Project Report: Down Syndrome PBMC RNA-seq

Contents

Goal

The primary goal of this project was to master the end-to-end RNA-seq bioinformatics pipeline using a real-world dataset (GSE151282), 4 blood samples (out of 8).

Pipeline Overview

The analysis followed these steps:

1. **Quality Control:** FastQC/MultiQC (Raw & Trimmed reads).
2. **Alignment:** HISAT2 (Genome GRCh38).
3. **Quantification:** featureCounts.
4. **Exploratory Data Analysis** (EDA): Dimensionality reduction via PCA (after VST normalization) to inspect sample grouping and batch effects.
5. **Differential Expression Analysis** (DESeq2)

---

1. Data Processing and Quality Control

The analysis was performed on 4 paired-end samples from the NCBI GEO – GSE151282 project. The study design is balanced for both condition (Down Syndrome vs. Healthy) and sex.

Sample ID	Condition	Sex
SRR11856162	Down Syndrome	Male

Sample ID	Condition	Sex
SRR11856166	Down Syndrome	Female
SRR11856164	Healthy	Male
SRR11856165	Healthy	Female

1.1 Downloading Raw Data

The raw sequencing data (FASTQ files) were retrieved from the NCBI SRA database using the `sratoolkit`.

```
# Project directory: /mnt/e/Projects/down_syndrome_PBMC

# Download SRA files in paired-end format and compress them on the fly
while read srr; do
    fastq-dump --split-files --gzip $srr -O data/fastq-raw
done < data/fastq-raw/SRR_Acc_List.txt
```

1.2 Quality Control (FastQC, MultiQC)

To assess the quality of the raw reads, FastQC for individual sample metrics and MultiQC for an aggregated report were used.

```
# Running FastQC on raw data
fastqc data/fastq-raw/*.fastq.gz -o results/fastqc-raw

# Aggregating results
multiqc results/fastqc-raw -o results/multiqc-raw
```

1.3 Data Trimming (Trimmomatic)

Though the raw data showed high quality, a trial trimming using Trimmomatic was performed on all raw reads to evaluate if removing low-quality bases and adapters would improve the mapping rate.

```
# Example command for a single sample (SRR11856166)
java -jar trimmomatic-0.40.jar PE -threads 4 \
    data/fastq-raw/SRR11856166_1.fastq.gz \
    data/fastq-raw/SRR11856166_2.fastq.gz \
    data/fastq-trimmed/SRR11856166_1_paired.fastq.gz \
    data/fastq-trimmed/SRR11856166_1_unpaired.fastq.gz \
    data/fastq-trimmed/SRR11856166_2_paired.fastq.gz \
    data/fastq-trimmed/SRR11856166_2_unpaired.fastq.gz \
    ILLUMINACLIP:TruSeq3-PE.fa:2:30:10 \
    LEADING:3 TRAILING:3 SLIDINGWINDOW:4:20 MINLEN:36
```

1.4 Comparison and Decision

The quality reports for the trimmed data were rerun.

```
# QC for trimmed paired-end reads
fastqc data/fastq-trimmed/*_paired.fastq.gz -o results/fastqc-trimmed
multiqc results/fastqc-trimmed -o results/multiqc-trimmed
```

Since trimming provided no significant improvement to data quality, the raw reads were used for the alignment phase.

2. Alignment: HISAT2 (Genome GRCh38)

2.1 Reference Genome Preparation

To ensure high-quality mapping, GRCh38 (release 109) human primary assembly from Ensembl was used. This involved downloading the genomic sequences (FASTA) and gene annotations (GTF), followed by building a HISAT2-specific index.

```
# Downloading and preparing the reference genome
mkdir -p results/reference && cd results/reference

wget ftp://ftp.ensembl.org/pub/release-109/fasta/homo_sapiens/dna/Homo_sapiens.GRCh38.dna.primary_assembl
wget [https://ftp.ensembl.org/pub/release-109/gtf/homo_sapiens/Homo_sapiens.GRCh38.109.gtf.gz](https://

gunzip *.gz

# Building the HISAT2 index
hisat2-build Homo_sapiens.GRCh38.dna.primary_assembly.fa GRCh38_index
```

2.2 Alignment reading (HISAT2)

The alignment was performed with HISAT2, a splice-aware aligner.

```
# Mapping paired-end reads and converting to sorted BAM
mkdir -p results/hisat2

for fwd in data/fastq-raw/*_1.fastq.gz; do
    base=$(basename "$fwd" _1.fastq.gz)
    rev="data/fastq-raw/${base}_2.fastq.gz"

    hisat2 -p 4 -x results/reference/GRCh38_index \
        -1 "$fwd" -2 "$rev" | \
    samtools view -@ 4 -bS - | \
    samtools sort -@ 4 -o "results/hisat2/${base}_sorted.bam"

    samtools index "results/hisat2/${base}_sorted.bam"
done
```

3. Quantification (featureCounts)

To convert the aligned reads into a digital expression matrix, featureCounts (gene-level counting) from the Subread package was used. The tool aggregates the alignments to the gene level based on the provided GTF annotation.

```
# Summarizing paired-end reads to genomic features
mkdir -p results/counts

featureCounts -T 4 -p -t exon -g gene_id \
```

```
    -a results/reference/Homo_sapiens.GRCh38.109.gtf \
    -o results/counts/featurecounts_counts.txt \
    results/hisat2/*_sorted.bam
```

The output of this step is a raw count matrix (`results/counts/`) which serves as the primary input for differential expression analysis.

4. Exploratory Data Analysis (EDA)

4.1 Data Preparation in R

The raw count matrix was imported into R. Data cleaning was performed, including removing technical columns and renaming samples for clarity. A metadata table (`colData`) was constructed to define the experimental groups (Condition and Sex).

```
library(magrittr)
library(kableExtra)
library(DESeq2)
library(ggplot2)

# Load count data
count_data <- read.table("results/counts/featurecounts_counts.txt",
                         header = TRUE, row.names = 1, sep = "\t", comment.char = "#")

# Clean column names and select sample columns
count_data <- count_data[, 6:ncol(count_data)]
colnames(count_data) <- gsub("results.hisat2.|_sorted.bam", "", colnames(count_data))


# Metadata(colData) creation
coldata <- data.frame(
  row.names = colnames(count_data),
  condition = factor(c("Down", "Healthy", "Healthy", "Down")),
  sex = factor(c("male", "male", "female", "female"))
)

# Tables output
kable(head(count_data, 10), booktabs = TRUE, caption = "Raw Counts (Top 10)") %>%
  kable_styling(bootstrap_options = c("striped", "hover"),
                latex_options = c("striped", "scale_down", "hold_position"))
```

Table 2: Raw Counts (Top 10)

	SRR11856162	SRR11856164	SRR11856165	SRR11856166
ENSG00000160072	171	128	127	49
ENSG00000279928	8	46	1	0
ENSG00000228037	18	5	1	2
ENSG00000142611	10	2	0	0
ENSG00000284616	0	0	0	0
ENSG00000157911	79	32	23	4

ENSG00000269896	22	20	6	0
ENSG00000228463	9	29	1	0
ENSG00000260972	0	0	0	0
ENSG00000224340	0	0	0	0

```
kable(coldata, booktabs = TRUE, caption = "Sample Metadata") %>%
  kable_styling(bootstrap_options = c("striped", "hover"),
                latex_options = c("striped", "hold_position"))
```

Table 3: Sample Metadata

	condition	sex
SRR11856162	Down	male
SRR11856164	Healthy	male
SRR11856165	Healthy	female
SRR11856166	Down	female

```
# Initialize DESeqDataSet
# Design accounts for 'sex' as a covariate to isolate the 'condition' effect
dds <- DESeqDataSetFromMatrix(countData = count_data,
                              colData = coldata,
                              design = ~ sex + condition)

# Filter out low-expressed genes (sum of counts < 10) to reduce noise
keep <- rowSums(counts(dds)) >= 10
dds <- dds[keep, ]
```

4.2 Statistical Parameter Estimation

Before performing the deseq analysis, the size factors and dispersions were estimated. This is important for normalizing the data and modeling the technical noise inherent in RNA-seq experiments.

```
# Estimating size factors to normalize for sequencing depth
dds <- estimateSizeFactors(dds)

# Estimating dispersions to model gene-wise variability
dds <- estimateDispersions(dds)
```

4.3 Variance Stabilization Transformation (VST)

Why VST?

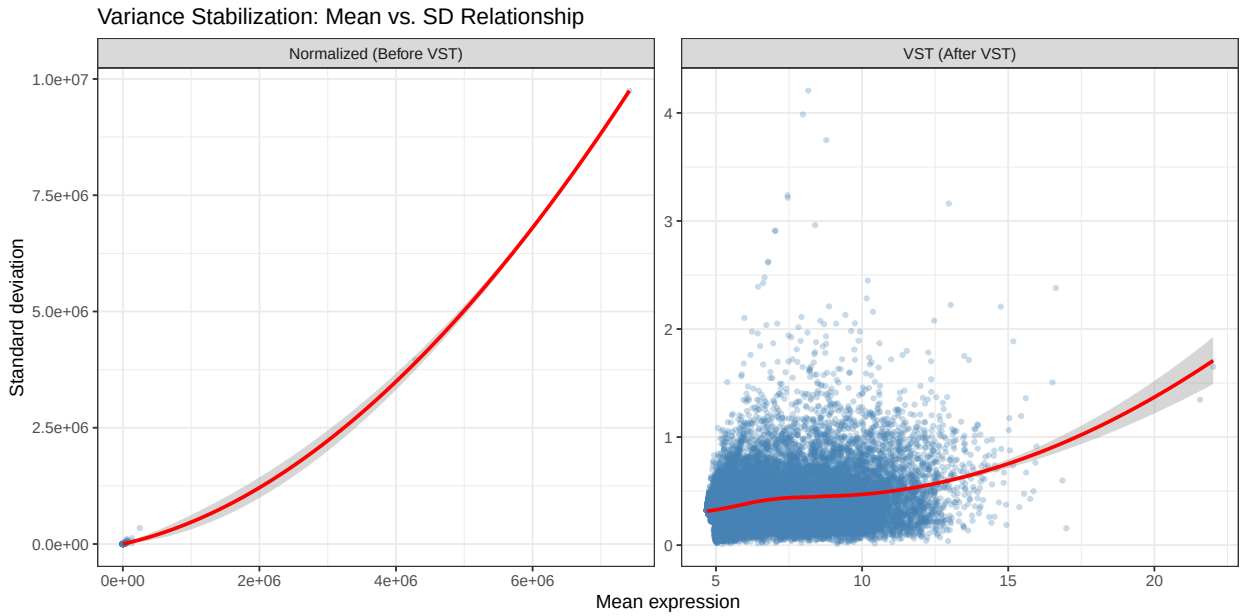
- Makes the variance independent of the mean.
- Prevents highly expressed genes from dominating the results during PCA.
- It is strictly used for visualization and QC, while the statistical testing (DESeq2) will still use the raw counts.

RNA-seq count data typically demonstrates heteroscedasticity where the variance increases with the mean expression level. To make the data suitable for exploratory graphics and clustering, we apply the VST). By setting `blind = FALSE`, the transformation accounts for the experimental design (`~ sex + condition`), we make sure that biological variance is preserved while technical noise is stabilized.

```
# Apply VST to the DESeqDataSet object
vsd <- varianceStabilizingTransformation(dds, blind = FALSE)

# Extract the transformed matrix for diagnostics
vsd_mat <- assay(vsd)
```

4.4 Homoscedasticity Check (Mean, SD Plots)

```
library(ggplot2)

# 1. Calculate stats for Normalized counts (Before VST)
raw_stats <- data.frame(
  mean = rowMeans(counts(dds, normalized=TRUE)),
  sd = apply(counts(dds, normalized=TRUE), 1, sd),
  type = "Normalized (Before VST)"
)

# 2. Calculate stats for VST data (After VST)
vst_stats <- data.frame(
  mean = rowMeans(vsd_mat),
  sd = apply(vsd_mat, 1, sd),
  type = "VST (After VST)"
)

combined_stats <- rbind(raw_stats, vst_stats)

# 3. Visualization with LOESS trend line
p_loess <- ggplot(combined_stats, aes(x = mean, y = sd)) +
  geom_point(alpha = 0.3, size = 1, color = "steelblue") +
  geom_smooth(method = "loess", color = "red") +
  facet_wrap(~type, scales = "free") +
  theme_bw() +
  labs(
    x = "Mean expression",
    y = "Standard deviation",
    title = "Variance Stabilization: Mean vs. SD Relationship"
  )

print(p_loess)
```



Interpretation:

On the left panel, we can see that the red line rises sharply. This means that the more a gene is expressed, the higher its absolute noise. So, when applying a PCA, genes with medium expression would be lost and the algorithm would only see the differences in genes with high counts (top right).

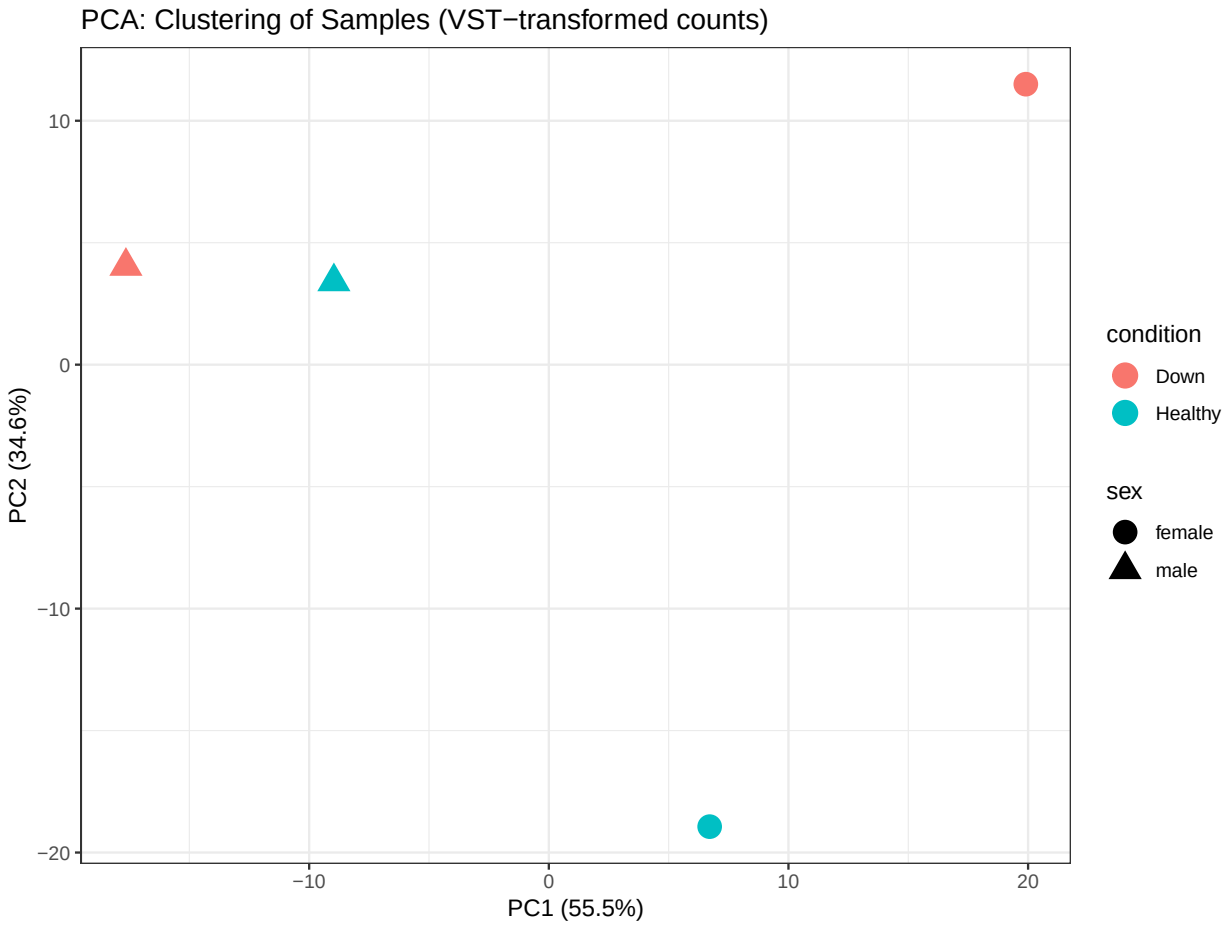
After applying VST, the red trend line indicates that the noise is now approximately the same for both weakly and highly expressed genes (homoscedasticity).

By applying VST, we have removed the technical dependency of noise on the mean value. The data is now distributed more symmetrically and in the PCA analysis every gene will have an equal weight.

4.5 PCA (Principal Component Analysis)

```
# Prepare data: Transpose so rows are samples and columns are genes
vsd_t <- t(vsd_mat)

# Calculate variance and filter out non-variable genes
gene_var <- apply(vsd_t, 2, var)
vsd_t <- vsd_t[, gene_var > 0]
gene_var <- gene_var[gene_var > 0]

# Select top 500 most variable genes for the analysis
top_genes <- order(gene_var, decreasing = TRUE)[1:500]
vsd_t <- vsd_t[, top_genes]

# Perform PCA (centering and scaling are applied)
pca_res <- prcomp(vsd_t, center = TRUE, scale. = TRUE)

# Calculate percentage of explained variance for each component
explained_var <- (pca_res$sdev^2) / sum(pca_res$sdev^2) * 100

# Prepare data frame for visualization
pca_df <- data.frame(
```

```
  PC1 = pca_res$x[, 1],
  PC2 = pca_res$x[, 2],
  condition = coldata$condition,
  sex = coldata$sex
)

p_pca <- ggplot(pca_df, aes(x = PC1, y = PC2, color = condition, shape = sex)) +
  geom_point(size = 5) +
  labs(
    x = paste0("PC1 (", round(explained_var[1], 1), "%)"),
    y = paste0("PC2 (", round(explained_var[2], 1), "%)"),
    title = "PCA: Clustering of Samples (VST-transformed counts)"
  ) +
  theme_bw()

print(p_pca)
```



Interpretation:

The PCA plot demonstrates a clear separation of samples.

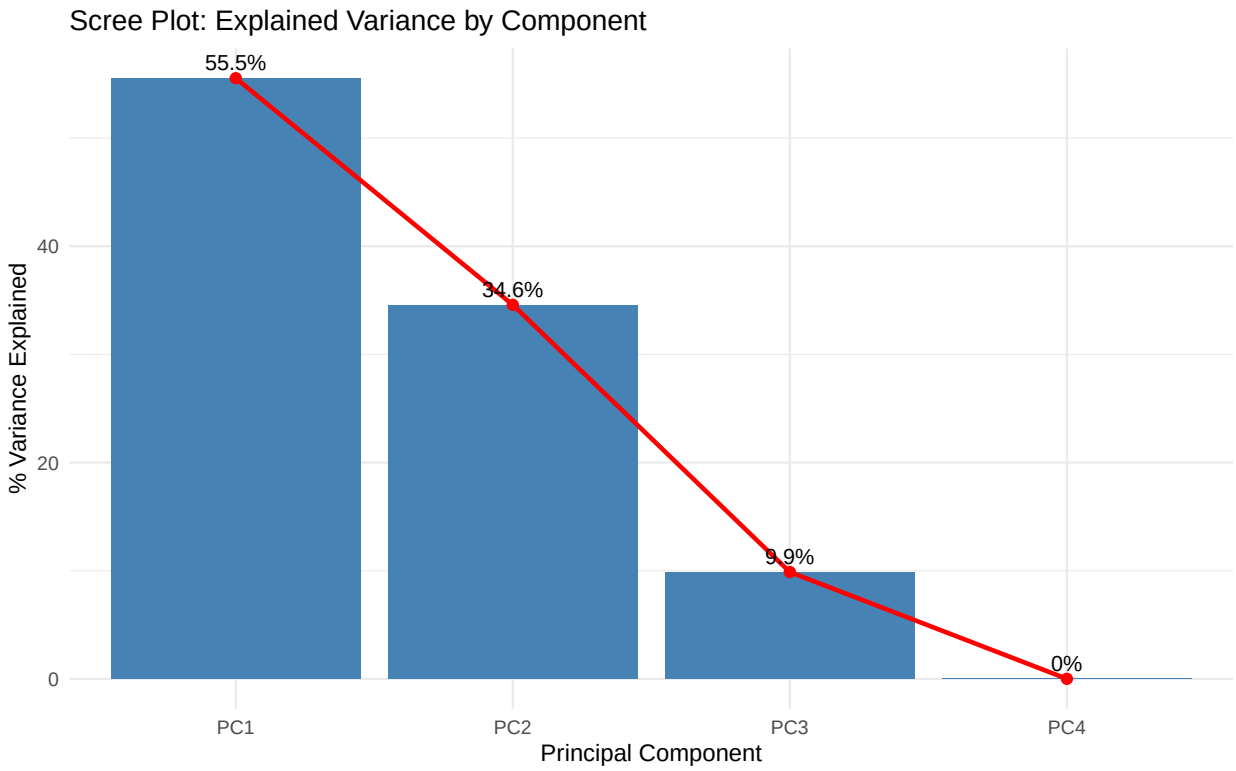
-> PC2 (34.6%) separates the samples by sex (males top, females bottom).

-> PC1 (55.5%) captures the disease effect, showing that Down Syndrome samples (red) deviate significantly from the healthy baseline (blue).

Interesting observation, the disease effect seems like to be sex-specific: the male Down syndrome sample shifts further left, while the female Down syndrome sample shifts far to the right.

Scree Plot helps determine the importance of each PC and ensures that the first two components are sufficient for downstream analysis.

```
pca_var <- pca_res$sdev^2
explained_var <- pca_var / sum(pca_var) * 100

scree_df <- data.frame(
  PC = factor(paste0("PC", 1:length(explained_var)),
              levels = paste0("PC", 1:length(explained_var))),
  Variance = explained_var
)


p_scree <- ggplot(scree_df, aes(x = PC, y = Variance)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_line(aes(y = Variance, group = 1), color = "red", linewidth = 1) +
  geom_point(aes(y = Variance), color = "red", size = 2) +
  geom_text(aes(label = paste0(round(Variance, 1), "%")),
            vjust = -0.5, size = 3.5) +
  labs(
    x = "Principal Component",
    y = "% Variance Explained",
    title = "Scree Plot: Explained Variance by Component"
  ) +
  theme_minimal()

print(p_scree)
```



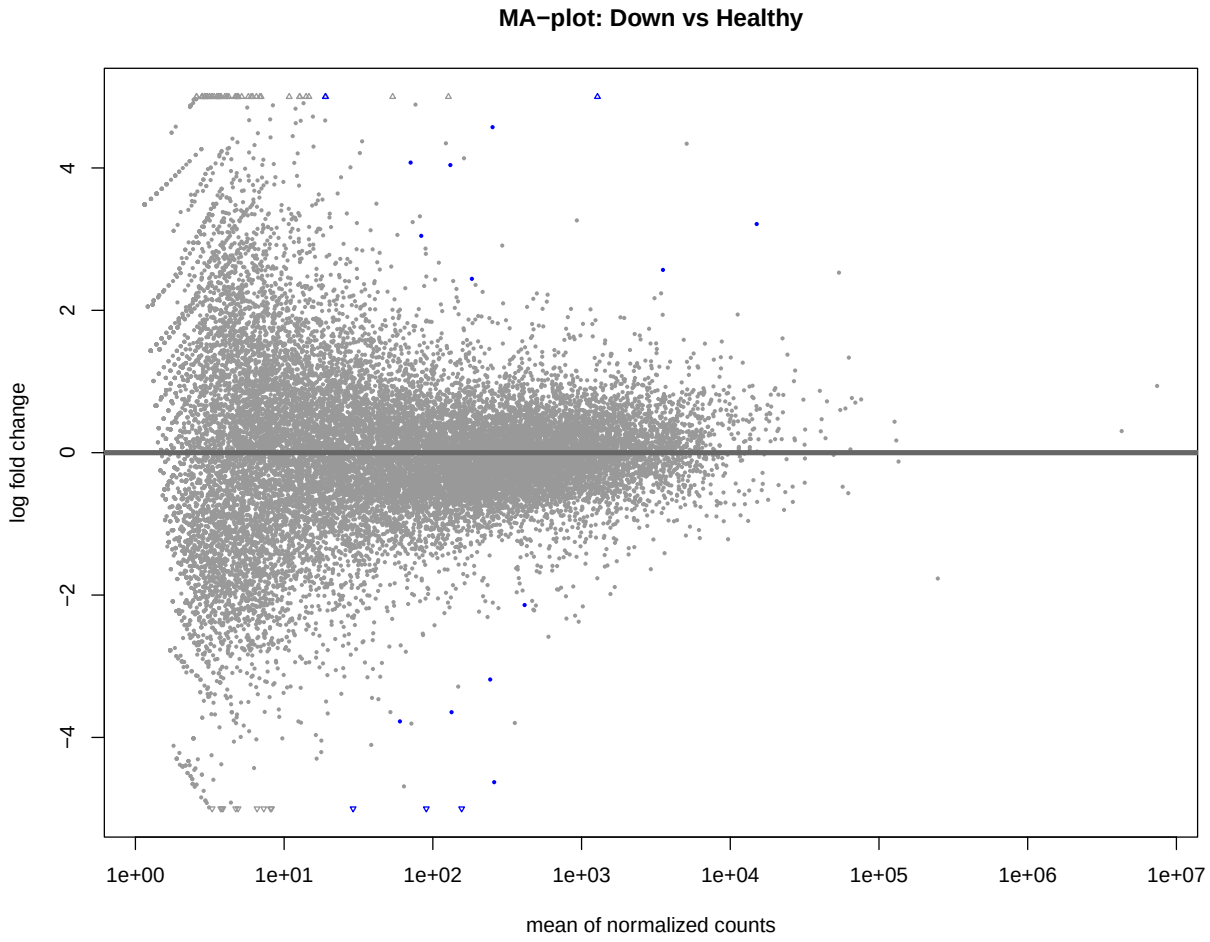
Interpretation:

The first two components (PC1 and PC2) explain 90.1% of the total variance in the dataset. This indicates that the primary biological factors (condition `Down Syndrome` and `sex`) dominate the transcriptomic profile.

5. Differential Expression Analysis (DESeq2)

```
library(magrittr)
library(kableExtra)

dds <- DESeq(dds)
res <- results(dds, contrast=c("condition", "Down", "Healthy"))

# Genes annotation (ENSG -> Symbol)
library(org.Hs.eg.db)

res$symbol <- mapIds(org.Hs.eg.db,
                     keys = rownames(res),
                     column = "SYMBOL",
                     keytype = "ENSEMBL",
                     multiVals = "first")

# Sort
resOrdered <- res[order(res$padj), ]

write.csv(as.data.frame(resOrdered), file = "results/DESeq2_results_annotated.csv")
```

```r
# MA-Plot
plotMA(res, ylim=c(-5,5), main="MA-plot: Down vs Healthy")
```



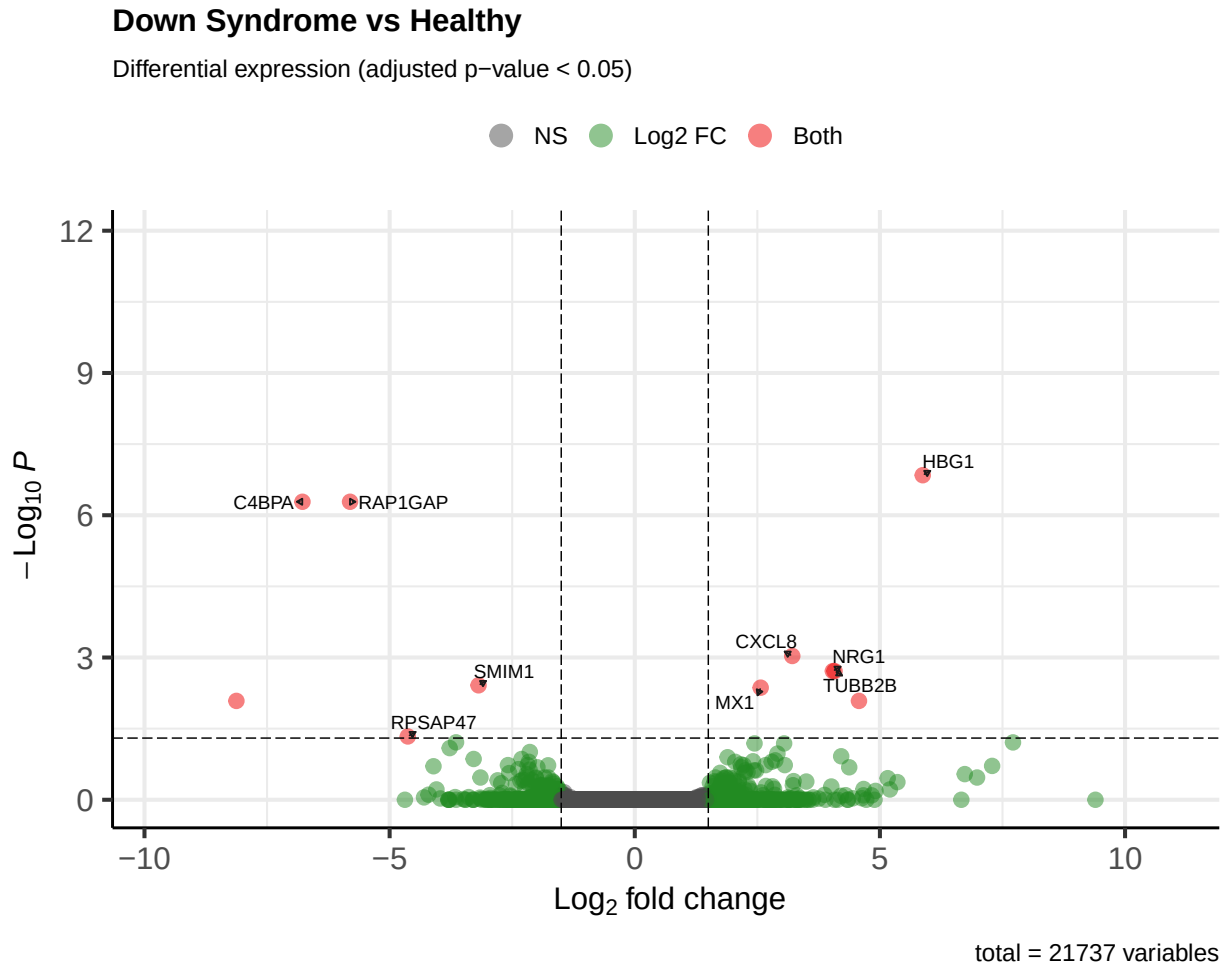
Interpretation:

The MA Plot shows the relationship between the average expression of genes (X-axis) and the change in expression between Down Syndrome and Healthy samples (Y-axis). Most genes (gray dots) are clustered around the horizontal zero line.It means our data normalization worked correctly and there is no systematic bias.

The blue dots represent genes that are statistically significant. The triangles at the top and bottom edges are genes with very high fold changes, representing the strongest candidates for biological markers.

```r
# Volcano Plot
library(EnhancedVolcano)

p_volcano <- EnhancedVolcano(res,
                             lab = res$symbol,
                             x = 'log2FoldChange',
                             y = 'padj',
                             title = 'Down Syndrome vs Healthy',
                             subtitle = 'Differential expression (adjusted p-value < 0.05)',
```

```
                            pCutoff = 0.05,
                            FCcutoff = 1.5,
                            pointSize = 3.0,
                            labSize = 4.0,
                            col = c('grey30', 'forestgreen', 'royalblue', 'red2'),
                            legendLabels = c('NS', 'Log2 FC', 'Adjusted p-value', 'Both'),
                            drawConnectors = TRUE,
                            widthConnectors = 0.5)

print(p_volcano)
```



Interpretation:

The Volcano Plot displays 21,737 variables (genes). Each point on the graph represents a single gene:

-> X-axis (Log2 Fold Change) represents the direction and strength of the expression change. Points to the right of 0 are up-regulated in Down Syndrome.Points to the left of 0 are down-regulated.

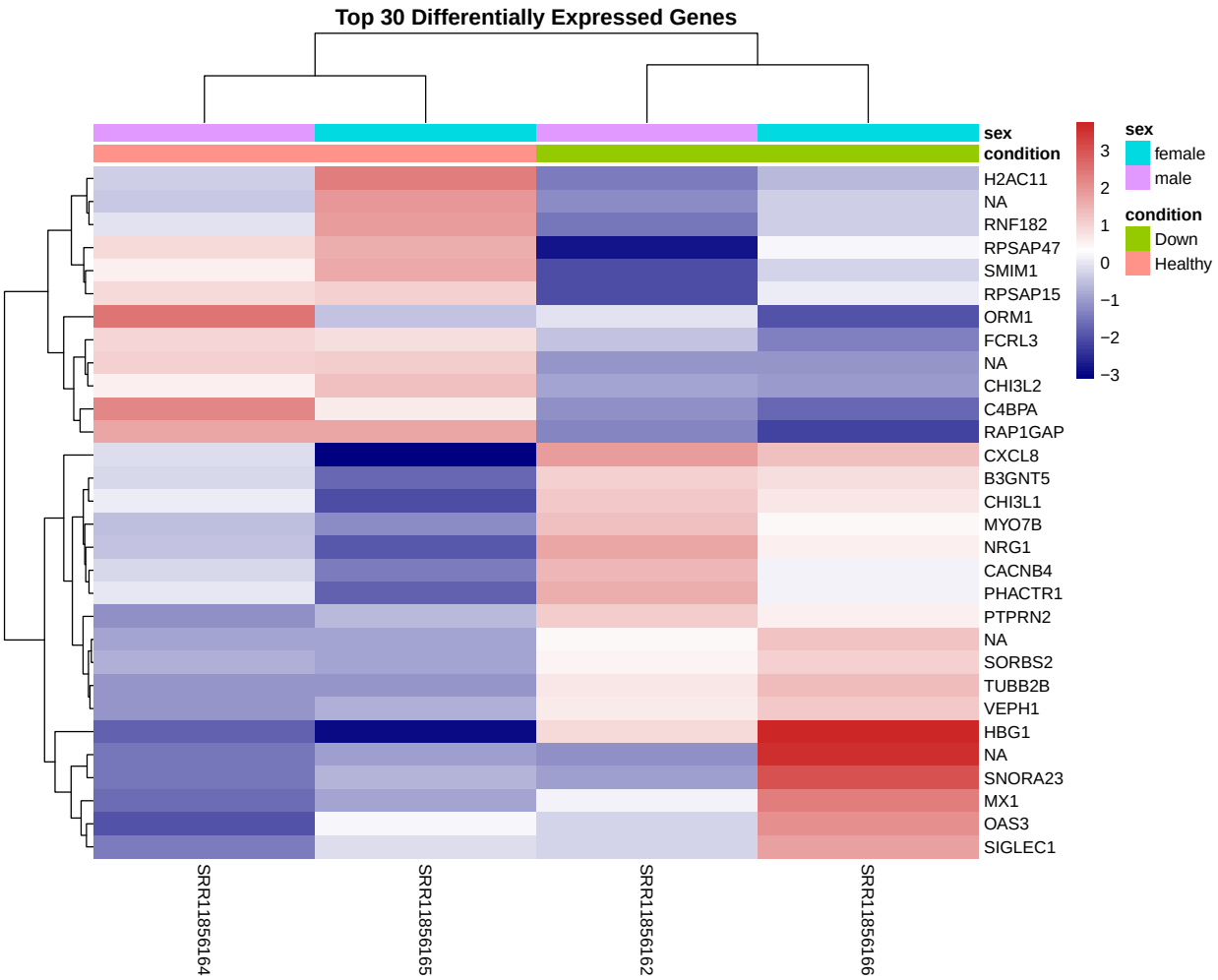
-> Y-axis (-Log10 adjusted P-value) represents statistical confidence. The higher the point, the more reliable the result.

Biological Insights:

The highest peak on the Y-axis is HBG1, indicates a very strong and consistent shift in hemoglobin-related gene expression.

Genes like C4BPA and RAP1GAP are found in the top-left quadrant, showing a significant decrease in expression compared to healthy controls.

```
# Heatmap
library(pheatmap)
library(grid)

# Top 30 genes
top30_genes <- head(order(res$padj), 30)

mat <- vsd_mat[top30_genes, ]
rownames(mat) <- res$symbol[top30_genes]

# Centering
mat <- mat - rowMeans(mat)

# Columns annotation
df_anno <- as.data.frame(colData(dds)[, c("condition", "sex")])

pheatmap(mat,
         annotation_col = df_anno,
         main = "Top 30 Differentially Expressed Genes",
         clustering_distance_rows = "euclidean",
         clustering_method = "complete",
         color = colorRampPalette(c("navy", "white", "firebrick3"))(100),
         border_color = NA,
         fontsize = 10,
         silent = FALSE)
```



Interpretation:

The hierarchical clustering heatmap of the top 30 differentially expressed genes shows a clear separation between Down Syndrome and Healthy samples. The dendrogram at the top correctly clusters individuals based on their clinical condition, regardless of sex.

A cluster of genes (including H2AC11 and C4BPA) are consistently down-regulated in the Down Syndrome group and a cluster (including MX1, NRG1, and HBG1) shows strong up-regulation.