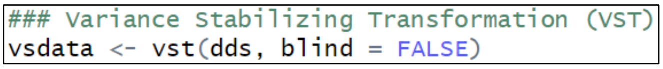
```
### Create the coldata data frame
coldata <- data.frame(
  row.names = colnames(Counts),
  condition = condition
  )
```



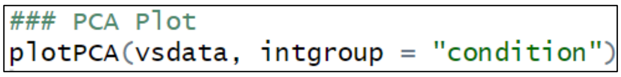
```
### Create the DESeq2 Dataset
dds <- DESeqDataSetFromMatrix(
  countData = Counts,
  colData = coldata,
  design = ~ condition
)
```



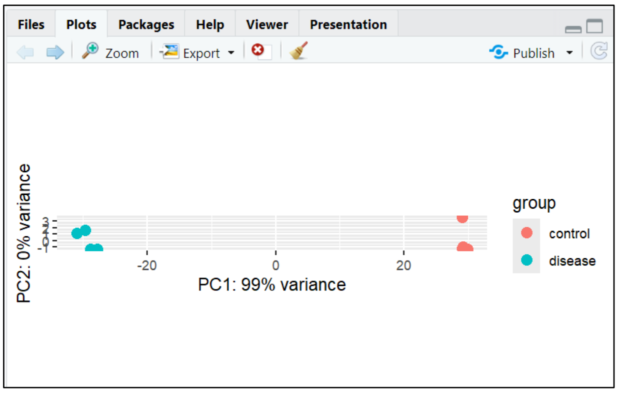
```
### Run DESeq
dds <- DESeq(dds)
```

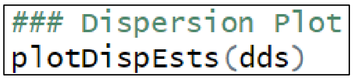


```
### Variance Stabilizing Transformation (VST)
vsdata <- vst(dds, blind = FALSE)
```

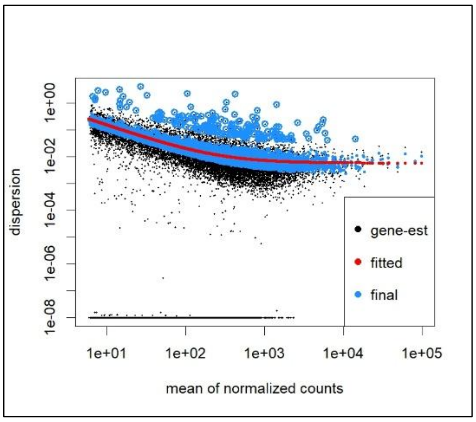


```
### PCA Plot
plotPCA(vsdata, intgroup = "condition")
```

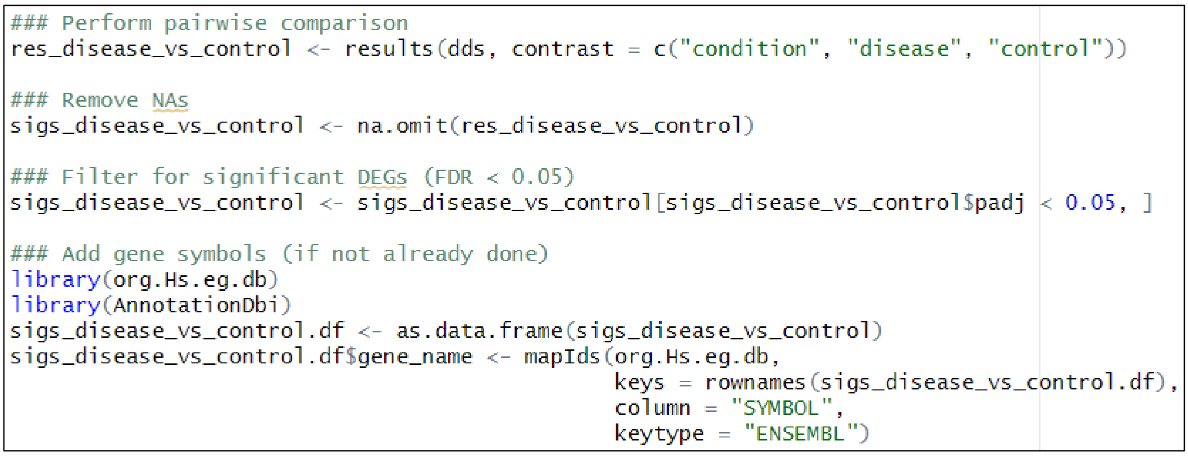




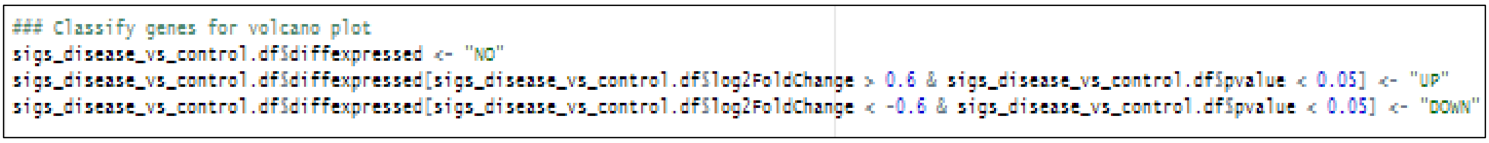
```
### Dispersion Plot
plotDispEsts(dds)
```



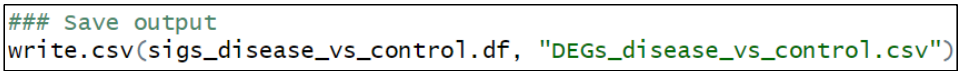
4. Pairwise comparisons between samples



```r
### Perform pairwise comparison
res_disease_vs_control <- results(dds, contrast = c("condition", "disease", "control"))

### Remove NAs
sigs_disease_vs_control <- na.omit(res_disease_vs_control)

### Filter for significant DEGs (FDR < 0.05)
sigs_disease_vs_control <- sigs_disease_vs_control[sigs_disease_vs_control$padj < 0.05, ]

### Add gene symbols (if not already done)
library(org.Hs.eg.db)
library(AnnotationDbi)
sigs_disease_vs_control.df <- as.data.frame(sigs_disease_vs_control)
sigs_disease_vs_control.df$gene_name <- mapIds(org.Hs.eg.db,
                                               keys = rownames(sigs_disease_vs_control.df),
                                               column = "SYMBOL",
                                               keytype = "ENSEMBL")
```

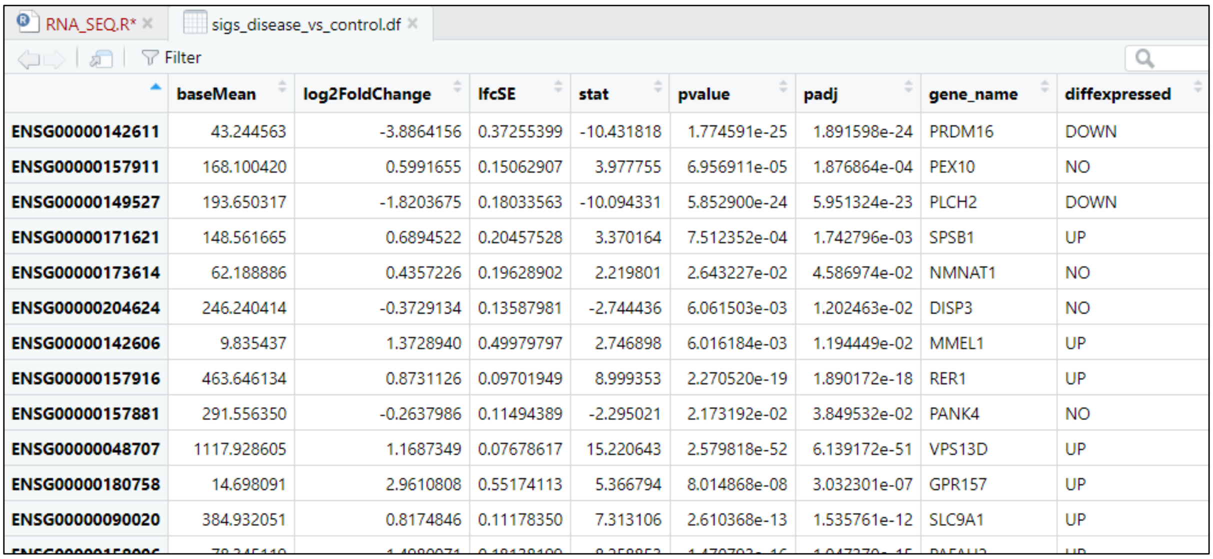


```r
### Classify genes for volcano plot
sigs_disease_vs_control.df$diffexpressed <- "NO"
sigs_disease_vs_control.df$diffexpressed[sigs_disease_vs_control.df$log2FoldChange > 0.6 & sigs_disease_vs_control.df$pvalue < 0.05] <- "UP"
sigs_disease_vs_control.df$diffexpressed[sigs_disease_vs_control.df$log2FoldChange < -0.6 & sigs_disease_vs_control.df$pvalue < 0.05] <- "DOWN"
```



```r
### Save output
write.csv(sigs_disease_vs_control.df, "DEGs_disease_vs_control.csv")
```

Out file of sigs_disease_vs_control.df

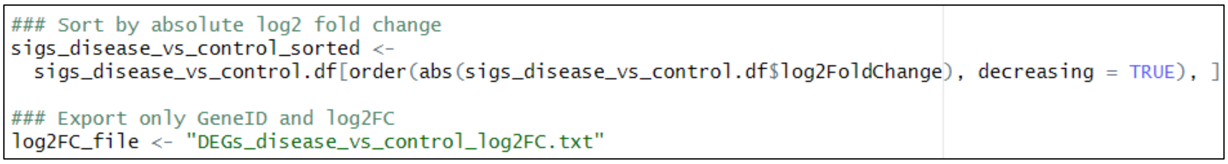


	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	gene_name	diffexpressed
ENSG00000142611	43.244563	-3.8864156	0.37255399	-10.431818	1.774591e-25	1.891598e-24	PRDM16	DOWN
ENSG00000157911	168.100420	0.5991655	0.15062907	3.977755	6.956911e-05	1.876864e-04	PEX10	NO
ENSG00000149527	193.650317	-1.8203675	0.18033563	-10.094331	5.852900e-24	5.951324e-23	PLCH2	DOWN
ENSG00000171621	148.561665	0.6894522	0.20457528	3.370164	7.512352e-04	1.742796e-03	SPSB1	UP
ENSG00000173614	62.188886	0.4357226	0.19628902	2.219801	2.643227e-02	4.586974e-02	NMNAT1	NO
ENSG00000204624	246.240414	-0.3729134	0.13587981	-2.744436	6.061503e-03	1.202463e-02	DISP3	NO
ENSG00000142606	9.835437	1.3728940	0.49979797	2.746898	6.016184e-03	1.194449e-02	MMEL1	UP
ENSG00000157916	463.646134	0.8731126	0.09701949	8.999353	2.270520e-19	1.890172e-18	RER1	UP
ENSG00000157881	291.556350	-0.2637986	0.11494389	-2.295021	2.173192e-02	3.849532e-02	PANK4	NO
ENSG00000048707	1117.928605	1.1687349	0.07678617	15.220643	2.579818e-52	6.139172e-51	VPS13D	UP
ENSG00000180758	14.698091	2.9610808	0.55174113	5.366794	8.014868e-08	3.032301e-07	GPR157	UP
ENSG00000090020	384.932051	0.8174846	0.11178350	7.313106	2.610368e-13	1.535761e-12	SLC9A1	UP

5. Generating an ordered list of DEGs

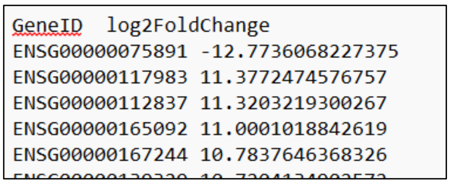
It helps you:

- See which genes are most strongly up- or down-regulated
- Prioritize biologically meaningful DEGs

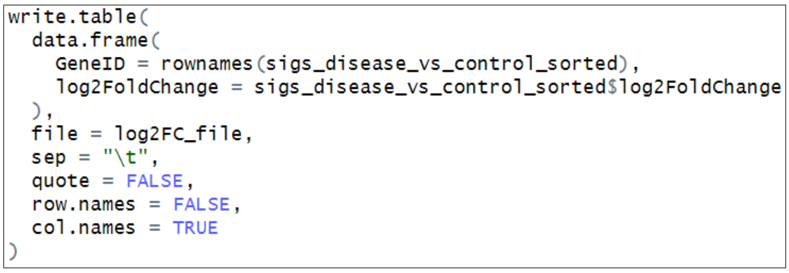


```
### Sort by absolute log2 fold change
sigs_disease_vs_control_sorted <-
  sigs_disease_vs_control.df[order(abs(sigs_disease_vs_control.df$log2FoldChange), decreasing = TRUE), ]

### Export only GeneID and log2FC
log2FC_file <- "DEGs_disease_vs_control_log2FC.txt"
```

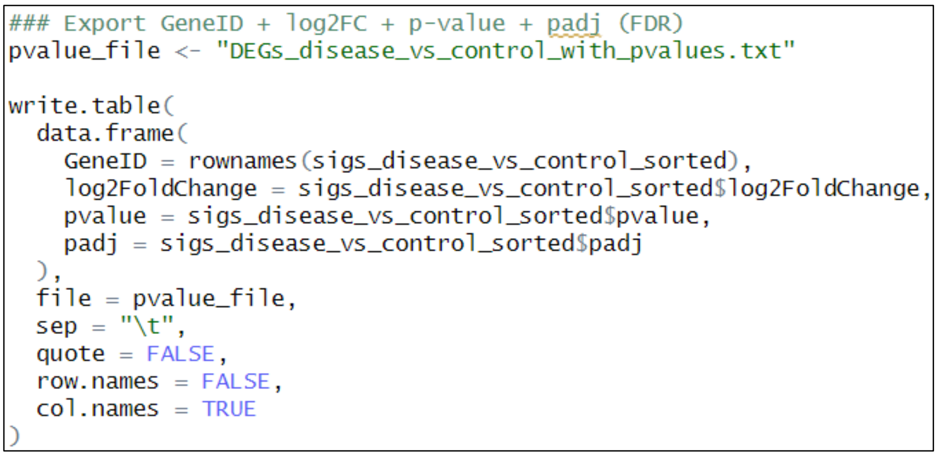
Output of DEGs_disease_vs_control_log2FC.txt



```
GeneID  log2FoldChange
ENSG00000075891 -12.7736068227375
ENSG00000117983 11.3772474576757
ENSG00000112837 11.3203219300267
ENSG00000165092 11.0001018842619
ENSG00000167244 10.7837646368326
```



```
write.table(
  data.frame(
    GeneID = rownames(sigs_disease_vs_control_sorted),
    log2FoldChange = sigs_disease_vs_control_sorted$log2FoldChange
  ),
  file = log2FC_file,
  sep = "\t",
  quote = FALSE,
  row.names = FALSE,
  col.names = TRUE
)
```

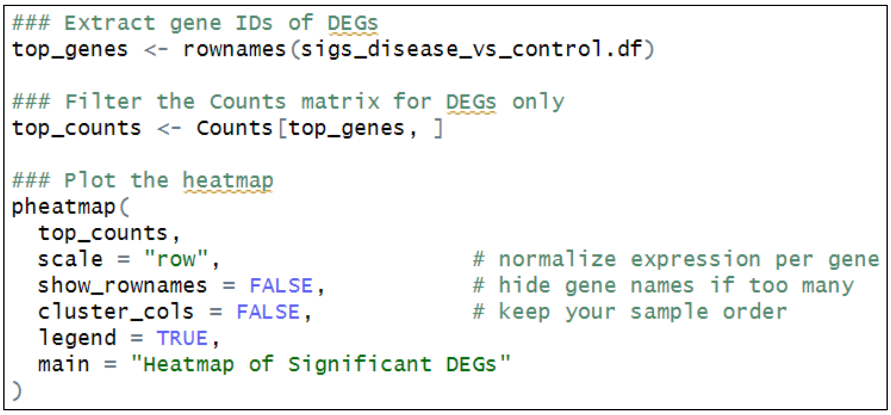


```
### Export GeneID + log2FC + p-value + padj (FDR)
pvalue_file <- "DEGs_disease_vs_control_with_pvalues.txt"

write.table(
  data.frame(
    GeneID = rownames(sigs_disease_vs_control_sorted),
    log2FoldChange = sigs_disease_vs_control_sorted$log2FoldChange,
    pvalue = sigs_disease_vs_control_sorted$pvalue,
    padj = sigs_disease_vs_control_sorted$padj
  ),
  file = pvalue_file,
  sep = "\t",
  quote = FALSE,
  row.names = FALSE,
  col.names = TRUE
)
```

Summary

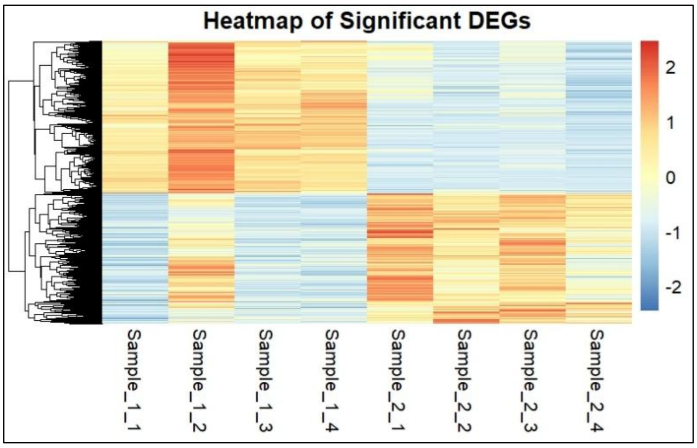
Step	Purpose
Sort by	log2FC
Export TXT table	Input to downstream tools like GOrilla, DAVID, Enrichr
Add p-value & FDR	Let reviewers or readers assess statistical strength
Use in volcano/heatmap	Easy to visualize and annotate top genes
Reproducible and shareable	Keeps your pipeline clean and documented

6. Generate heatmaps

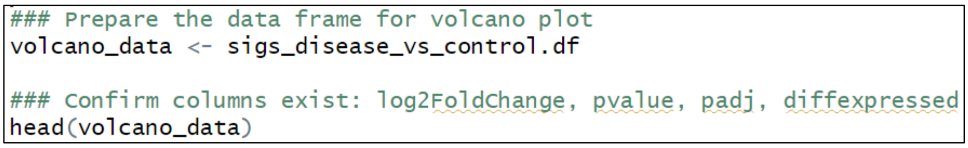


```
### Extract gene IDs of DEGs
top_genes <- rownames(sigs_disease_vs_control.df)

### Filter the Counts matrix for DEGs only
top_counts <- Counts[top_genes, ]

### Plot the heatmap
pheatmap(
  top_counts,
  scale = "row",                 # normalize expression per gene
  show_rownames = FALSE,         # hide gene names if too many
  cluster_cols = FALSE,          # keep your sample order
  legend = TRUE,
  main = "Heatmap of Significant DEGs"
)
```

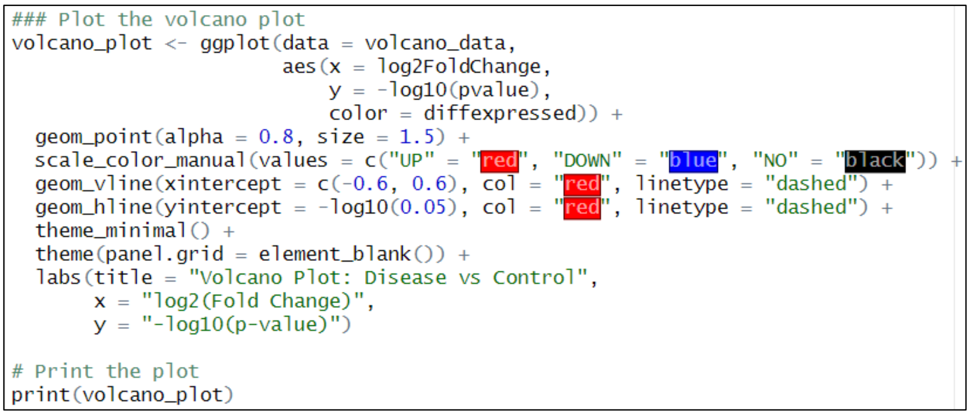
Output Heatmap



7. A) Generating volcano plots

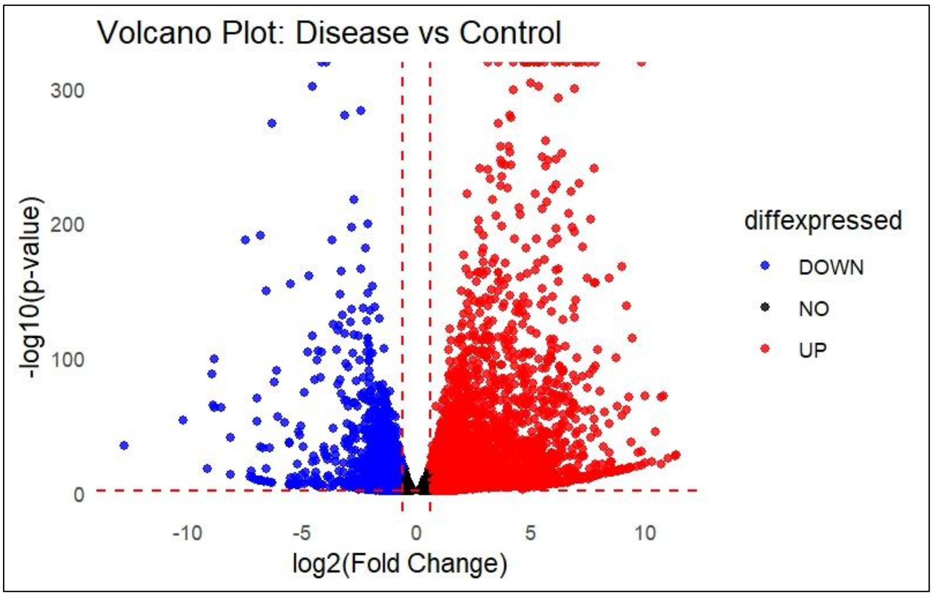


```
### Prepare the data frame for volcano plot
volcano_data <- sigs_disease_vs_control.df

### Confirm columns exist: log2FoldChange, pvalue, padj, diffexpressed
head(volcano_data)
```

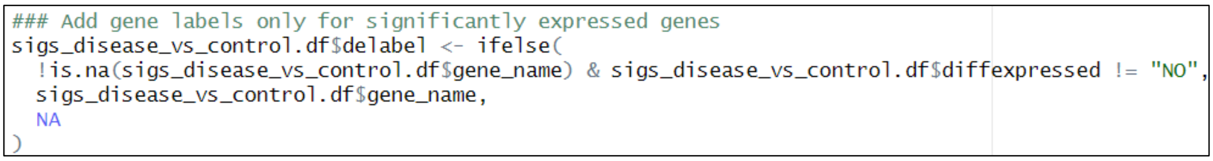


```
### Plot the volcano plot
volcano_plot <- ggplot(data = volcano_data,
                       aes(x = log2FoldChange,
                           y = -log10(pvalue),
                           color = diffexpressed)) +
  geom_point(alpha = 0.8, size = 1.5) +
  scale_color_manual(values = c("UP" = "red", "DOWN" = "blue", "NO" = "black")) +
  geom_vline(xintercept = c(-0.6, 0.6), col = "red", linetype = "dashed") +
  geom_hline(yintercept = -log10(0.05), col = "red", linetype = "dashed") +
  theme_minimal() +
  theme(panel.grid = element_blank()) +
  labs(title = "Volcano Plot: Disease vs Control",
       x = "log2(Fold Change)",
       y = "-log10(p-value)")

# Print the plot
print(volcano_plot)
```

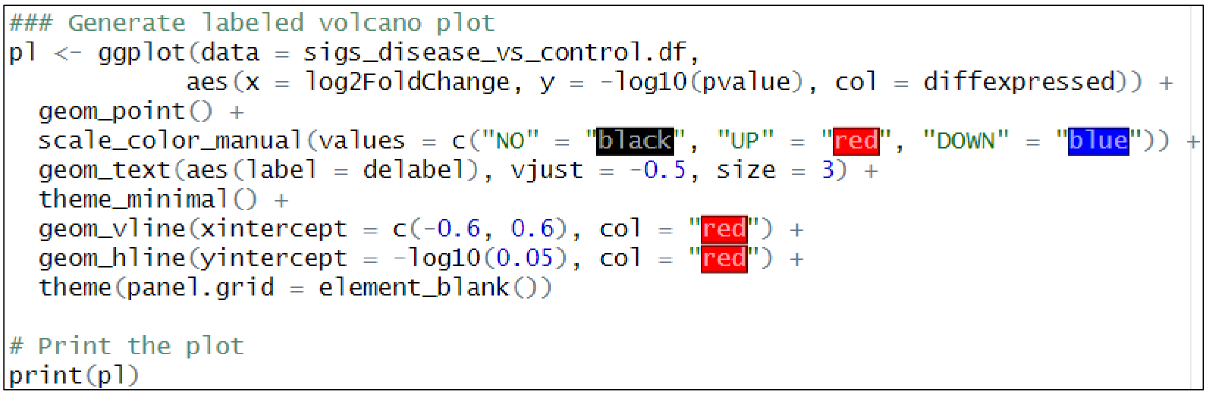
Volcano plot



7. B) Generating volcano plots with labelled DEGs

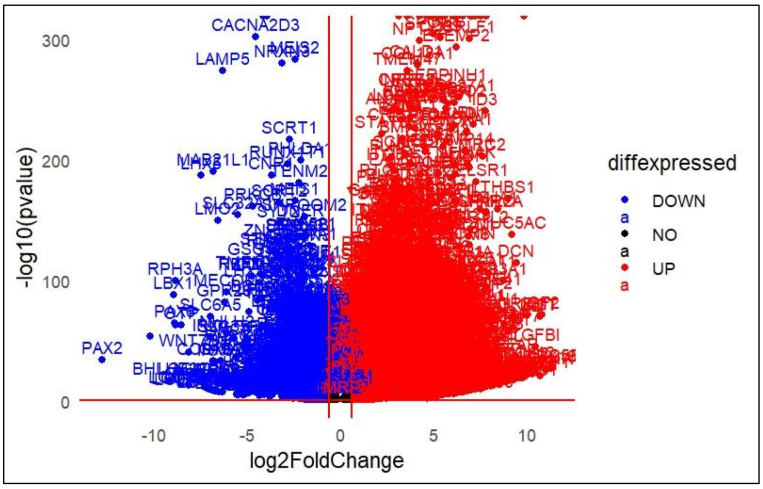


```
### Add gene labels only for significantly expressed genes
sigs_disease_vs_control.df$delabel <- ifelse(
  !is.na(sigs_disease_vs_control.df$gene_name) & sigs_disease_vs_control.df$diffexpressed != "NO",
  sigs_disease_vs_control.df$gene_name,
  NA
)
```



```
### Generate labeled volcano plot
pl <- ggplot(data = sigs_disease_vs_control.df,
             aes(x = log2FoldChange, y = -log10(pvalue), col = diffexpressed)) +
  geom_point() +
  scale_color_manual(values = c("NO" = "black", "UP" = "red", "DOWN" = "blue")) +
  geom_text(aes(label = delabel), vjust = -0.5, size = 3) +
  theme_minimal() +
  geom_vline(xintercept = c(-0.6, 0.6), col = "red") +
  geom_hline(yintercept = -log10(0.05), col = "red") +
  theme(panel.grid = element_blank())

# Print the plot
print(pl)
```

Volcano plot with Labelled DEGs


CACNA2D3
LAMP5
MEIS2
SCRT1
RUNX1
ENM2
LMO2
RPH3A
LBX1
PAX2
WNT7
DCN
THBS1
LSR1
ID3
TMEM47
MUC5AC
TGFBI
-log10(pvalue)
log2FoldChange
diffexpressed
DOWN
NO
UP