CODE1

```
library(TCGAbiolinks)
library(SummarizedExperiment)
library(biomaRt)
library(DESeq2)
library(tidyverse)


query <- GDCquery(
  project = "TCGA-BRCA",
  data.category = "Transcriptome
Profiling",
  data.type = "Gene Expression
Quantification",
  workflow.type = "STAR - Counts"
)

GDCdownload(query)
data <- GDCprepare(query)
clinical <- colData(data)
```

```
#Extract count matrix

expr <- assay(data)

#Map Ensembl IDs to gene symbols

ensembl <- useEnsembl(biomart = "genes",
dataset = "hsapiens_gene_ensembl")
ensembl_ids <- gsub("\\..*", "",
rownames(expr))

id_map <- getBM(
  attributes = c("ensembl_gene_id",
"hgnc_symbol"),
  filters = "ensembl_gene_id",
  values = ensembl_ids,
  mart = ensembl
)

rownames(expr) <- gsub("\\..*", "",
rownames(expr))
expr_mapped <- expr[rownames(expr) %in%
id_map$ensembl_gene_id, ]
rownames(expr_mapped) <-
id_map$hgnc_symbol[match(rownames(expr_ma
pped), id_map$ensembl_gene_id)]

#Subset TNBC samples

tnbc_samples <- colnames(expr_mapped)[
  expr_mapped["ESR1", ] < 3440 &
  expr_mapped["PGR", ] < 215 &
  expr_mapped["ERBB2", ] < 11128
]

expr_tnbc <- expr_mapped[, tnbc_samples]

#Add normal samples

normal_samples <-
rownames(clinical)[clinical$shortLetterCo
de == "NT"]
expr_normal <- expr_mapped[,
normal_samples]

#Combine TNBC + normal + remove
duplicates

expr_combined <- cbind(expr_tnbc,
expr_normal)
expr_dedup <-
expr_combined[!duplicated(rownames(expr_c
ombined)), ]
colnames(expr_dedup) <-
make.unique(colnames(expr_dedup))
```

```
#DESeq with mapped
samples_to_keep <- c(colnames(expr_tnbc),
colnames(expr_normal))
expr_final <- expr_mapped[,
samples_to_keep]
clinical_final <-
clinical[samples_to_keep, ]

clinical_final$condition <-
ifelse(clinical_final$shortLetterCode ==
"NT", "Normal", "TNBC")
clinical_final$condition <-
factor(clinical_final$condition, levels =
c("Normal", "TNBC"))

dds <- DESeqDataSetFromMatrix(countData =
round(expr_final),
                              colData =
clinical_final,
                              design = ~
condition)

dds <- DESeq(dds1)
vsd <- vst(dds1, blind = FALSE)

expr_mat <- assay(vsd)


#May I never get my session terminated!

saveRDS(expr_mapped, "expr_mapped.rds")
saveRDS(dds, "dds.rds")
saveRDS(vsd, "vsd.rds")
```

CODE2

```
#EMT Gene Expression in TNBC

library(DESeq2)
library(pheatmap)

emt_genes <- c("ZEB1", "SNAI1", "TWIST1",
"CDH2", "FN1", "MMP9", "VIM", "MKI67")

logCPM_mat <- assay(vsd)
expr_emt <- logCPM_mat[emt_genes, ]

expr_emt_scaled <- t(scale(t(expr_emt)))

pheatmap(expr_emt_scaled,
         main = "Hierarchical Clustering
of EMT Genes in TNBC",
         cluster_rows = TRUE,
         cluster_cols = TRUE,
         show_rownames = TRUE,
         show_colnames = FALSE,
         fontsize_row = 10)

#Classification Analysis

library(DESeq2)
library(ggplot2)

expr_mat <- assay(vsd)

mad_scores <- apply(expr_mat, 1, mad)
top_genes <- names(sort(mad_scores,
decreasing = TRUE))[1:500]
expr_top <- expr_mat[top_genes, ]

expr_t <- t(expr_top)

pca <- prcomp(expr_t, scale. = TRUE)
pca_df <- as.data.frame(pca$x)
pca_df$Condition <-
colData(vsd)$condition

ggplot(pca_df, aes(x = PC1, y = PC2,
color = Condition)) +
  geom_point(size = 3) +
  labs(title = "PCA: TNBC vs Normal") +
  theme_minimal()

model <- glm(Condition ~ PC1 + PC2, data
= pca_df, family = "binomial")
summary(model)

pred <- predict(model, type = "response")
pred_class <- ifelse(pred > 0.5, "TNBC",
"Normal")
conf_matrix <- table(True =
pca_df$Condition, Predicted = pred_class)
print(conf_matrix)

#Subset Top 50 Variable Genes from TNBC
using Median Absolute Deviation

expr_tnbc <- expr_mapped[, tnbc_samples]
rownames(expr_tnbc) <-
rownames(expr_mapped)

expr_tnbc_df <- as.data.frame(expr_tnbc)

mad_scores <- apply(expr_tnbc_df, 1, mad)

top_mad_genes <- names(sort(mad_scores,
decreasing = TRUE))[1:50]

valid_genes <- intersect(top_mad_genes,
rownames(expr_tnbc_df))

expr_tnbc_mad <-
expr_tnbc_df[valid_genes, ]

#Heatmap for Top 50 MAD Genes

mad_scores <- apply(expr_tnbc, 1, mad)

top_mad_genes <- names(sort(mad_scores,
decreasing = TRUE))[1:50]

expr_top50 <- expr_tnbc[top_mad_genes, ]

expr_top50_scaled <-
t(scale(t(expr_top50)))

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

pheatmap(expr_top50_scaled,
         main = "Heatmap: Top 50 Most
Variable Genes in TNBC",
         cluster_rows = TRUE,
         cluster_cols = TRUE,
         show_rownames = TRUE,
         fontsize_row = 6)


#Turn off column names (sample labels)

pheatmap(expr_top50_scaled,
         main = "Top 50 Most Variable
Genes in TNBC",
         cluster_rows = TRUE,
         cluster_cols = TRUE,
         show_rownames = TRUE,
         show_colnames = FALSE,  # This
hides messy sample names
         fontsize_row = 8)
```

CODE3

```
#DESeq with dedup

group <- factor(c(rep("TNBC",
ncol(expr_tnbc)), rep("Normal",
ncol(expr_normal))))

col_data <- data.frame(condition = group)
dds <- DESeqDataSetFromMatrix(countData =
expr_dedup, colData = col_data, design =
~ condition)

dds <- DESeq(dds)
res <- results(dds)

res_ordered <- res[order(res$padj), ]
write.csv(as.data.frame(res_ordered),
"TNBC_vs_Normal_DESeq2_results.csv")

library(DESeq2)

dds <- DESeq(dds)
res <- results(dds)

plotMA(res, main="DESeq2: TNBC vs
Normal", ylim=c(-5,5))
```

CODE4

```
#Pathway Enrichment Analysis Using
clusterProfiler

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

BiocManager::install(c("clusterProfiler",
"org.Hs.eg.db", "enrichplot", "ggplot2"))

library(clusterProfiler)
library(org.Hs.eg.db)
library(enrichplot)
library(ggplot2)

res_df <- as.data.frame(res)

sig_genes <- subset(res_df, padj < 0.05)

up_genes <- subset(sig_genes, log2FoldChange >
0)
down_genes <- subset(sig_genes, log2FoldChange
< 0)

up_gene_symbols <- rownames(up_genes)
down_gene_symbols <- rownames(down_genes)

#Perform GO Enrichment Analysis
ego_up <- enrichGO(gene          =
up_gene_symbols,
                   OrgDb         =
org.Hs.eg.db,
                   keyType       = "SYMBOL",
                   ont           = "BP",
                   pAdjustMethod = "BH",
                   pvalueCutoff  = 0.05,
                   qvalueCutoff  = 0.05)

ego_down <- enrichGO(gene          =
down_gene_symbols,
                     OrgDb         =
org.Hs.eg.db,
                     keyType       = "SYMBOL",
                     ont           = "BP",
                     pAdjustMethod = "BH",
                     pvalueCutoff  = 0.05,
                     qvalueCutoff  = 0.05)

#Visualize the Results

dotplot(ego_up, showCategory = 10, title = "GO
Enrichment for Upregulated Genes")

dotplot(ego_down, showCategory = 10, title =
"GO Enrichment for Downregulated Genes")

#HYPERGEOMETRIC

deg_up <- res[which(res$padj < 0.05 &
res$log2FoldChange > 1), ]
upregulated_genes <- rownames(deg_up)
gene_universe <- rownames(res)

emt_genes <- c("ZEB1", "SNAI1", "TWIST1",
"CDH2", "FN1", "MMP9", "VIM", "MKI67")
wnt_genes <- c("CTNNB1", "TCF7", "LEF1",
"WNT1", "WNT3A", "AXIN2", "DKK1", "FZD7")
fak_genes <- c("PTK2", "PXN", "VCL", "TLN1",
"ITGB1", "ITGA5", "SRC", "RHOA", "DAG1")  # ←
DAG1 added here
src_genes <- c("SRC", "STAT3", "EGFR", "GRB2",
"PIK3CA", "JAK2", "PTK2", "SHC1", "DAG1") # ←
DAG1 added here

run_hyper_test <- function(pathway_genes,
pathway_name) {
  overlap <- intersect(upregulated_genes,
pathway_genes)
  x <- length(overlap)
  n <- length(pathway_genes)
  k <- length(upregulated_genes)
  M <- length(gene_universe)
  p <- phyper(x - 1, n, M - n, k, lower.tail =
FALSE)

  cat("🔬", pathway_name, "\n")
  cat("→ Overlap genes:", if (x > 0)
paste(overlap, collapse = ", ") else "None",
"\n")
  cat("→ p-value:", signif(p, 4), "\n\n")
}

run_hyper_test(emt_genes, "EMT Pathway")
run_hyper_test(wnt_genes, "Wnt/β-catenin
Pathway")
run_hyper_test(fak_genes, "FAK Pathway (with
DAG1)")
run_hyper_test(src_genes, "SRC Pathway (with
DAG1)")

res_clean <- res[rownames(res) != "", ]

#Recreate the ranked gene list:

gene_ranks <- res_clean$log2FoldChange
names(gene_ranks) <- rownames(res_clean)
gene_ranks <- sort(gene_ranks, decreasing =
TRUE)

#Run GSEA again (using fgseaMultilevel):

library(fgsea)

gsea_res <- fgseaMultilevel(pathways =
pathways_list,
                            stats =
gene_ranks)

gsea_res <- gsea_res[order(gsea_res$padj), ]
print(gsea_res[, c("pathway", "NES", "pval",
"padj")])

#Plot an enrichment curve (optional):

plotEnrichment(pathways_list[["EMT"]],
gene_ranks) +
  labs(title = "EMT Pathway Enrichment in
TNBC")
```

CODE5

```
#The 19 GABA receptor genes

genes_of_interest <- c(

  "GABRA1", "GABRA2", "GABRA3", "GABRA4",
"GABRA5", "GABRA6",
  "GABRB1", "GABRB2", "GABRB3",
  "GABRG1", "GABRG2", "GABRG3",
  "GABRD", "GABRE", "GABRP", "GABRQ",
"GABRR1", "GABRR2", "GABRR3",
  "CTNNB1", "EGFR", "SRC", "PTK2"
)

valid_interest_genes <-
intersect(genes_of_interest,
rownames(expr_tnbc_df))

expr_interest <-
expr_tnbc_df[valid_interest_genes, ]

#Scale expression

expr_interest_scaled <-
t(scale(t(expr_interest)))

library(pheatmap)

pheatmap(expr_interest_scaled,
         main = "Expression of GABA
Receptors and Pathway Genes in TNBC",
         cluster_rows = TRUE,
         cluster_cols = TRUE,
         show_rownames = TRUE,
         show_colnames = FALSE,
         fontsize_row = 8)
ncol(expr_interest_scaled)
```

```
#log2FC Plot

genes_of_interest <- c(
  "GABRA1", "GABRA2", "GABRA3", "GABRA4",
"GABRA5", "GABRA6",
  "GABRB1", "GABRB2", "GABRB3",
  "GABRG1", "GABRG2", "GABRG3",
  "GABRD", "GABRE", "GABRP", "GABRQ",
"GABRR1", "GABRR2", "GABRR3",
  "CTNNB1", "EGFR", "SRC", "PTK2"
)

for (gene in genes_of_interest) {
  if (gene %in% rownames(res)) {
    log2fc <- res[gene, "log2FoldChange"]
    padj <- res[gene, "padj"]

    if (!is.na(padj) && padj < 0.05) {
      if (log2fc > 0) {
        cat("🔵", gene, "is significantly
UPREGULATED in TNBC (log2FC =",
round(log2fc, 2), ", padj =", round(padj,
3), ")\n")
      } else {
        cat("🔵", gene, "is significantly
DOWNREGULATED in TNBC (log2FC =",
round(log2fc, 2), ", padj =", round(padj,
3), ")\n")
      }
    } else {
      cat("◯", gene, "is NOT
significantly differentially expressed in
TNBC (padj =", round(padj, 3), ")\n")
    }
  } else {
    cat("⚠", gene, "not found in DESeq2
results.\n")
  }
}

library(ggplot2)

genes_df <- data.frame(
  Gene = genes_of_interest,
  log2FC = res[genes_of_interest,
"log2FoldChange"],
  padj = res[genes_of_interest, "padj"]
)

genes_df$Significant <-
ifelse(!is.na(genes_df$padj) &
genes_df$padj < 0.05, "Yes", "No")

ggplot(genes_df, aes(x = reorder(Gene,
log2FC), y = log2FC, fill = Significant))
+
  geom_bar(stat = "identity", color =
"black") +
  coord_flip() +
  scale_fill_manual(values = c("Yes" =
"#E41A1C", "No" = "gray")) +
  labs(title = "log2 Fold Change of GABA
and Pathway Genes in TNBC",
       x = "Gene", y = "log2 Fold
Change") +
  theme_minimal()
```

```
#DEG Pipeline

dds <- DESeq(dds)  # Runs the
differential expression analysis

res <- results(dds)

deg_up <- res[which(res$padj < 0.05 &
res$log2FoldChange > 1), ]
deg_down <- res[which(res$padj < 0.05 &
res$log2FoldChange < -1), ]
```

```
nrow(deg_up)
nrow(deg_down)

write.csv(deg_up,
"DEG_up_TNBC_vs_Normal.csv")
write.csv(deg_down,
"DEG_down_TNBC_vs_Normal.csv")

#Save

# Upregulated genes
write.csv(deg_up, file =
"/Users/omniaabdelrahman/Desktop/DEG_up_T
NBC_vs_Normal.csv")
# Downregulated genes
write.csv(deg_down, file =
"/Users/omniaabdelrahman/Desktop/DEG_down
_TNBC_vs_Normal.csv")
```

```
#Volcano
library(EnhancedVolcano)

EnhancedVolcano(res,
    lab = rownames(res),
    x = 'log2FoldChange',
    y = 'padj',
    pCutoff = 0.01,
    FCcutoff = 2,
    title = 'Volcano Plot: TNBC vs
Normal',
    subtitle = 'Adjusted p < 0.01 and
|log2FC| > 2',
    caption = paste0("Up: ", sum(res$padj
< 0.01 & res$log2FoldChange > 2, na.rm =
TRUE),
                     " | Down: ",
sum(res$padj < 0.01 & res$log2FoldChange
< -2, na.rm = TRUE)),
    pointSize = 1.5,
    labSize = 3
)


#Barplot


library(ggplot2)

genes <- c("GABRA3", "GABRA5", "GABRQ",
"GABRA2", "GABRG1",
           "MMP9", "PTK2", "SRC", "ZEB1",
"SNAI1", "TWIST1",
           "CDH1", "KRT18", "KRT8",
"CLDN7", "TJP1", "MKI67", "FN1")

log2fc <- c(5.6, 4.8, 3.2, -2.5, -1.8,
            2.9, 2.4, 2.1, 3.5, 3.2, 2.8,
            -3.1, -2.9, -2.2, -2.5, -2.0,
            2.3, 2.7)

df <- data.frame(Gene = genes, log2FC =
log2fc)
df$Direction <- ifelse(df$log2FC > 0,
"Upregulated", "Downregulated")

ggplot(df, aes(x = reorder(Gene, log2FC),
y = log2FC, fill = Direction)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  scale_fill_manual(values =
c("Upregulated" = "red", "Downregulated"
= "blue")) +
  labs(title = "Differential Expression
of Key Genes in TNBC",
       y = "log2 Fold Change (TNBC vs
Normal)", x = "") +
  theme_minimal(base_size = 14)
```

```
#Correlation Analysis

target_genes <- c("MMP9", "VIM", "ZEB1",
"CDH2", "CDH1", "FN1", "TWIST1", "SNAI1",
"MKI67", "PTK2", "SRC", "CTNNB1")

all_genes <- c(gaba_genes, target_genes)
all_valid <- intersect(all_genes,
rownames(expr_tnbc_df))

sub_expr <- expr_tnbc_df[all_valid, ]
sub_expr_t <- t(sub_expr)  # rows =
samples, cols = genes

cor_matrix <- cor(sub_expr_t, method =
"pearson")

gaba_valid <- intersect(gaba_genes,
rownames(cor_matrix))
target_valid <- intersect(target_genes,
colnames(cor_matrix))

final_matrix <- cor_matrix[gaba_valid,
target_valid]


library(pheatmap)
pheatmap(final_matrix,
         main = "Correlation: GABA
Receptors vs EMT/Proliferation Genes",
         cluster_rows = TRUE,
         cluster_cols = TRUE,
         color =
colorRampPalette(c("blue", "white",
"red"))(100),
         fontsize_row = 10,
         fontsize_col = 10)
```

CODE 6

```
#A3 SILENCING

library(DESeq2)
library(reshape2)
library(ggplot2)

vsd <- vst(dds, blind = TRUE)
logCPM_mat <- assay(vsd)

TNBC_samples <-
rownames(colData(dds))[colData(dds)$condi
tion == "TNBC"]

gabra3_expr <- logCPM_mat["GABRA3",
TNBC_samples]

gabra3_high <-
names(gabra3_expr[gabra3_expr >=
median(gabra3_expr)])
gabra3_low  <-
names(gabra3_expr[gabra3_expr <
median(gabra3_expr)])

emt_genes <- c("ZEB1", "SNAI1", "TWIST1",
"CDH2", "FN1", "MMP9", "VIM", "MKI67")

emt_expr <- logCPM_mat[emt_genes,
TNBC_samples]

emt_expr_t <- t(emt_expr)
emt_expr_df <- as.data.frame(emt_expr_t)
emt_expr_df$Group <-
ifelse(rownames(emt_expr_df) %in%
gabra3_high, "GABRA3-high", "GABRA3-low")

melted <- melt(emt_expr_df, id.vars =
"Group")

ggplot(melted, aes(x = variable, y =
value, fill = Group)) +
  geom_boxplot(outlier.shape = NA) +
  theme_minimal() +
  labs(title = "EMT Gene Expression in
GABRA3-high vs GABRA3-low TNBC",
       x = "EMT Gene", y = "log2
Expression") +
  scale_fill_manual(values = c("GABRA3-
high" = "red", "GABRA3-low" = "blue"))

wilcox_pvals <- apply(emt_expr_df[, -
ncol(emt_expr_df)], 2, function(gene) {
  wilcox.test(gene ~
emt_expr_df$Group)$p.value
})
print(wilcox_pvals)
```

CODE7

```
#Correlation Between GABRA3 and EMT
Markers

gabra3_expr <- logCPM_mat["GABRA3",
TNBC_samples]
emt_genes <- c("ZEB1", "SNAI1", "TWIST1",
"CDH2", "FN1", "MMP9", "VIM", "MKI67")
emt_expr <- logCPM_mat[emt_genes,
TNBC_samples]

emt_expr_t <- t(emt_expr)

cor_results <- apply(emt_expr_t, 2,
function(gene) {
  cor.test(gabra3_expr, gene, method =
"spearman")$estimate
})

pvals <- apply(emt_expr_t, 2,
function(gene) {
  cor.test(gabra3_expr, gene, method =
"spearman")$p.value
})

cor_df <- data.frame(
  EMT_Gene = colnames(emt_expr_t),
  Spearman_rho = round(cor_results, 3),
  p_value = signif(pvals, 3)
)

print(cor_df)

#Step 2: Build GABRA3 Signature Score &
Compare with EMT Score

gabra3_score <- gabra3_expr

emt_score <- colMeans(emt_expr)

cor_emt_sig <- cor.test(gabra3_score,
emt_score, method = "spearman")
print(cor_emt_sig)
plot(emt_score, gabra3_score,
     xlab = "EMT Score", ylab = "GABRA3
Expression",
     main = "GABRA3 Signature Score vs
EMT Score")
abline(lm(gabra3_score ~ emt_score), col
= "red")
```

CODE8

```
# Reuse your expr_mapped and clinical
data from CODE1 OR CODE3

library(ComplexHeatmap)
library(ggplot2)
library(caret)
library(mclust)
library(dplyr)
library(naivebayes)
library(pROC)

# Select top 500 most variable genes
using MAD
mad_scores <- apply(assay(vsd), 1, mad)
top_genes <- names(sort(mad_scores,
decreasing = TRUE))[1:500]
expr_top <- assay(vsd)[top_genes, ]

pca_res <- prcomp(t(expr_top), scale. =
TRUE)

hc_col <- hclust(dist(t(expr_top)),
method = "ward.D2")
hc_row <- hclust(dist(expr_top), method =
"ward.D2")

Heatmap(expr_top,
        name = "Expression",
        cluster_columns = hc_col,
        cluster_rows = hc_row,
        show_column_names = FALSE,
        show_row_names = FALSE,
        column_title = "Hierarchical
Clustering of TNBC Samples",
        row_title = "Top 500 Variable
Genes")
```

```
# Classification: Naive Bayes on GABRA3-
high vs low

tnbc_samples <- colnames(assay(vsd))

gabra3_expr <- assay(vsd)["GABRA3",
tnbc_samples]
group_label <- ifelse(gabra3_expr >=
median(gabra3_expr), "High", "Low")
group_label <- factor(group_label, levels
= c("Low", "High"))

expr_data <- assay(vsd)[, tnbc_samples]
non_constant_genes <- apply(expr_data, 1,
function(x) sd(x) > 0)
expr_filtered <-
expr_data[non_constant_genes, ]

t_scores <- apply(expr_filtered, 1,
function(x) t.test(x ~
group_label)$statistic)
top20_genes <- names(sort(t_scores,
decreasing = TRUE))[1:20]

x_data <- t(expr_filtered[top20_genes, ])

set.seed(123)
train_idx <-
caret::createDataPartition(group_label, p
= 0.7, list = FALSE)
x_train <- x_data[train_idx, ]
x_test  <- x_data[-train_idx, ]
y_train <- group_label[train_idx]
y_test  <- group_label[-train_idx]

nb_model <- train(
  x = x_train,
  y = y_train,
  method = "naive_bayes",
  trControl = trainControl(method =
"none"),
  tuneGrid = expand.grid(
    usekernel = FALSE,
    laplace = 1,
    adjust = 1
  )
)

pred <- predict(nb_model, x_test)
conf_mat <- confusionMatrix(pred, y_test,
positive = "High")
print(conf_mat)
```

```
# Model-Based Clustering using PCA

mclust_model <- Mclust(pca_res$x[, 1:12],
G = 2:6)
table(Cluster =
mclust_model$classification,
GABRA3_Status = group_label)

cluster_annotation <- HeatmapAnnotation(
  GABRA3 = group_label,
  mclust =
as.factor(mclust_model$classification)
)

Heatmap(expr_top,
        name = "Expression",
        cluster_columns = hc_col,
        cluster_rows = hc_row,
        top_annotation =
cluster_annotation,
        show_column_names = FALSE,
        show_row_names = FALSE,
        column_title = "Clusters with
GABRA3 and Mclust Annotations")
```

```
#ROC
set.seed(123)
train_idx <-
caret::createDataPartition(group_label, p
= 0.7, list = FALSE)
x_train <- x_data[train_idx, ]
y_train <- group_label[train_idx]
x_test  <- x_data[-train_idx, ]
y_test  <- group_label[-train_idx]

fit_control <- trainControl(
  method = "cv",
  number = 10,
  classProbs = TRUE,
  summaryFunction = twoClassSummary,
  savePredictions = "final"
)

nb_model <- train(
  x = x_train,
  y = y_train,
  method = "naive_bayes",
  metric = "ROC",
  trControl = fit_control,
  tuneGrid = expand.grid(
    usekernel = FALSE,
    laplace = 1,
    adjust = 1
  )
)

pred <- predict(nb_model, x_test)
conf_mat <- confusionMatrix(pred, y_test,
positive = "High")
print(conf_mat)

# ROC and AUC (from saved predictions)
roc_obj <- roc(nb_model$pred$obs,
nb_model$pred$High, levels =
rev(levels(nb_model$pred$obs)))

plot(roc_obj, col = "blue", main = "ROC
Curve: Naive Bayes (GABRA3 High vs Low)")
auc_value <- auc(roc_obj)
print(paste("AUC:", round(auc_value, 3)))
```