GEO Expression Analysis - Boxplots

```r
# ---- Load packages ----
# Note: If org.Hs.eg.db is not installed, run these commands first:
# if (!require("BiocManager", quietly = TRUE)) install.packages("BiocManager")
# BiocManager::install("org.Hs.eg.db")

library(dplyr)          # Data manipulation
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)          # Data reshaping
library(tibble)         # Modern data frames
library(ggplot2)        # Plotting
library(ggpubr)         # Publication-ready plots with statistics
library(GEOquery)       # Download GEO datasets and annotations
```

```
## Loading required package: Biobase
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, saveRDS, setdiff,
##     table, tapply, union, unique, unsplit, which.max, which.min
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
```

```
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

## Setting options('download.file.method.GEOquery'='auto')

## Setting options('GEOquery.inmemory.gpl'=FALSE)
```

```
library(org.Hs.eg.db)   # Human gene annotations (Entrez ID to Symbol mapping)
```

```
## Loading required package: AnnotationDbi

## Loading required package: stats4

## Loading required package: IRanges

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:tidyr':
##
##     expand

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following object is masked from 'package:utils':
##
##     findMatches

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

##
## Attaching package: 'IRanges'

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

##
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:dplyr':
##
##     select

##
```

```
library(AnnotationDbi)  # Annotation database interface
```

```
# ---- Read expression data ----
# These are Series Matrix files downloaded from GEO
# comment.char = "!" skips metadata lines at the top of the file
GSE76808 <- read.table("GSE76808_series_matrix.txt", header = TRUE, row.names = 1,
                       sep = "\t", comment.char = "!", check.names = FALSE)
GSE48149 <- read.table("GSE48149_series_matrix.txt", header = TRUE, row.names = 1,
                       sep = "\t", comment.char = "!", check.names = FALSE)
GSE81292 <- read.table("GSE81292_series_matrix.txt", header = TRUE, row.names = 1,
                       sep = "\t", comment.char = "!", check.names = FALSE)
```

```r
# Read metadata (sample sheet with Sample ID and Condition columns)
metadata <- read.csv('samplesheet.csv', header = TRUE)
```

```r
# ---- Download platform annotations ----
# Each GEO dataset uses a specific microarray platform (GPL)
# We need the platform annotations to map probe IDs to gene symbols

gse1 <- getGEO("GSE76808", GSEMatrix = FALSE)
```

```
## Reading file....
## Parsing....
## Found 19 entities...
## GPL571 (1 of 20 entities)
## GSM2038267 (2 of 20 entities)
## GSM2038268 (3 of 20 entities)
## GSM2038269 (4 of 20 entities)
## GSM2038270 (5 of 20 entities)
## GSM2038271 (6 of 20 entities)
## GSM2038272 (7 of 20 entities)
## GSM2038273 (8 of 20 entities)
## GSM2038274 (9 of 20 entities)
## GSM2038275 (10 of 20 entities)
## GSM2038276 (11 of 20 entities)
## GSM2038277 (12 of 20 entities)
## GSM2038278 (13 of 20 entities)
## GSM2038279 (14 of 20 entities)
## GSM2038280 (15 of 20 entities)
## GSM2038281 (16 of 20 entities)
## GSM2038282 (17 of 20 entities)
## GSM2038283 (18 of 20 entities)
## GSM2038284 (19 of 20 entities)
```

```r
gse2 <- getGEO("GSE48149", GSEMatrix = FALSE)
```

```
## Reading file....
## Parsing....
## Found 54 entities...
## GPL16221 (1 of 55 entities)
## GSM1169960 (2 of 55 entities)
## GSM1169961 (3 of 55 entities)
## GSM1169962 (4 of 55 entities)
```

```
## GSM1169963 (5 of 55 entities)
## GSM1169964 (6 of 55 entities)
## GSM1169965 (7 of 55 entities)
## GSM1169966 (8 of 55 entities)
## GSM1169967 (9 of 55 entities)
## GSM1169968 (10 of 55 entities)
## GSM1169969 (11 of 55 entities)
## GSM1169970 (12 of 55 entities)
## GSM1169971 (13 of 55 entities)
## GSM1169972 (14 of 55 entities)
## GSM1169973 (15 of 55 entities)
## GSM1169974 (16 of 55 entities)
## GSM1169975 (17 of 55 entities)
## GSM1169976 (18 of 55 entities)
## GSM1169977 (19 of 55 entities)
## GSM1169978 (20 of 55 entities)
## GSM1169979 (21 of 55 entities)
## GSM1169980 (22 of 55 entities)
## GSM1169981 (23 of 55 entities)
## GSM1169982 (24 of 55 entities)
## GSM1169983 (25 of 55 entities)
## GSM1169984 (26 of 55 entities)
## GSM1169985 (27 of 55 entities)
## GSM1169986 (28 of 55 entities)
## GSM1169987 (29 of 55 entities)
## GSM1169988 (30 of 55 entities)
## GSM1169989 (31 of 55 entities)
## GSM1169990 (32 of 55 entities)
## GSM1169991 (33 of 55 entities)
## GSM1169992 (34 of 55 entities)
## GSM1169993 (35 of 55 entities)
## GSM1169994 (36 of 55 entities)
## GSM1169995 (37 of 55 entities)
## GSM1169996 (38 of 55 entities)
## GSM1169997 (39 of 55 entities)
## GSM1169998 (40 of 55 entities)
```

```
## GSM1169999 (41 of 55 entities)
## GSM1170000 (42 of 55 entities)
## GSM1170001 (43 of 55 entities)
## GSM1170002 (44 of 55 entities)
## GSM1170003 (45 of 55 entities)
## GSM1170004 (46 of 55 entities)
## GSM1170005 (47 of 55 entities)
## GSM1170006 (48 of 55 entities)
## GSM1170007 (49 of 55 entities)
## GSM1170008 (50 of 55 entities)
## GSM1170009 (51 of 55 entities)
## GSM1170010 (52 of 55 entities)
## GSM1170011 (53 of 55 entities)
## GSM1170012 (54 of 55 entities)
```

```
gse3 <- getGEO("GSE81292", GSEMatrix = FALSE)
```

```
## Reading file....
## Parsing....
## Found 21 entities...
## GPL18991 (1 of 22 entities)
## GSM2149850 (2 of 22 entities)
## GSM2149851 (3 of 22 entities)
## GSM2149852 (4 of 22 entities)
## GSM2149853 (5 of 22 entities)
## GSM2149854 (6 of 22 entities)
## GSM2149855 (7 of 22 entities)
## GSM2149856 (8 of 22 entities)
## GSM2149857 (9 of 22 entities)
## GSM2149858 (10 of 22 entities)
## GSM2149859 (11 of 22 entities)
## GSM2149860 (12 of 22 entities)
## GSM2149861 (13 of 22 entities)
## GSM2149862 (14 of 22 entities)
## GSM2149863 (15 of 22 entities)
## GSM2149864 (16 of 22 entities)
## GSM2149865 (17 of 22 entities)
## GSM2149866 (18 of 22 entities)
```

```
## GSM2149867 (19 of 22 entities)
## GSM2149868 (20 of 22 entities)
## GSM2149869 (21 of 22 entities)
```

```
# Extract platform IDs
GPL1 <- gse1@gsms[[1]]@header$platform_id
GPL2 <- gse2@gsms[[1]]@header$platform_id
GPL3 <- gse3@gsms[[1]]@header$platform_id

# Download platform annotation tables
gpl1 <- getGEO(GPL1, AnnotGPL = TRUE)
gpl2 <- getGEO(GPL2, AnnotGPL = TRUE)
```

```
## Annotation GPL not available, so will use submitter GPL instead
```

```
gpl3 <- getGEO(GPL3, AnnotGPL = TRUE)
```

```
## Annotation GPL not available, so will use submitter GPL instead
```

```
annot1 <- Table(gpl1)
annot2 <- Table(gpl2)
annot3_raw <- Table(gpl3)

# Note: GPL18991 (used by GSE81292) doesn't have gene symbols directly
# It has Entrez Gene IDs in the "ORF" column, which we convert to symbols
```

```
# ---- Define genes of interest ----
genes_of_interest <- c(
  "MMP7", "KRT17", "SPP1", "GDF15",
  "CDKN2A", "FRZB", "PDE1A", "NAP1L2"
)

# Note: Not all platforms measure all genes
# KRT17 is missing from GPL571 (GSE76808)
```

```
# ---- Process GSE76808 ----
# Platform: GPL571 (Affymetrix Human Genome U133A 2.0 Array)

# Map probes to gene symbols
GSE76808_mapped <- GSE76808 %>%
  rownames_to_column("ProbeID")

# Create annotation subset
# Using bracket notation to avoid backtick issues with "Gene symbol" column name
annot1_subset <- data.frame(
  ID = annot1$ID,
  Gene = annot1[["Gene symbol"]],
  stringsAsFactors = FALSE
)

GSE76808_mapped <- GSE76808_mapped %>%
  left_join(annot1_subset, by = c("ProbeID" = "ID")) %>%
  filter(!is.na(Gene) & Gene != "")

# Filter for genes of interest and collapse duplicate probes
```

```r
# Multiple probes can map to the same gene; we take the mean expression
GSE76808_collapsed <- GSE76808_mapped %>%
  filter(Gene %in% genes_of_interest) %>%
  dplyr::select(-ProbeID) %>%
  group_by(Gene) %>%
  summarize(across(where(is.numeric), mean), .groups = "drop")

# Convert to long format for ggplot
expr_long_76808 <- GSE76808_collapsed %>%
  pivot_longer(
    cols = -Gene,
    names_to = "Sample",
    values_to = "Expression"
  ) %>%
  left_join(metadata, by = "Sample")
```

```r
# ---- Process GSE48149 ----
# Platform: GPL16221 (Illumina HumanHT-12 WG-DASL V4.0)

GSE48149_mapped <- GSE48149 %>%
  rownames_to_column("ProbeID")

# Create annotation subset
annot2_subset <- data.frame(
  ID = annot2$ID,
  Gene = annot2$Symbol,
  stringsAsFactors = FALSE
)

GSE48149_mapped <- GSE48149_mapped %>%
  left_join(annot2_subset, by = c("ProbeID" = "ID")) %>%
  filter(!is.na(Gene) & Gene != "")

# Filter for genes of interest and collapse duplicates
GSE48149_collapsed <- GSE48149_mapped %>%
  filter(Gene %in% genes_of_interest) %>%
  dplyr::select(-ProbeID) %>%
  group_by(Gene) %>%
  summarize(across(where(is.numeric), mean), .groups = "drop")

# Convert to long format and remove samples with missing condition info
expr_long_48149 <- GSE48149_collapsed %>%
  pivot_longer(
    cols = -Gene,
    names_to = "Sample",
    values_to = "Expression"
  ) %>%
  left_join(metadata, by = "Sample") %>%
  filter(!is.na(Condition))
```

```r
# ---- Process GSE81292 ----
# Platform: GPL18991 (Affymetrix Human Gene 1.1 ST Array)
# This platform uses Entrez Gene IDs instead of gene symbols
```

```r
# Map Entrez IDs to gene symbols using org.Hs.eg.db
gene_symbols <- mapIds(org.Hs.eg.db,
                       keys = as.character(annot3_raw$ORF),
                       column = "SYMBOL",
                       keytype = "ENTREZID",
                       multiVals = "first")
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```r
# Create annotation subset
# Note: Some Entrez IDs may not map to symbols (returns NA)
annot3_subset <- data.frame(
  ID = annot3_raw$ID,
  Gene = as.character(gene_symbols[as.character(annot3_raw$ORF)]),
  stringsAsFactors = FALSE
) %>%
  filter(!is.na(Gene) & Gene != "")

GSE81292_mapped <- GSE81292 %>%
  rownames_to_column("ProbeID") %>%
  left_join(annot3_subset, by = c("ProbeID" = "ID")) %>%
  filter(!is.na(Gene) & Gene != "")

# Filter for genes of interest and collapse duplicates
GSE81292_collapsed <- GSE81292_mapped %>%
  filter(Gene %in% genes_of_interest) %>%
  dplyr::select(-ProbeID) %>%
  group_by(Gene) %>%
  summarize(across(where(is.numeric), mean), .groups = "drop")

# Convert to long format and remove samples with missing condition info
expr_long_81292 <- GSE81292_collapsed %>%
  pivot_longer(
    cols = -Gene,
    names_to = "Sample",
    values_to = "Expression"
  ) %>%
  left_join(metadata, by = "Sample") %>%
  filter(!is.na(Condition))
```

```r
# ---- Verify the output ----
cat("GSE76808 - Rows:", nrow(expr_long_76808), "Genes:", n_distinct(expr_long_76808$Gene), "\n")
```

```
## GSE76808 - Rows: 126 Genes: 7
```

```r
cat("GSE48149 - Rows:", nrow(expr_long_48149), "Genes:", n_distinct(expr_long_48149$Gene), "\n")
```

```
## GSE48149 - Rows: 256 Genes: 8
```

```r
cat("GSE81292 - Rows:", nrow(expr_long_81292), "Genes:", n_distinct(expr_long_81292$Gene), "\n")
```

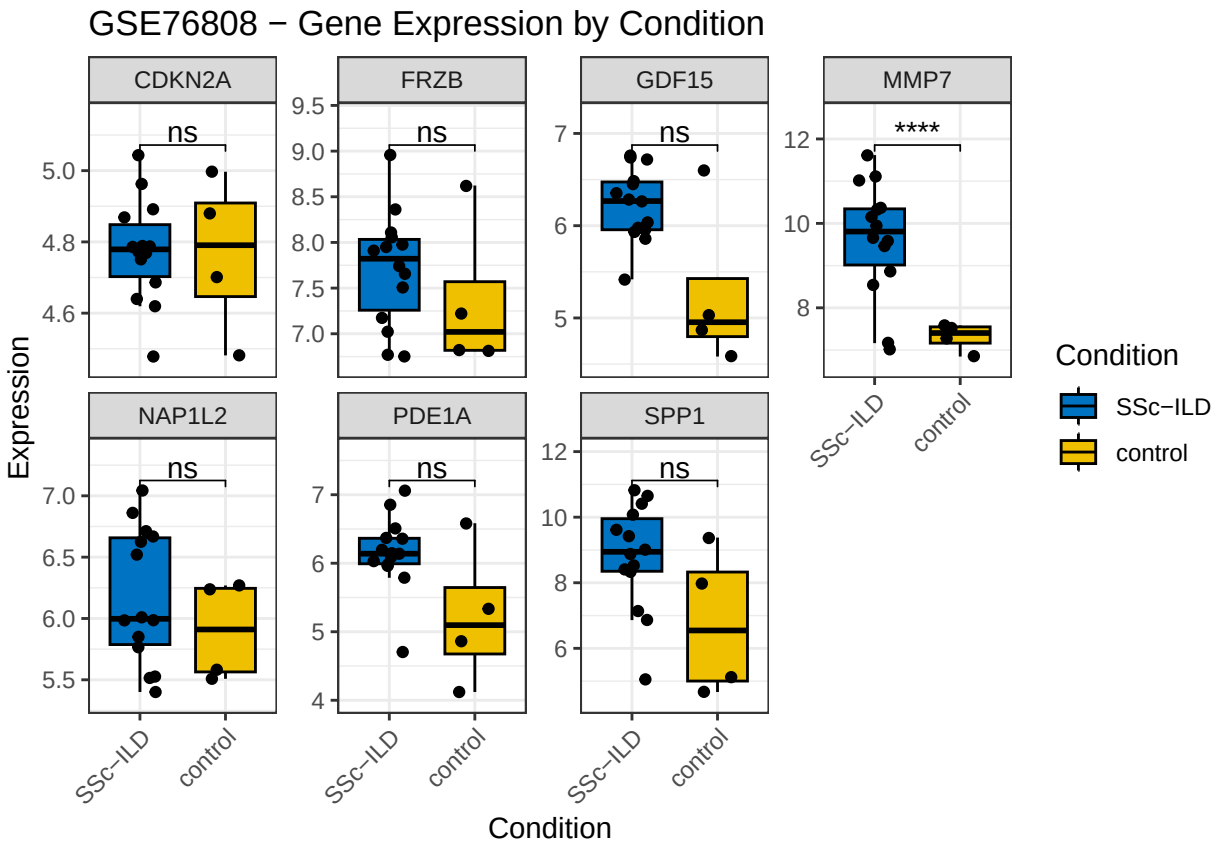
```
## GSE81292 - Rows: 160 Genes: 8
```

```r
# List which genes are present in each dataset
cat("\nGenes in GSE76808:", paste(sort(unique(expr_long_76808$Gene)), collapse=", "), "\n")
```

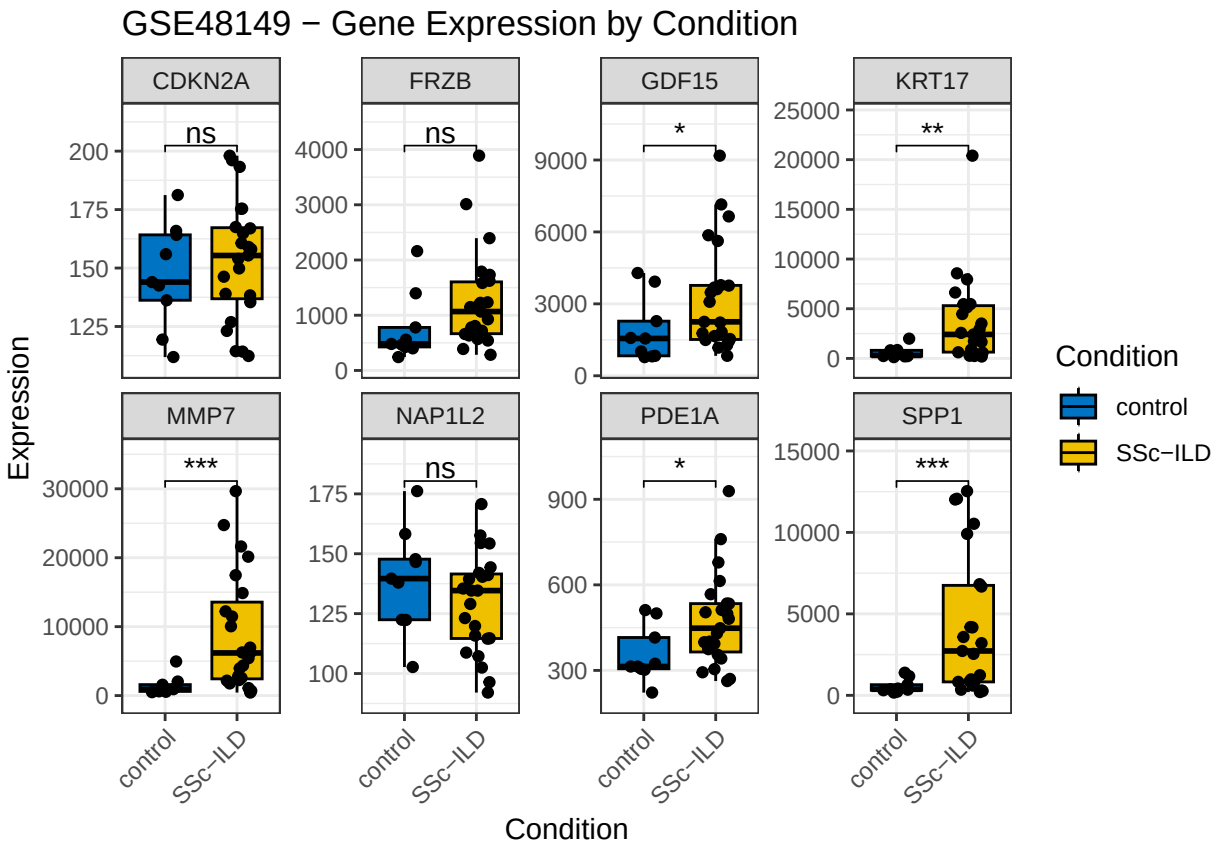
```
##
```

```
## Genes in GSE76808: CDKN2A, FRZB, GDF15, MMP7, NAP1L2, PDE1A, SPP1
cat("Genes in GSE48149:", paste(sort(unique(expr_long_48149$Gene)), collapse=", "), "\n")

## Genes in GSE48149: CDKN2A, FRZB, GDF15, KRT17, MMP7, NAP1L2, PDE1A, SPP1
cat("Genes in GSE81292:", paste(sort(unique(expr_long_81292$Gene)), collapse=", "), "\n")

## Genes in GSE81292: CDKN2A, FRZB, GDF15, KRT17, MMP7, NAP1L2, PDE1A, SPP1
# Identify missing genes
cat("\nMissing from GSE76808:", paste(setdiff(genes_of_interest, unique(expr_long_76808$Gene)), collaps

##
## Missing from GSE76808: KRT17
cat("Missing from GSE48149:", paste(setdiff(genes_of_interest, unique(expr_long_48149$Gene)), collapse=

## Missing from GSE48149:
cat("Missing from GSE81292:", paste(setdiff(genes_of_interest, unique(expr_long_81292$Gene)), collapse=

## Missing from GSE81292:
# ---- Create boxplots with significance testing ----
# Define pairwise comparison for statistical testing
comparisons <- list(c("control", "SSc-ILD"))

# GSE76808 boxplot
# facet.by creates separate panels for each gene
# scales = "free_y" allows each gene to have its own y-axis scale
# This is important because genes have vastly different expression levels
p_76808 <- ggboxplot(expr_long_76808, x = "Condition", y = "Expression", fill = "Condition",
          palette = "jco", add = "jitter",
          facet.by = "Gene", scales = "free_y", nrow = 2) +
  stat_compare_means(comparisons = comparisons, method = "t.test", label = "p.signif") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(expand = expansion(mult = c(0.1, 0.2))) +  # Add space for significance bars
  labs(title = "GSE76808 - Gene Expression by Condition")

# GSE48149 boxplot
p_48149 <- ggboxplot(expr_long_48149, x = "Condition", y = "Expression", fill = "Condition",
          palette = "jco", add = "jitter",
          facet.by = "Gene", scales = "free_y", nrow = 2) +
  stat_compare_means(comparisons = comparisons, method = "t.test", label = "p.signif") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(expand = expansion(mult = c(0.1, 0.2))) +
  labs(title = "GSE48149 - Gene Expression by Condition")

# GSE81292 boxplot
p_81292 <- ggboxplot(expr_long_81292, x = "Condition", y = "Expression", fill = "Condition",
          palette = "jco", add = "jitter",
          facet.by = "Gene", scales = "free_y", nrow = 2) +
  stat_compare_means(comparisons = comparisons, method = "t.test", label = "p.signif") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
```

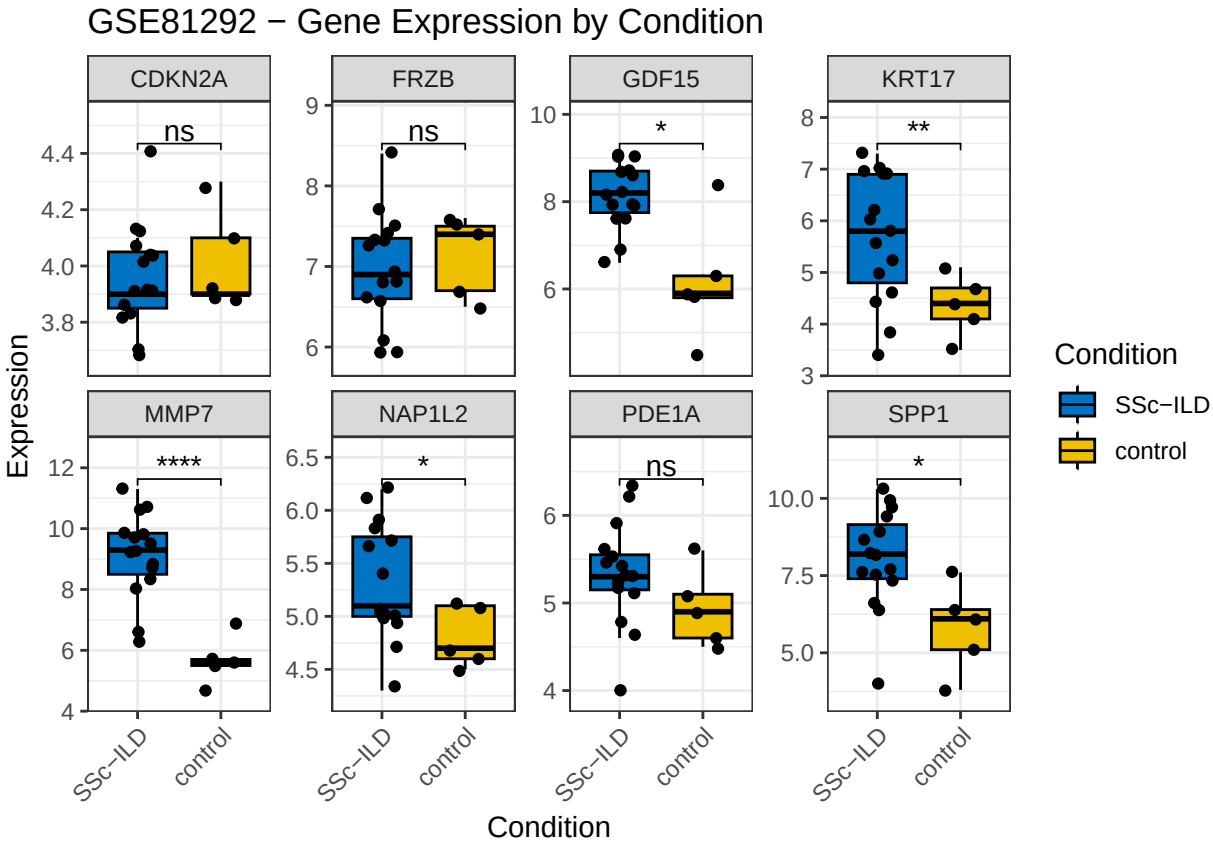
```
  scale_y_continuous(expand = expansion(mult = c(0.1, 0.2))) +
  labs(title = "GSE81292 - Gene Expression by Condition")

# Save plots as high-resolution PNG files
ggsave("GSE76808_genes.png", plot = p_76808, width = 9, height = 6, dpi = 300)
ggsave("GSE48149_genes.png", plot = p_48149, width = 9, height = 6, dpi = 300)
ggsave("GSE81292_genes.png", plot = p_81292, width = 9, height = 6, dpi = 300)

# Display plots
p_76808
```




```
p_48149
```


GSE48149 − Gene Expression by Condition
CDKN2A
FRZB
GDF15
KRT17
MMP7
NAP1L2
PDE1A
SPP1
ns
*
**
***
Expression
Condition
control
SSc−ILD

p_81292


GSE81292 − Gene Expression by Condition
CDKN2A
FRZB
GDF15
KRT17
MMP7
NAP1L2
PDE1A
SPP1
ns
ns
*
**
****
*
ns
*
Expression
Condition
SSc−ILD
control
Condition

Notes on Common Issues:

1. Missing genes: Different microarray platforms measure different genes.

Not all genes of interest may be present in all datasets.

2. Different y-axis scales: This is expected! Genes have vastly different

expression levels. Using free_y scales lets you see relative differences

for each gene.

3. Namespace conflicts: AnnotationDbi and dplyr both have a select() function.

Use dplyr::select() to explicitly specify which one you want.

4. Column name issues: Some annotation tables have spaces in column names

(e.g., "Gene symbol"). Use bracket notation annot[["Gene symbol"]] or

rename during the join to avoid backtick issues.

5. Platform-specific mapping: Some platforms (like GPL18991) require

additional steps to convert probe IDs to gene symbols via Entrez IDs.