Enhanced Dopamine MSNs vs Others Epigenetic Analysis

Deepak Poduval

2025-08-25

## Load and Initial Data Exploration

# Load the RDS file  
so <- readRDS("raw\_data/GSE167920\_Results\_full\_nuclei\_processed\_final.rds")  
  
# Explore data structure  
cat("Data Overview\n")

## Data Overview

cat(paste("Number of cells:", ncol(so)), "\n")

## Number of cells: 63884

cat(paste("Number of genes:", nrow(so)), "\n")

## Number of genes: 2000

cat(paste("Available assays:", paste(names(so@assays), collapse = ", ")), "\n")

## Available assays: RNA, integrated

# Check metadata  
cat("Metadata Columns\n")

## Metadata Columns

print(colnames(so@meta.data))

## [1] "cell.id" "nFeatures\_RNA" "nCount.RNA" "percent.ribo"   
## [5] "seurat\_clusters" "region\_name" "cell\_type" "cell\_type\_2"   
## [9] "monkey"

# Check cell types  
if ("cell\_type" %in% colnames(so@meta.data)) {  
 cat("Cell Type Distribution\n")  
 print(table(so@meta.data$cell\_type))  
}

## Cell Type Distribution  
##   
## Oligos Astrocytes DRD2 DRD1   
## 14314 17424 7310 9148   
## Interneurons Oligos\_Pre Microglia Endothelial   
## 7629 3079 2430 1723   
## Mural/Fibroblast Unknown1 Unknown2 Unknown3   
## 827 0 0 0   
## Unknown4   
## 0

# Check for D1/D2 MSN populations  
cat("Checking For D1/D2 MSNs\n")

## Checking For D1/D2 MSNs

msn\_types <- grep("DRD|D1|D2|MSN", unique(so@meta.data$cell\_type), value = TRUE, ignore.case = TRUE)  
cat(paste("MSN-related cell types found:", paste(msn\_types, collapse = ", ")), "\n")

## MSN-related cell types found: DRD1, DRD2

# Basic metadata structure  
str(so@meta.data, max.level = 1)

## 'data.frame': 63884 obs. of 9 variables:  
## $ cell.id : int 61609 61619 61620 61636 61661 61682 61718 61719 61720 61741 ...  
## $ nFeatures\_RNA : int 5061 3604 3940 5650 5248 4431 5151 3452 3531 4684 ...  
## $ nCount.RNA : num 15884 8927 9198 20929 17577 ...  
## $ percent.ribo : num 0.00327 0.00493 0.00402 0.00253 0.00256 ...  
## $ seurat\_clusters: Factor w/ 44 levels "0","1","2","3",..: 25 25 17 10 17 25 10 33 33 17 ...  
## $ region\_name : Factor w/ 3 levels "caudate","nacc",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ cell\_type : Factor w/ 13 levels "Oligos","Astrocytes",..: 4 4 4 4 4 4 4 4 4 4 ...  
## $ cell\_type\_2 : Factor w/ 8 levels "Oligos","Astrocytes",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ monkey : Factor w/ 2 levels "Monkey\_F","Monkey\_P": 1 1 1 1 1 1 1 1 1 1 ...

## Quality Control Assessment

# Clear any problematic graph objects first  
so@graphs <- list()  
  
# Calculate additional QC metrics if not present  
if (!"percent.mt" %in% colnames(so@meta.data)) {  
 so[["percent.mt"]] <- PercentageFeatureSet(so, pattern = "^MT-")  
}  
if (!"percent.ribo" %in% colnames(so@meta.data)) {  
 so[["percent.ribo"]] <- PercentageFeatureSet(so, pattern = "^RP[SL]")  
}  
  
# QC summaries  
summary(so@meta.data$nFeatures\_RNA)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1001 2074 2770 3209 4361 11404

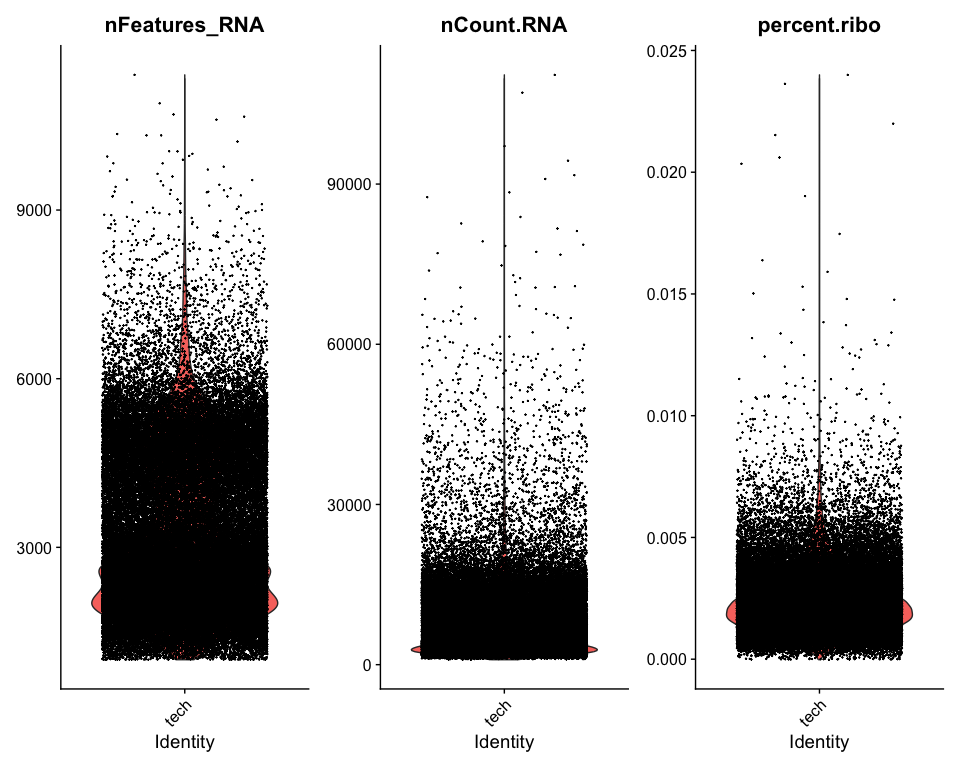
summary(so@meta.data$nCount.RNA)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 926.6 2932.8 4681.0 7260.8 10068.0 110443.0

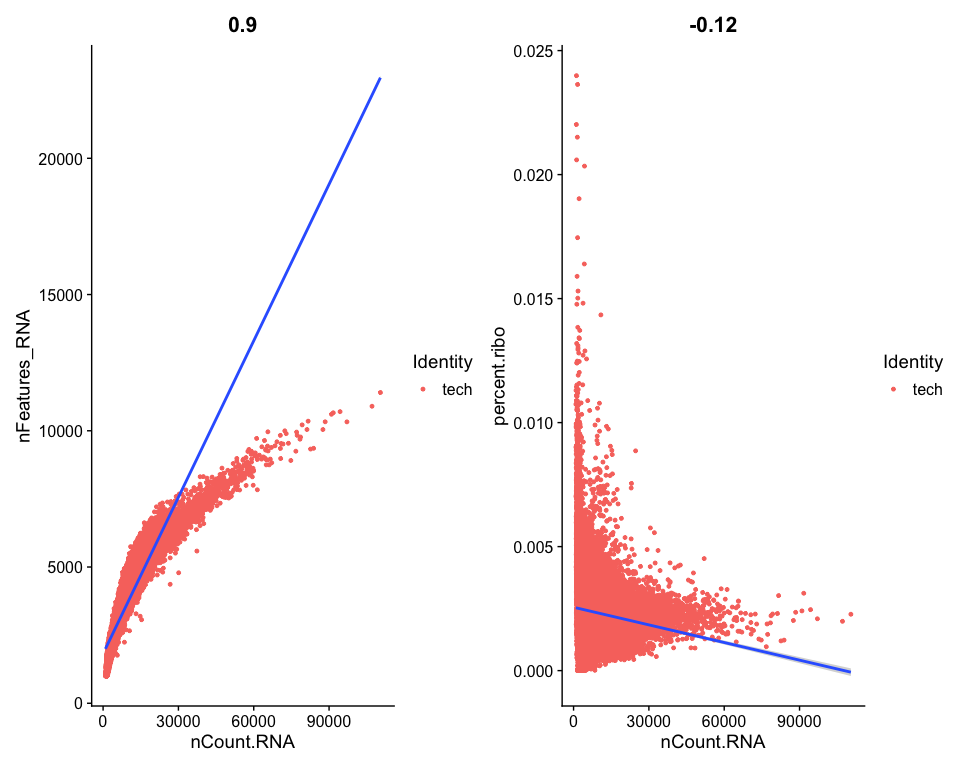
summary(so@meta.data$percent.ribo)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000000 0.001516 0.002180 0.002383 0.002970 0.023988

# QC plots  
p1 <- VlnPlot(so, features = c("nFeatures\_RNA", "nCount.RNA", "percent.ribo"), ncol = 3, pt.size = 0.1)  
print(p1)



ggsave("MSNvsO\_figures/QC\_violin\_plots.png", p1, width = 15, height = 6)  
  
p2 <- FeatureScatter(so, feature1 = "nCount.RNA", feature2 = "nFeatures\_RNA") + geom\_smooth(method = "lm")  
p3 <- FeatureScatter(so, feature1 = "nCount.RNA", feature2 = "percent.ribo") + geom\_smooth(method = "lm")  
p4 <- p2 + p3  
print(p4)



ggsave("MSNvsO\_figures/QC\_scatter\_plots.png", p4, width = 12, height = 6)

## Data Filtering and Quality Control

cat("Before Filtering\n")

## Before Filtering

cat(paste("Cells:", ncol(so)), "\n")

## Cells: 63884

cat(paste("Genes:", nrow(so)), "\n")

## Genes: 2000

if ("cell\_type" %in% colnames(so@meta.data)) {  
 cat("Cell Type Distribution Before Filtering\n")  
 print(table(so@meta.data$cell\_type))  
}

## Cell Type Distribution Before Filtering  
##   
## Oligos Astrocytes DRD2 DRD1   
## 14314 17424 7310 9148   
## Interneurons Oligos\_Pre Microglia Endothelial   
## 7629 3079 2430 1723   
## Mural/Fibroblast Unknown1 Unknown2 Unknown3   
## 827 0 0 0   
## Unknown4   
## 0

# Apply filtering based on QC metrics  
so <- subset(so, subset = nFeatures\_RNA > 200 &  
 nFeatures\_RNA < 8000 &  
 nCount.RNA > 1000 &  
 nCount.RNA < 80000 &  
 percent.ribo < 0.02)  
  
cat("After Filtering\n")

## After Filtering

cat(paste("Cells:", ncol(so)), "\n")

## Cells: 63683

cat(paste("Genes:", nrow(so)), "\n")

## Genes: 2000

if ("cell\_type" %in% colnames(so@meta.data)) {  
 cat("Cell Type Distribution After Filtering\n")  
 print(table(so@meta.data$cell\_type))  
}

## Cell Type Distribution After Filtering  
##   
## Oligos Astrocytes DRD2 DRD1   
## 14311 17422 7278 9117   
## Interneurons Oligos\_Pre Microglia Endothelial   
## 7504 3078 2430 1716   
## Mural/Fibroblast   
## 827

## Data Normalization and Scaling

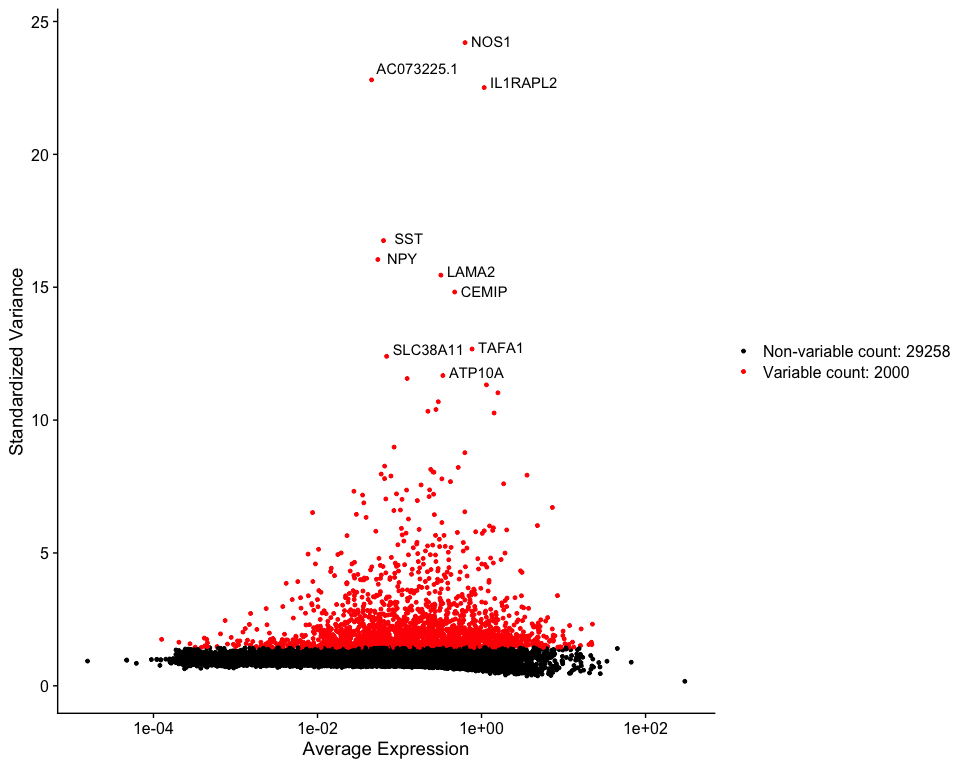
DefaultAssay(so) <- "RNA"  
  
# Normalize if needed  
if (is.null(so[["RNA"]]@data) || ncol(so[["RNA"]]@data) == 0) {  
 cat("Normalizing RNA data...\n")  
 so <- NormalizeData(so)  
} else {  
 cat("RNA data already normalized\n")  
}

## RNA data already normalized

# Variable features  
if (length(VariableFeatures(so)) == 0) {  
 cat("Finding variable features...\n")  
 so <- FindVariableFeatures(so, selection.method = "vst", nfeatures = 2000)  
} else {  
 cat("Variable features already identified\n")  
}

## Finding variable features...

# Plot variable features  
top10 <- head(VariableFeatures(so), 10)  
p5 <- VariableFeaturePlot(so)  
p6 <- LabelPoints(plot = p5, points = top10, repel = TRUE)  
print(p6)

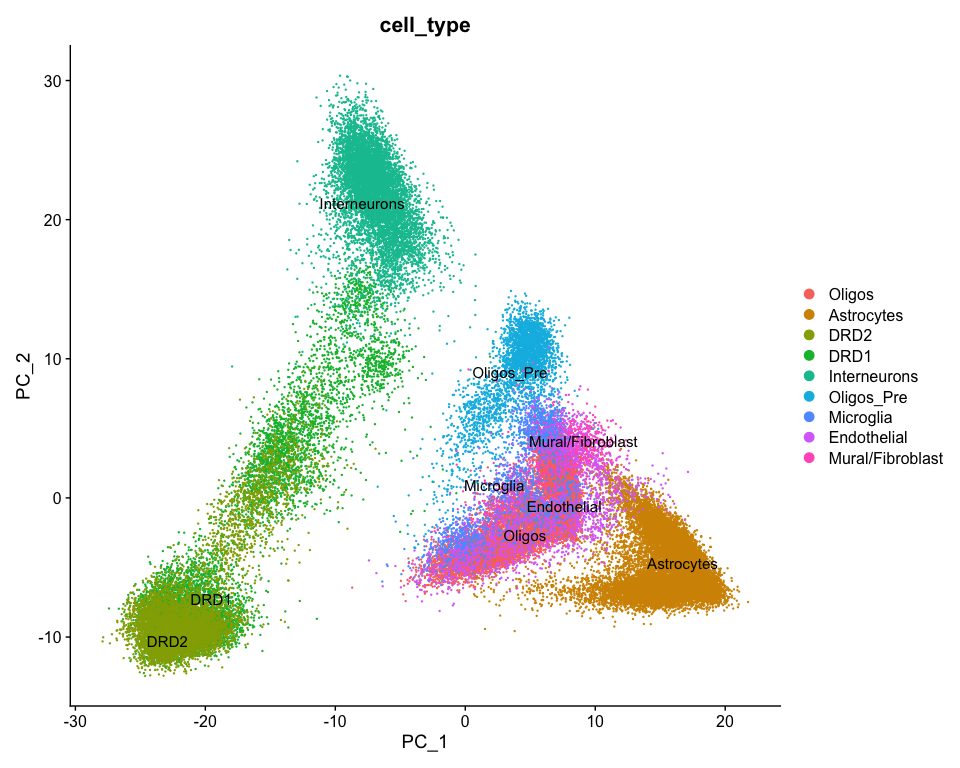


ggsave("MSNvsO\_figures/Variable\_Features.png", p6, width = 12, height = 8)  
  
# Scale only variable features for speed  
if (is.null(so[["RNA"]]@scale.data) || ncol(so[["RNA"]]@scale.data) == 0) {  
 cat("Scaling RNA variable features...\n")  
 so <- ScaleData(so, features = VariableFeatures(so))  
} else {  
 cat("RNA data already scaled\n")  
}

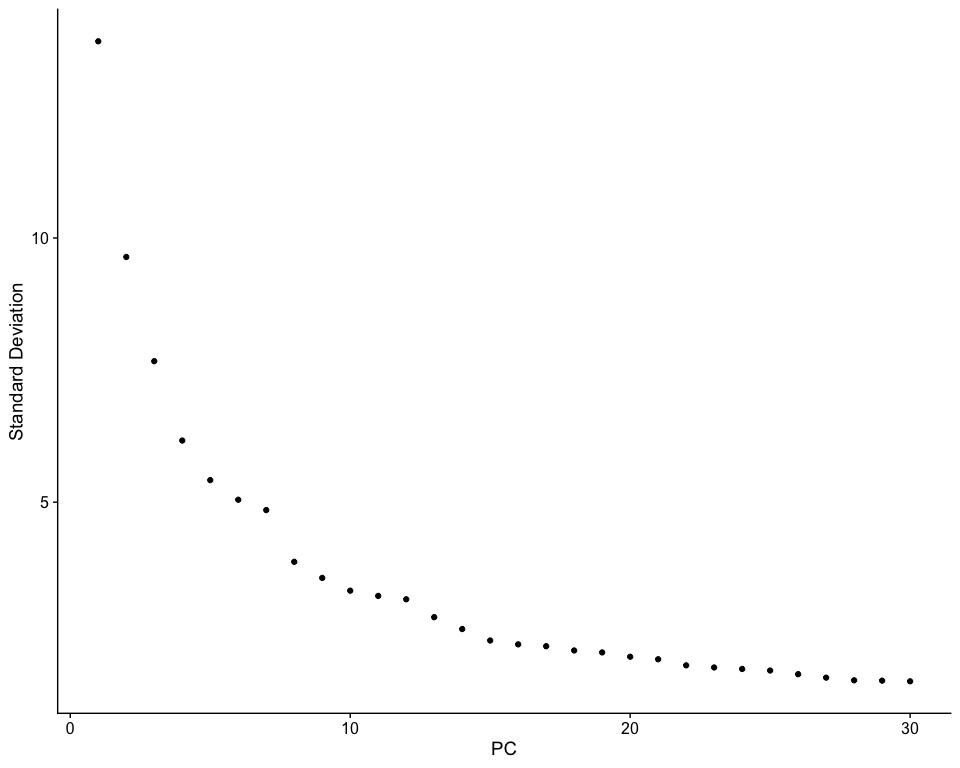
## Scaling RNA variable features...

## PCA and Dimensionality Reduction

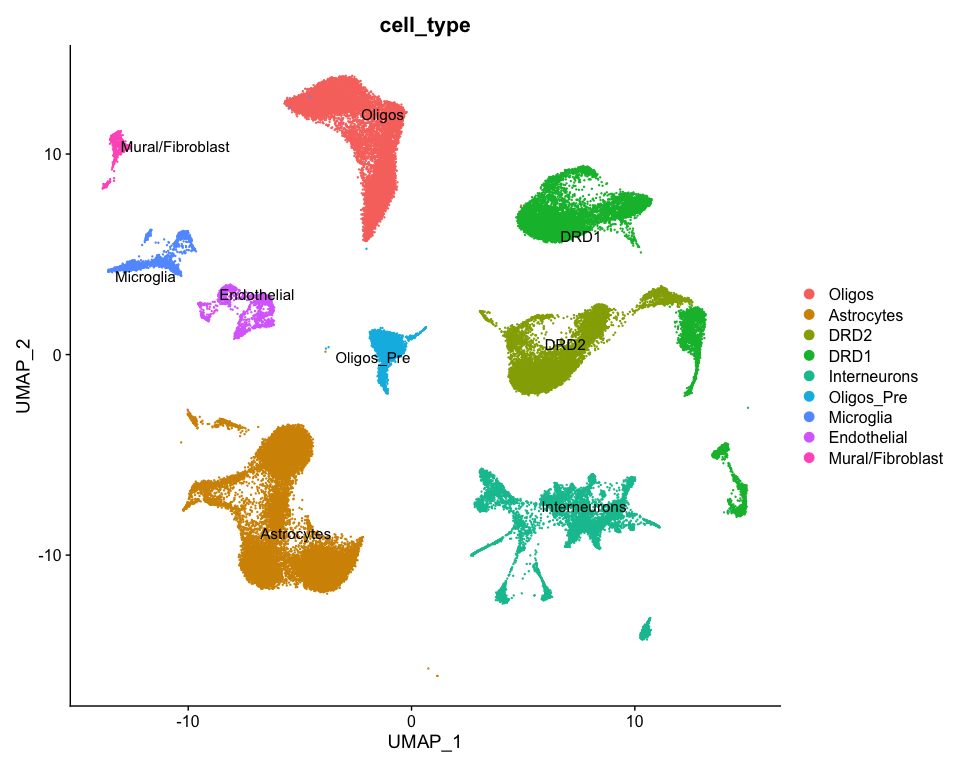
if (!"pca" %in% names(so@reductions)) {  
 cat("Running PCA...\n")  
 so <- RunPCA(so, features = VariableFeatures(so))  
}  
  
p7 <- DimPlot(so, reduction = "pca", group.by = "cell\_type", label = TRUE, repel = TRUE)  
print(p7)



ggsave("MSNvsO\_figures/PCA\_plot.png", p7, width = 12, height = 8)  
  
p8 <- ElbowPlot(so, ndims = 50)  
print(p8)



ggsave("MSNvsO\_figures/Elbow\_plot.png", p8, width = 10, height = 6)  
  
if (!"umap" %in% names(so@reductions)) {  
 cat("Running UMAP...\n")  
 set.seed(1234)  
 so <- RunUMAP(so, dims = 1:20)  
}  
  
p9 <- DimPlot(so, reduction = "umap", group.by = "cell\_type", label = TRUE, repel = TRUE)  
print(p9)



ggsave("MSNvsO\_figures/UMAP\_CellType.png", p9, width = 12, height = 10)

## Dopamine Receptor Expression Analysis and MSN Classification

# Key dopamine-related markers (filtered for reliably detected ones)  
dopamine\_markers <- c("DRD1", "DRD2", "PPP1R1B", "PDE1B", "BCL11B", "KIAA1211L", "PDE2A", "SLIT3", "NGEF")  
  
present\_markers <- dopamine\_markers[dopamine\_markers %in% rownames(so)]  
missing\_markers <- setdiff(dopamine\_markers, present\_markers)  
  
cat("Dopamine Marker Availability\n")

## Dopamine Marker Availability

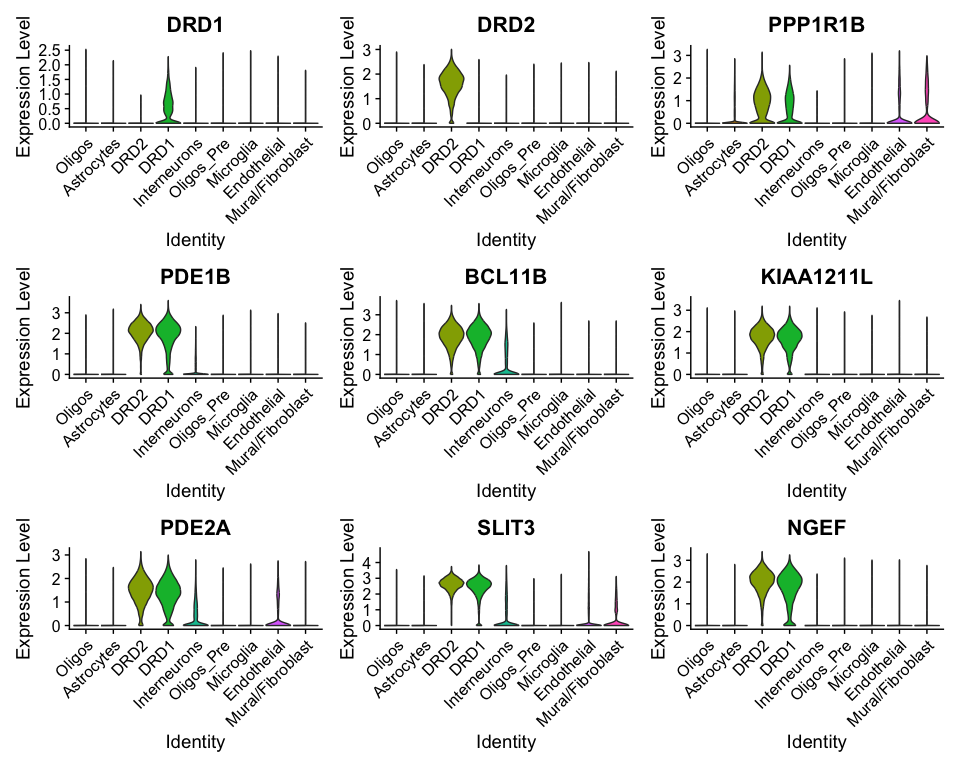
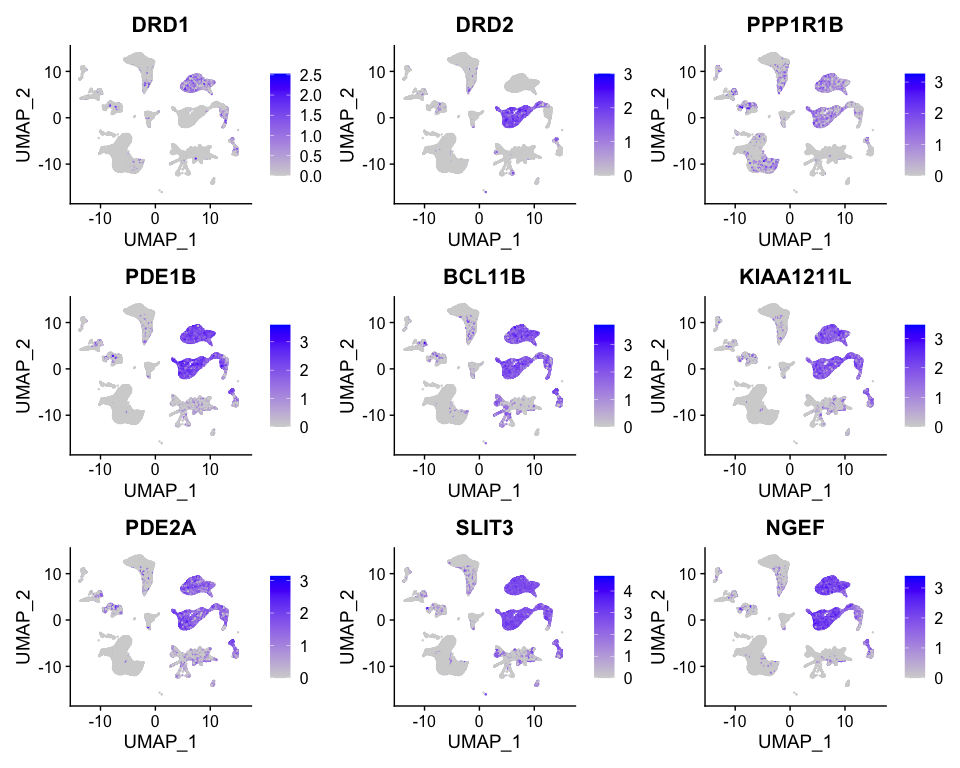
cat(paste("Present:", paste(present\_markers, collapse = ", ")), "\n")

## Present: DRD1, DRD2, PPP1R1B, PDE1B, BCL11B, KIAA1211L, PDE2A, SLIT3, NGEF

cat(paste("Missing:", paste(missing\_markers, collapse = ", ")), "\n")

## Missing:

if (length(present\_markers) > 0) {  
 p10 <- FeaturePlot(so, features = present\_markers, ncol = 3, reduction = "umap")  
 print(p10); ggsave("MSNvsO\_figures/Dopamine\_Markers\_FeaturePlot.png", p10, width = 15, height = 10)  
  
 p11 <- VlnPlot(so, features = present\_markers, group.by = "cell\_type", ncol = 3, pt.size = 0)  
 print(p11); ggsave("MSNvsO\_figures/Dopamine\_Markers\_ViolinPlot.png", p11, width = 15, height = 10)  
}



# D1/D2 classification and MSN vs Others grouping  
if (!"D1\_D2\_class" %in% colnames(so@meta.data) && all(c("DRD1","DRD2") %in% rownames(so))) {  
 drd1\_exp <- GetAssayData(so, assay = "RNA", layer = "data")["DRD1", ]  
 drd2\_exp <- GetAssayData(so, assay = "RNA", layer = "data")["DRD2", ]  
 so$D1\_D2\_class <- ifelse(drd1\_exp > 1 & drd2\_exp < 0.5, "D1R+",  
 ifelse(drd2\_exp > 1 & drd1\_exp < 0.5, "D2R+", "Mixed/Other"))  
 cat("D1R/D2R Classification\n"); print(table(so$D1\_D2\_class))  
}

## D1R/D2R Classification  
##   
## D1R+ D2R+ Mixed/Other   
## 2354 7378 53951

# Create MSN vs Others classification  
so$MSN\_vs\_Others <- "Others"  
  
# Method 1: Use cell\_type annotation if available  
if ("cell\_type" %in% colnames(so@meta.data)) {  
 msn\_cell\_types <- grep("DRD1|DRD2|MSN", unique(so@meta.data$cell\_type), value = TRUE, ignore.case = TRUE)  
 if (length(msn\_cell\_types) > 0) {  
 so$MSN\_vs\_Others[so$cell\_type %in% msn\_cell\_types] <- "MSNs"  
 cat("Using cell\_type annotation for MSN identification\n")  
 cat("MSN cell types:", paste(msn\_cell\_types, collapse = ", "), "\n")  
 }  
}

## Using cell\_type annotation for MSN identification  
## MSN cell types: DRD1, DRD2

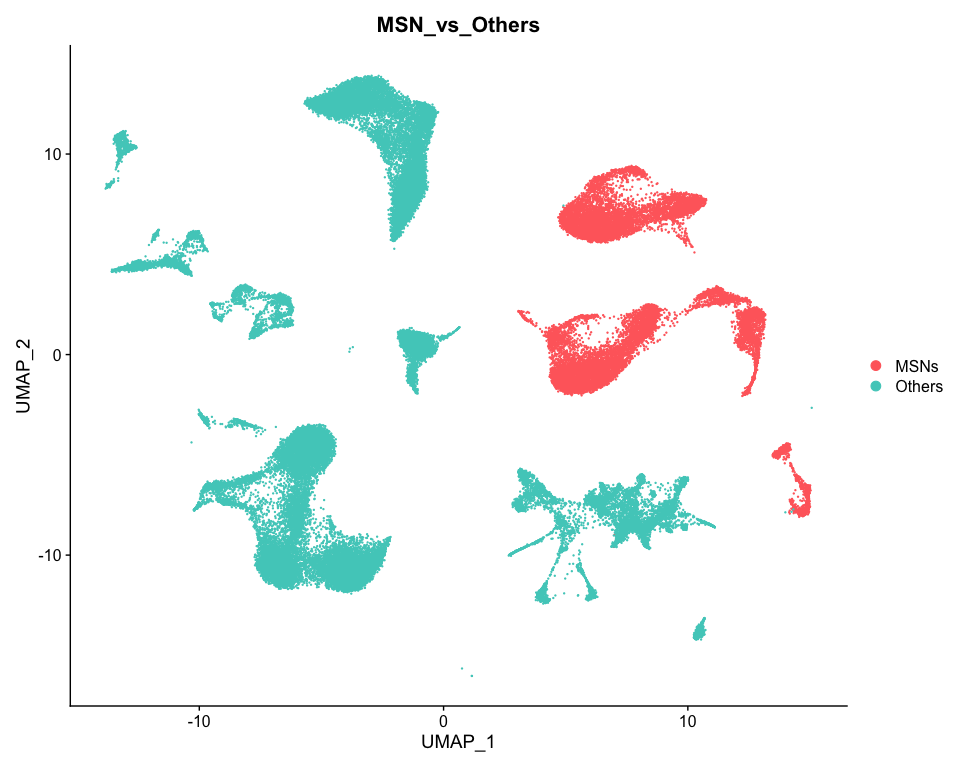
# Method 2: Use D1/D2 classification as backup  
if (sum(so$MSN\_vs\_Others == "MSNs") == 0 && "D1\_D2\_class" %in% colnames(so@meta.data)) {  
 so$MSN\_vs\_Others[so$D1\_D2\_class %in% c("D1R+", "D2R+")] <- "MSNs"  
 cat("Using D1R+/D2R+ classification for MSN identification\n")  
}  
  
cat("MSN vs Others Classification\n")

## MSN vs Others Classification

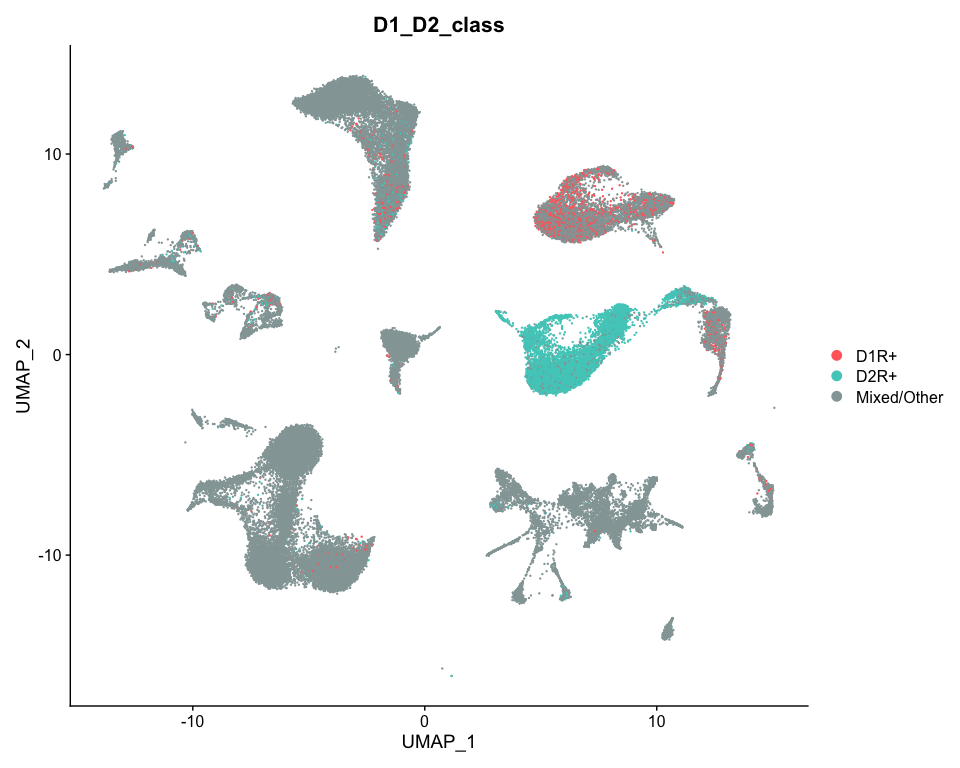
print(table(so$MSN\_vs\_Others))

##   
## MSNs Others   
## 16395 47288

# Plot MSN vs Others classification  
p12 <- DimPlot(so, group.by = "MSN\_vs\_Others", reduction = "umap",  
 cols = c("MSNs" = "#FF6B6B", "Others" = "#4ECDC4"))  
print(p12); ggsave("MSNvsO\_figures/MSN\_vs\_Others\_Classification.png", p12, width = 10, height = 8)



# Also plot D1/D2 within MSNs if available  
if ("D1\_D2\_class" %in% colnames(so@meta.data)) {  
 p12b <- DimPlot(so, group.by = "D1\_D2\_class", reduction = "umap",  
 cols = c("D1R+" = "#FF6B6B", "D2R+" = "#4ECDC4", "Mixed/Other" = "#95A5A6"))  
 print(p12b); ggsave("MSNvsO\_figures/D1R\_D2R\_Classification.png", p12b, width = 10, height = 8)  
}



## Filter Dataset for Analysis

cat("Dataset Filtering for MSN vs Others Analysis\n")

## Dataset Filtering for MSN vs Others Analysis

# Filter out cells with too few genes/counts for robust analysis  
so\_filtered <- so  
  
# Remove any cells that couldn't be classified  
so\_filtered <- subset(so\_filtered, subset = MSN\_vs\_Others %in% c("MSNs", "Others"))  
  
cat("Filtered Dataset Summary\n")

## Filtered Dataset Summary

cat(paste("Total cells:", ncol(so\_filtered)), "\n")

## Total cells: 63683

cat(paste("Total genes:", nrow(so\_filtered)), "\n")

## Total genes: 31258

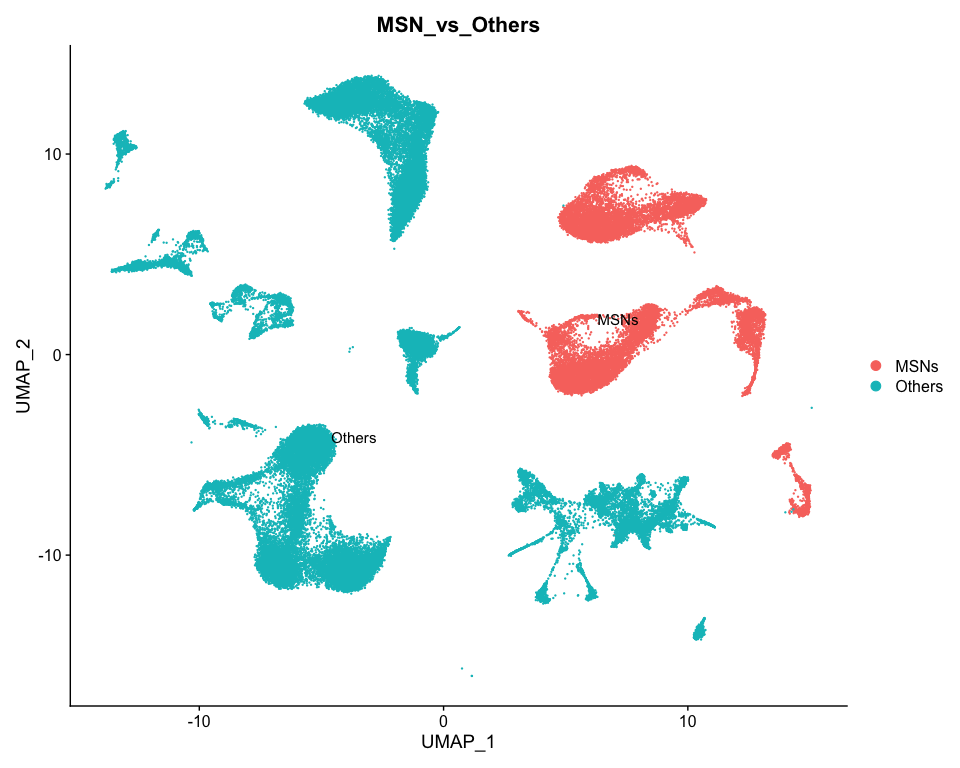
print(table(so\_filtered$MSN\_vs\_Others))

##   
## MSNs Others   
## 16395 47288

DefaultAssay(so\_filtered) <- "RNA"  
  
# Filter low-expression genes  
non\_zero\_genes <- rownames(so\_filtered)[Matrix::rowSums(GetAssayData(so\_filtered, layer = "counts")) > 0]  
so\_filtered <- subset(so\_filtered, features = non\_zero\_genes)  
cat(paste("Genes after filtering:", nrow(so\_filtered)), "\n")

## Genes after filtering: 31258

if (ncol(so\_filtered) > 100) {  
 p13 <- DimPlot(so\_filtered, reduction = "umap", group.by = "MSN\_vs\_Others", label = TRUE)  
 print(p13); ggsave("MSNvsO\_figures/Filtered\_Dataset\_UMAP.png", p13, width = 10, height = 8)  
}

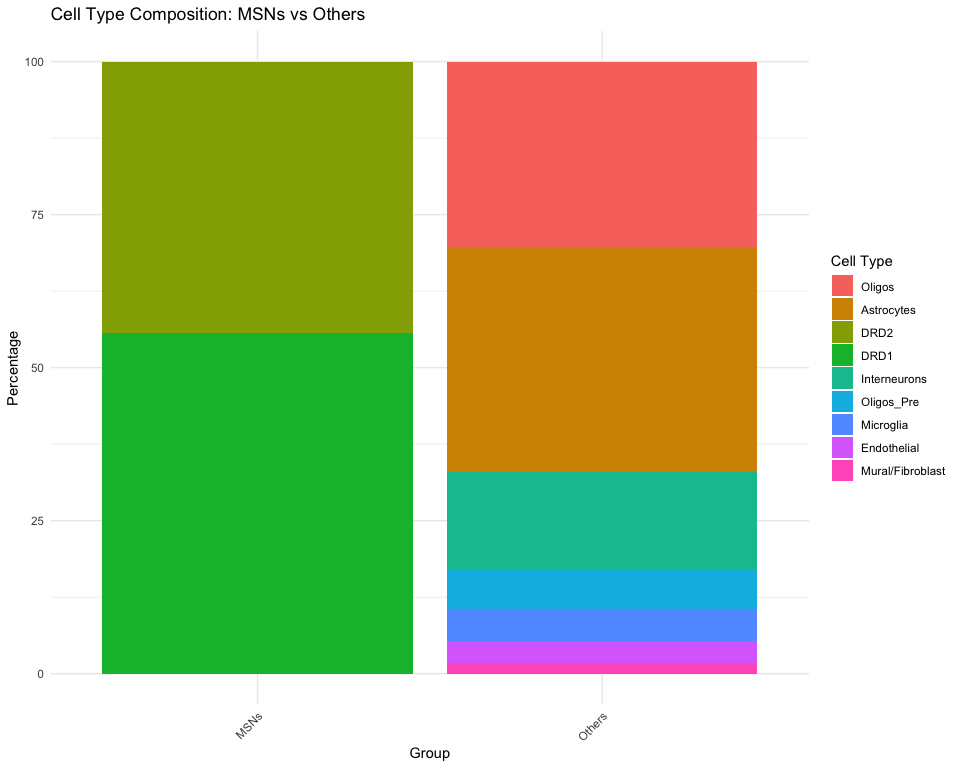


## Cell Type Composition Analysis

cat("Detailed Cell Type Composition Analysis\n")

## Detailed Cell Type Composition Analysis

# Cell type composition analysis  
if ("cell\_type" %in% colnames(so\_filtered@meta.data)) {  
 cell\_composition <- so\_filtered@meta.data %>%  
 group\_by(MSN\_vs\_Others, cell\_type) %>%  
 summarise(count = n(), .groups = "drop") %>%  
 group\_by(MSN\_vs\_Others) %>%  
 mutate(percentage = count/sum(count) \* 100)  
  
 # Visualize composition  
 p\_composition <- ggplot(cell\_composition, aes(x = MSN\_vs\_Others, y = percentage, fill = cell\_type)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 labs(title = "Cell Type Composition: MSNs vs Others",  
 x = "Group", y = "Percentage", fill = "Cell Type") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))  
   
 print(p\_composition)  
 ggsave("MSNvsO\_figures/Cell\_Type\_Composition.png", p\_composition, width = 12, height = 8)  
   
 # Save composition table  
 write.csv(cell\_composition, "MSNvsO\_tables/Cell\_Type\_Composition.csv", row.names = FALSE)  
 print(cell\_composition)  
}



## # A tibble: 9 × 4  
## # Groups: MSN\_vs\_Others [2]  
## MSN\_vs\_Others cell\_type count percentage  
## <chr> <fct> <int> <dbl>  
## 1 MSNs DRD2 7278 44.4   
## 2 MSNs DRD1 9117 55.6   
## 3 Others Oligos 14311 30.3   
## 4 Others Astrocytes 17422 36.8   
## 5 Others Interneurons 7504 15.9   
## 6 Others Oligos\_Pre 3078 6.51  
## 7 Others Microglia 2430 5.14  
## 8 Others Endothelial 1716 3.63  
## 9 Others Mural/Fibroblast 827 1.75

# Quality metrics by group  
qc\_by\_group <- so\_filtered@meta.data %>%  
 group\_by(MSN\_vs\_Others) %>%  
 summarise(  
 mean\_nFeatures = mean(nFeatures\_RNA),  
 mean\_nCounts = mean(nCount.RNA),  
 mean\_percent\_ribo = mean(percent.ribo),  
 median\_nFeatures = median(nFeatures\_RNA),  
 median\_nCounts = median(nCount.RNA),  
 .groups = "drop"  
 )  
  
print(qc\_by\_group)

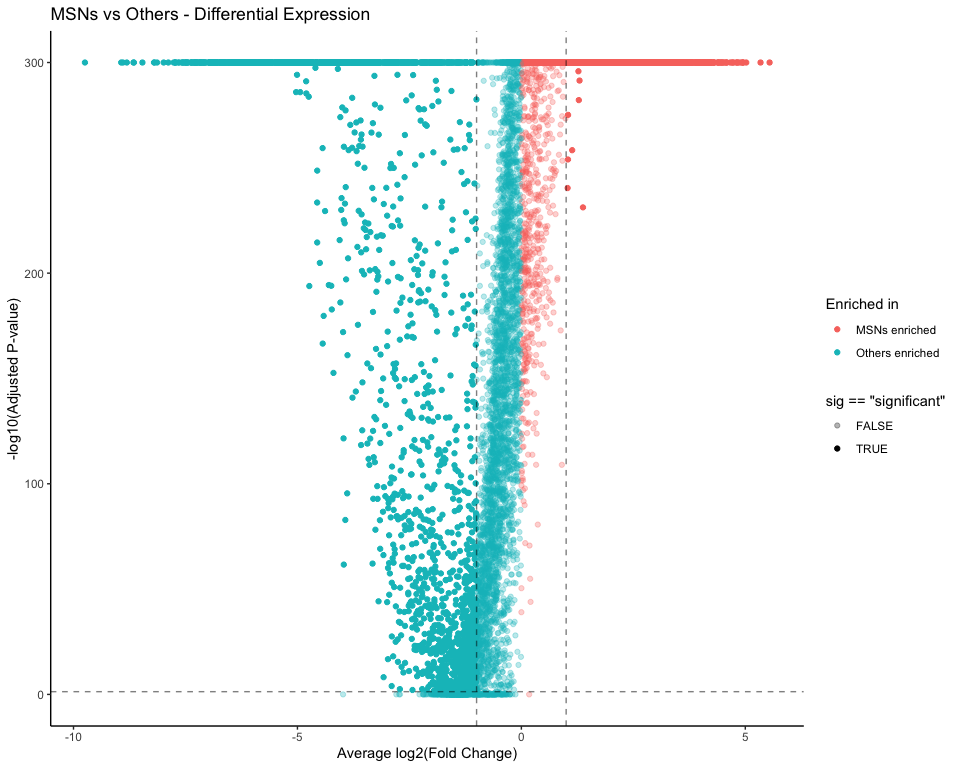
## # A tibble: 2 × 6  
## MSN\_vs\_Others mean\_nFeatures mean\_nCounts mean\_percent\_ribo median\_nFeatures  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 MSNs 4607. 13001. 0.00256 4601  
## 2 Others 2703. 5069. 0.00232 2421  
## # ℹ 1 more variable: median\_nCounts <dbl>

write.csv(qc\_by\_group, "MSNvsO\_tables/QC\_Metrics\_by\_Group.csv", row.names = FALSE)

## Enhanced Differential Expression Analysis

if (ncol(so\_filtered) > 50) {  
 cat("Enhanced Differential Expression Analysis: MSNs vs Others\n")  
  
 Idents(so\_filtered) <- so\_filtered$MSN\_vs\_Others  
   
 cat("Comparing: MSNs vs Others\n")  
 print(table(Idents(so\_filtered)))  
  
 # Run differential expression  
 markers\_msn\_vs\_others <- FindMarkers(  
 so\_filtered,  
 ident.1 = "MSNs",  
 ident.2 = "Others",  
 only.pos = FALSE,  
 min.pct = 0.1,  
 logfc.threshold = 0  
 )  
  
 markers\_msn\_vs\_others$gene <- rownames(markers\_msn\_vs\_others)  
  
 write.csv(markers\_msn\_vs\_others, "MSNvsO\_tables/DE\_MSNs\_vs\_Others.csv", row.names = FALSE)  
  
 # Apply significance criteria  
 sc\_sig <- markers\_msn\_vs\_others %>%  
 filter(!is.na(p\_val\_adj)) %>%  
 filter(p\_val\_adj < 0.05, abs(avg\_log2FC) > 1)  
  
 write.csv(sc\_sig, "MSNvsO\_tables/DE\_MSNs\_vs\_Others\_significant.csv", row.names = FALSE)  
  
 cat("Top Differential Genes (significant)\n")  
 print(head(sc\_sig[order(sc\_sig$p\_val\_adj), ], 10))  
  
 # Volcano plot  
 dfv <- markers\_msn\_vs\_others %>%  
 mutate(  
 sig = ifelse(!is.na(p\_val\_adj) & p\_val\_adj < 0.05 & abs(avg\_log2FC) > 1, "significant", "ns"),  
 log10\_padj = -log10(p\_val\_adj + 1e-300),  
 dir = ifelse(avg\_log2FC > 0, "MSNs enriched", "Others enriched")  
 )  
  
 p14 <- ggplot(dfv, aes(x = avg\_log2FC, y = log10\_padj)) +  
 geom\_point(aes(alpha = sig == "significant", color = dir)) +  
 scale\_alpha\_manual(values = c("TRUE" = 1, "FALSE" = 0.3)) +  
 labs(x = "Average log2(Fold Change)", y = "-log10(Adjusted P-value)",  
 title = "MSNs vs Others - Differential Expression",  
 color = "Enriched in") +  
 theme\_classic() +  
 geom\_hline(yintercept = -log10(0.05), linetype = "dashed", alpha = 0.5) +  
 geom\_vline(xintercept = c(-1, 1), linetype = "dashed", alpha = 0.5)  
   
 print(p14)  
 ggsave("MSNvsO\_figures/Volcano\_Plot\_MSNs\_vs\_Others.png", p14, width = 12, height = 8)  
  
} else {  
 cat("Insufficient cells for differential expression analysis\n")  
}

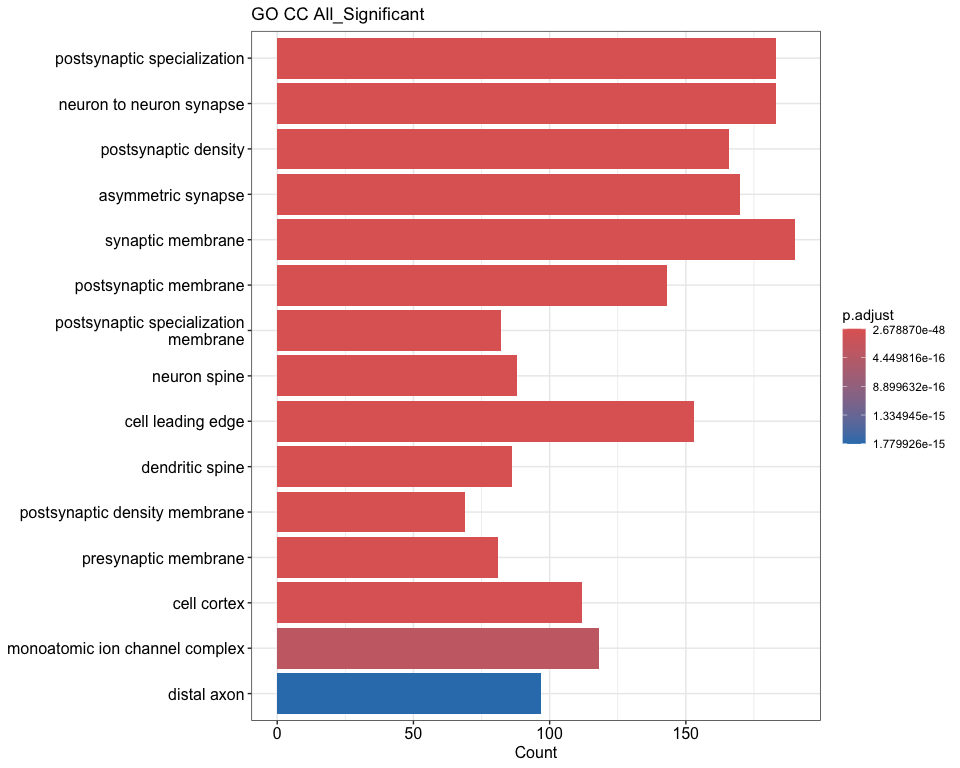
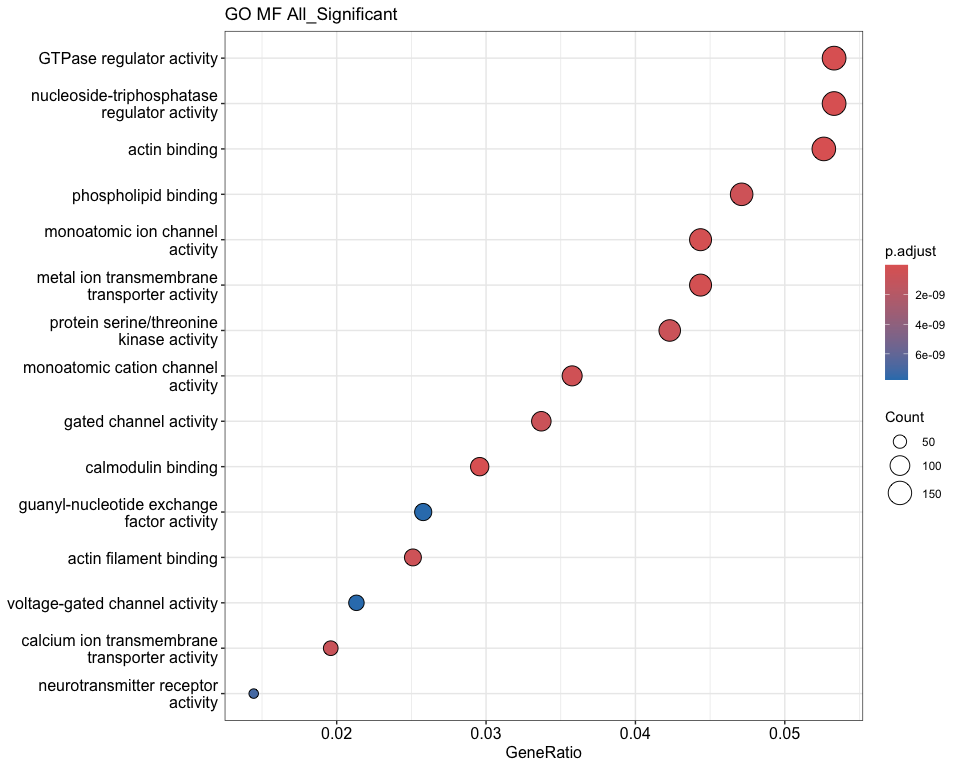
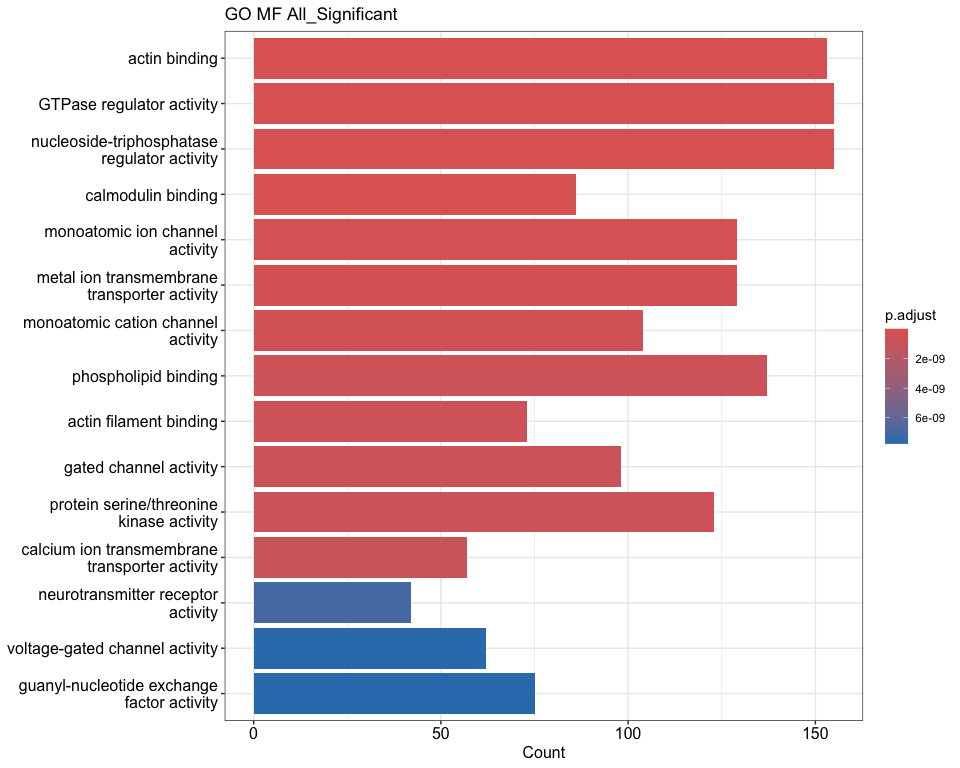
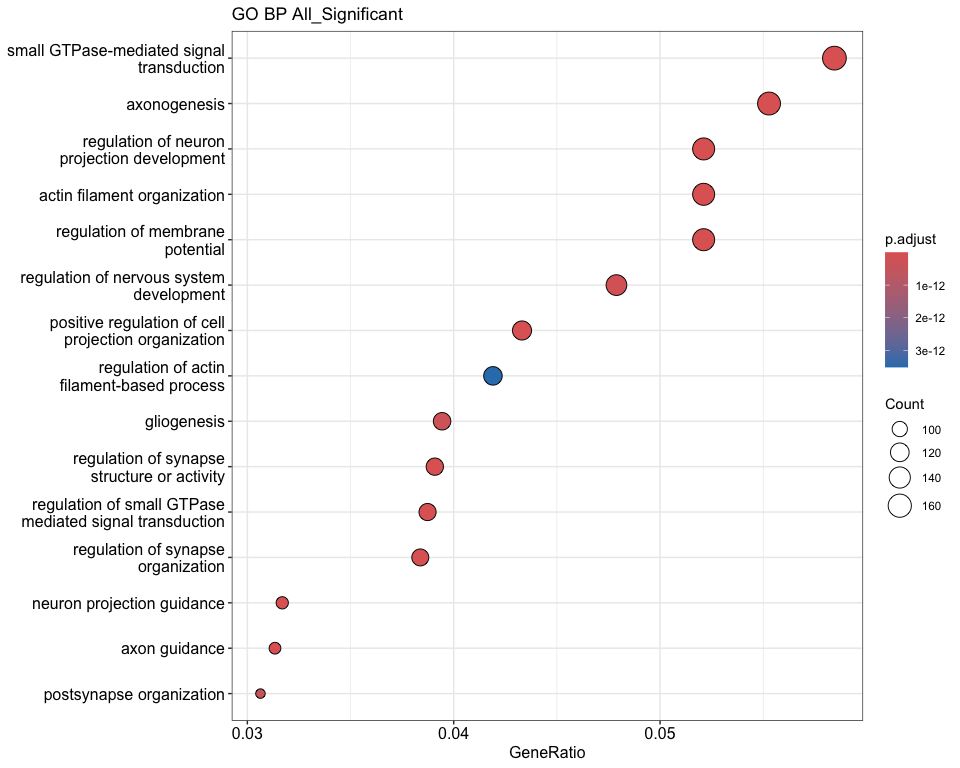
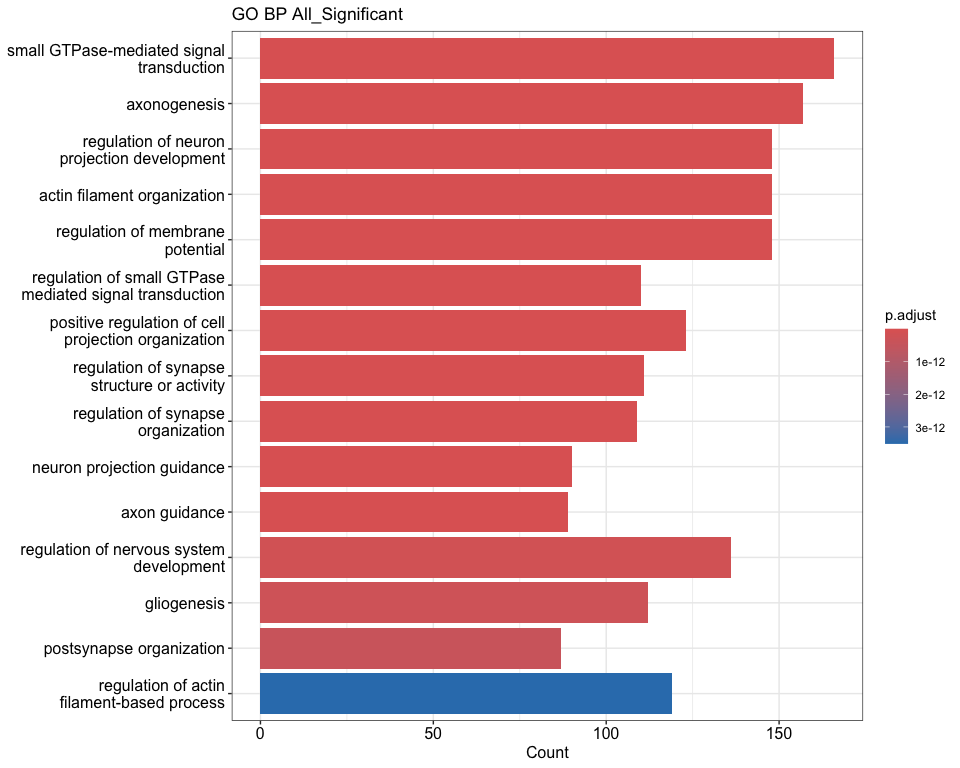
## Enhanced Differential Expression Analysis: MSNs vs Others  
## Comparing: MSNs vs Others  
##   
## MSNs Others   
## 16395 47288   
## Top Differential Genes (significant)  
## p\_val avg\_log2FC pct.1 pct.2 p\_val\_adj gene  
## KIAA1211L 0 4.198860 0.980 0.108 0 KIAA1211L  
## PDE1B 0 4.954113 0.965 0.113 0 PDE1B  
## NGEF 0 4.389232 0.954 0.116 0 NGEF  
## BCL11B 0 3.917073 0.985 0.153 0 BCL11B  
## CA12 0 4.915865 0.882 0.056 0 CA12  
## CACNA1H 0 4.183185 0.899 0.074 0 CACNA1H  
## PDE2A 0 3.908405 0.957 0.135 0 PDE2A  
## PTPN5 0 4.228786 0.975 0.158 0 PTPN5  
## AC005906.2 0 3.832358 0.957 0.148 0 AC005906.2  
## TESC 0 4.579178 0.866 0.057 0 TESC



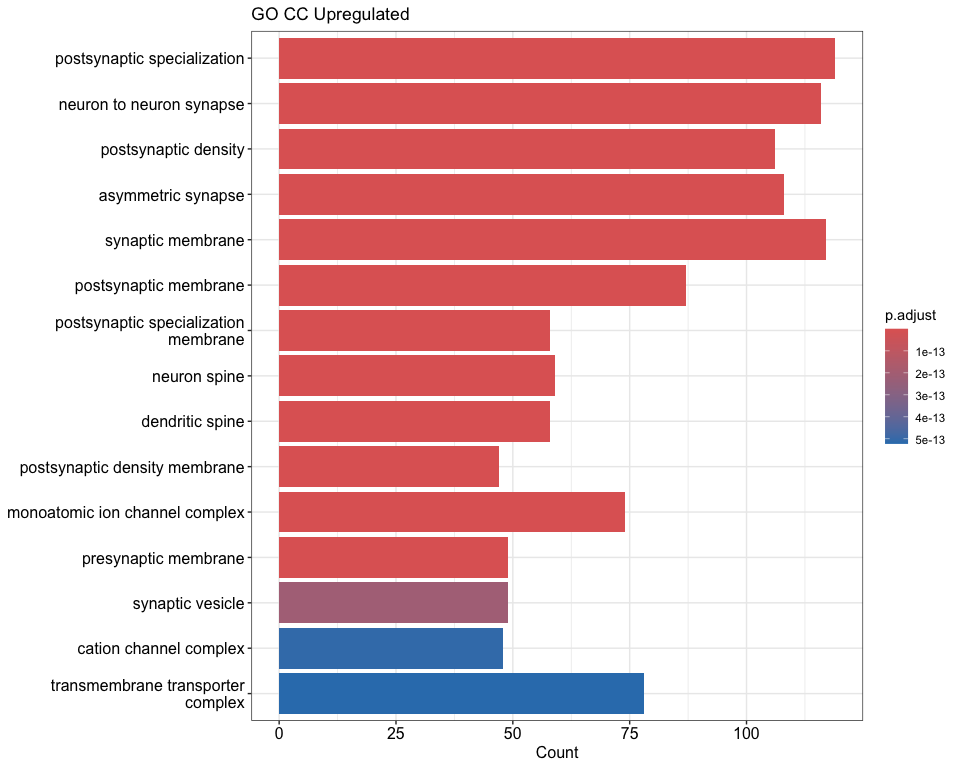
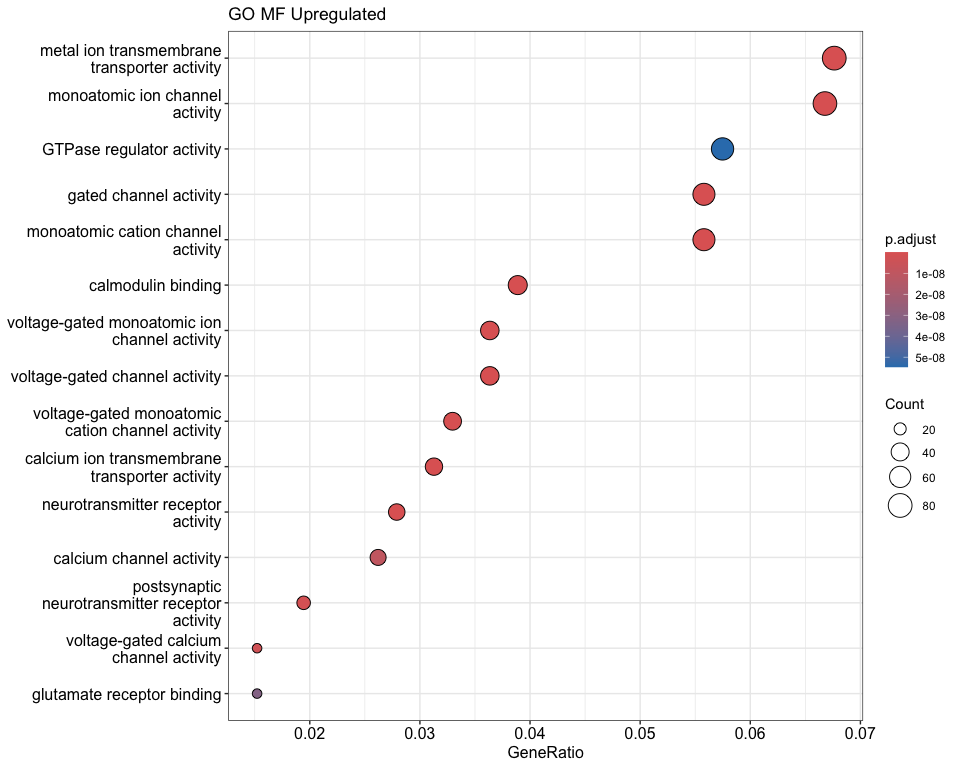
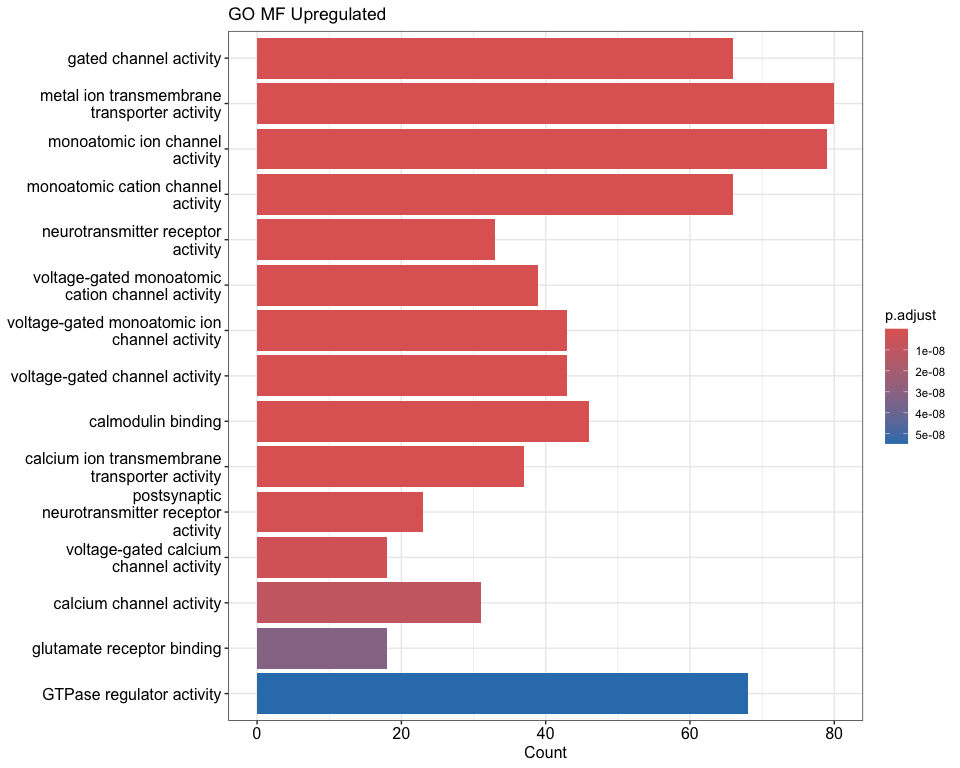
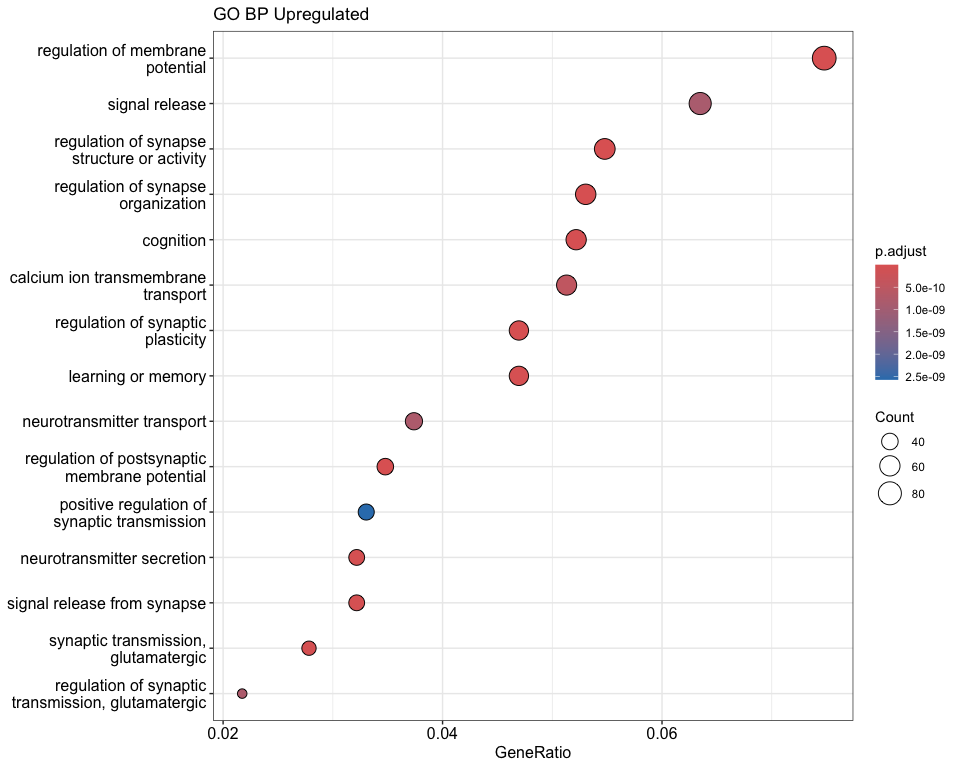
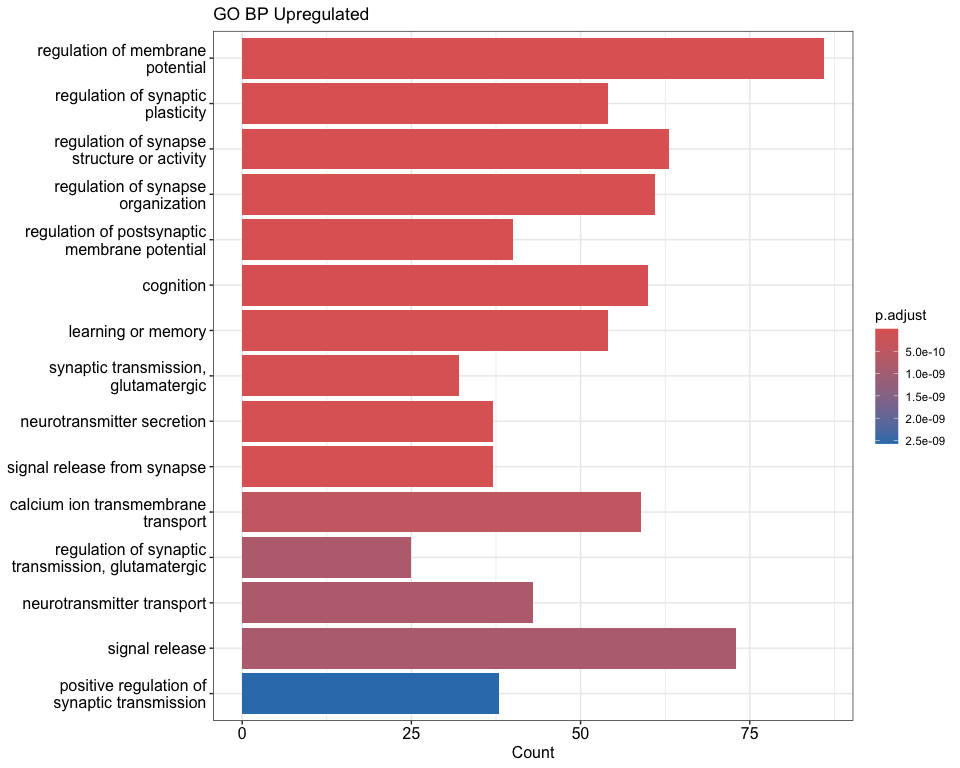
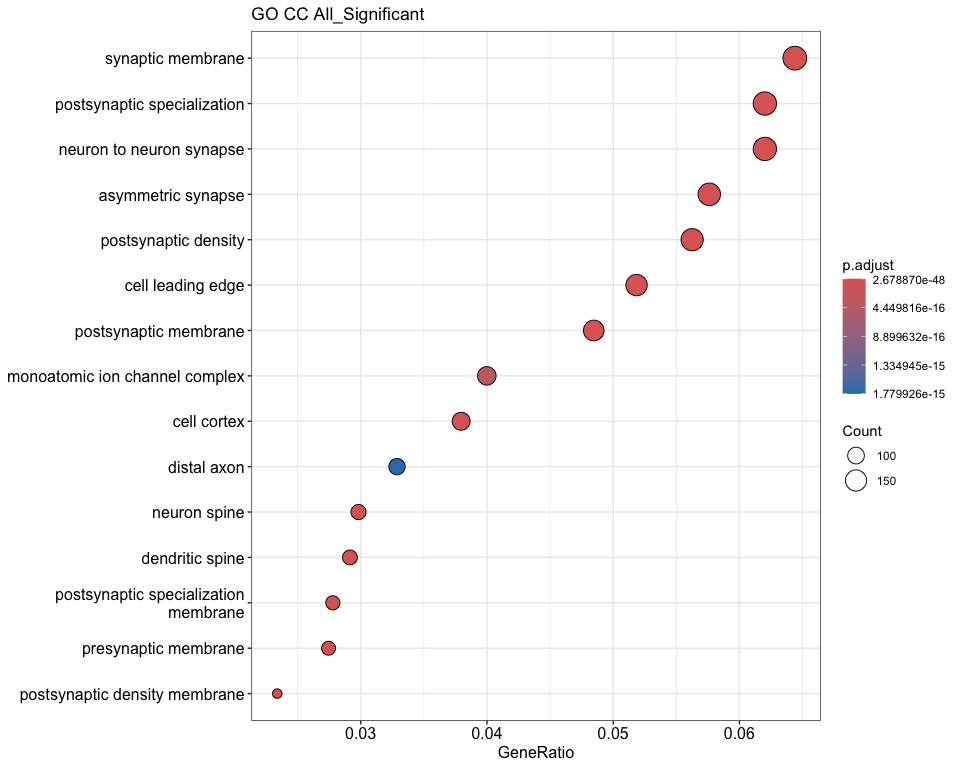
## GO and KEGG Enrichment Analysis

if (exists("markers\_msn\_vs\_others") && nrow(markers\_msn\_vs\_others) > 100) {  
 cat("Running GO and KEGG Enrichment Analysis\n")  
   
 # Get significant genes for enrichment  
 sig\_genes\_up <- markers\_msn\_vs\_others %>%  
 filter(!is.na(p\_val\_adj), p\_val\_adj < 0.05, avg\_log2FC > 1) %>%  
 pull(gene)  
   
 sig\_genes\_down <- markers\_msn\_vs\_others %>%  
 filter(!is.na(p\_val\_adj), p\_val\_adj < 0.05, avg\_log2FC < -1) %>%  
 pull(gene)  
   
 sig\_genes\_all <- markers\_msn\_vs\_others %>%  
 filter(!is.na(p\_val\_adj), p\_val\_adj < 0.05, abs(avg\_log2FC) > 1) %>%  
 pull(gene)  
   
 cat(paste("Upregulated genes:", length(sig\_genes\_up)), "\n")  
 cat(paste("Downregulated genes:", length(sig\_genes\_down)), "\n")  
 cat(paste("Total significant genes:", length(sig\_genes\_all)), "\n")  
   
 # Function to run GO enrichment  
 run\_go\_enrichment <- function(gene\_list, ont\_type, title\_suffix) {  
 if (length(gene\_list) < 10) {  
 cat(paste("Too few genes for", title\_suffix, "analysis\n"))  
 return(NULL)  
 }  
   
 # Filter genes that exist in the annotation database  
 valid\_genes <- gene\_list[gene\_list %in% keys(org.Hs.eg.db, keytype = "SYMBOL")]  
   
 if (length(valid\_genes) < 10) {  
 cat(paste("Too few valid genes for", title\_suffix, "analysis\n"))  
 return(NULL)  
 }  
   
 ego <- enrichGO(  
 gene = valid\_genes,  
 OrgDb = org.Hs.eg.db,  
 keyType = "SYMBOL",  
 ont = ont\_type,  
 pAdjustMethod = "BH",  
 pvalueCutoff = 0.05,  
 qvalueCutoff = 0.2,  
 readable = TRUE  
 )  
   
 if (is.null(ego) || nrow(as.data.frame(ego)) == 0) {  
 cat(paste("No significant GO terms found for", title\_suffix), "\n")  
 return(NULL)  
 }  
   
 ego\_df <- as.data.frame(ego)  
   
 # Save results  
 write.csv(ego\_df, paste0("MSNvsO\_tables/GO\_", ont\_type, "\_", title\_suffix, ".csv"), row.names = FALSE)  
   
 # Create plots  
 if (nrow(ego\_df) > 0) {  
 p\_bar <- barplot(ego, showCategory = 15, title = paste("GO", ont\_type, title\_suffix))  
 p\_dot <- dotplot(ego, showCategory = 15, title = paste("GO", ont\_type, title\_suffix))  
   
 ggsave(paste0("MSNvsO\_figures/GO\_", ont\_type, "\_", title\_suffix, "\_Barplot.png"),   
 p\_bar, width = 12, height = 8)  
 ggsave(paste0("MSNvsO\_figures/GO\_", ont\_type, "\_", title\_suffix, "\_Dotplot.png"),   
 p\_dot, width = 12, height = 8)  
   
 print(p\_bar)  
 print(p\_dot)  
 }  
   
 return(ego)  
 }  
   
 # Function to run KEGG enrichment  
 run\_kegg\_enrichment <- function(gene\_list, title\_suffix) {  
 if (length(gene\_list) < 10) {  
 cat(paste("Too few genes for KEGG", title\_suffix, "analysis\n"))  
 return(NULL)  
 }  
   
 # Convert symbols to ENTREZ IDs  
 gene\_entrez <- bitr(gene\_list, fromType = "SYMBOL", toType = "ENTREZID",   
 OrgDb = org.Hs.eg.db, drop = TRUE)  
   
 if (nrow(gene\_entrez) < 10) {  
 cat(paste("Too few mappable genes for KEGG", title\_suffix, "analysis\n"))  
 return(NULL)  
 }  
   
 ekegg <- enrichKEGG(  
 gene = gene\_entrez$ENTREZID,  
 organism = "hsa",  
 pvalueCutoff = 0.05,  
 pAdjustMethod = "BH",  
 qvalueCutoff = 0.2  
 )  
   
 if (is.null(ekegg) || nrow(as.data.frame(ekegg)) == 0) {  
 cat(paste("No significant KEGG pathways found for", title\_suffix), "\n")  
 return(NULL)  
 }  
   
 # Convert back to gene symbols for readability  
 ekegg <- setReadable(ekegg, OrgDb = org.Hs.eg.db, keyType = "ENTREZID")  
 ekegg\_df <- as.data.frame(ekegg)  
   
 # Save results  
 write.csv(ekegg\_df, paste0("MSNvsO\_tables/KEGG\_", title\_suffix, ".csv"), row.names = FALSE)  
   
 # Create plots  
 if (nrow(ekegg\_df) > 0) {  
 p\_bar <- barplot(ekegg, showCategory = 15, title = paste("KEGG Pathways", title\_suffix))  
 p\_dot <- dotplot(ekegg, showCategory = 15, title = paste("KEGG Pathways", title\_suffix))  
   
 ggsave(paste0("MSNvsO\_figures/KEGG\_", title\_suffix, "\_Barplot.png"),   
 p\_bar, width = 12, height = 8)  
 ggsave(paste0("MSNvsO\_figures/KEGG\_", title\_suffix, "\_Dotplot.png"),   
 p\_dot, width = 12, height = 8)  
   
 print(p\_bar)  
 print(p\_dot)  
 }  
   
 return(ekegg)  
 }  
   
 # Run GO enrichment for all significant genes  
 if (length(sig\_genes\_all) >= 10) {  
 cat("Running GO enrichment for all significant genes\n")  
 ego\_bp\_all <- run\_go\_enrichment(sig\_genes\_all, "BP", "All\_Significant")  
 ego\_mf\_all <- run\_go\_enrichment(sig\_genes\_all, "MF", "All\_Significant")  
 ego\_cc\_all <- run\_go\_enrichment(sig\_genes\_all, "CC", "All\_Significant")  
 }  
   
 # Run GO enrichment for upregulated genes  
 if (length(sig\_genes\_up) >= 10) {  
 cat("Running GO enrichment for upregulated genes\n")  
 ego\_bp\_up <- run\_go\_enrichment(sig\_genes\_up, "BP", "Upregulated")  
 ego\_mf\_up <- run\_go\_enrichment(sig\_genes\_up, "MF", "Upregulated")  
 ego\_cc\_up <- run\_go\_enrichment(sig\_genes\_up, "CC", "Upregulated")  
 }  
   
 # Run GO enrichment for downregulated genes  
 if (length(sig\_genes\_down) >= 10) {  
 cat("Running GO enrichment for downregulated genes\n")  
 ego\_bp\_down <- run\_go\_enrichment(sig\_genes\_down, "BP", "Downregulated")  
 ego\_mf\_down <- run\_go\_enrichment(sig\_genes\_down, "MF", "Downregulated")  
 ego\_cc\_down <- run\_go\_enrichment(sig\_genes\_down, "CC", "Downregulated")  
 }  
   
 # Run KEGG enrichment  
 if (length(sig\_genes\_all) >= 10) {  
 cat("Running KEGG enrichment for all significant genes\n")  
 ekegg\_all <- run\_kegg\_enrichment(sig\_genes\_all, "All\_Significant")  
 }  
   
 if (length(sig\_genes\_up) >= 10) {  
 cat("Running KEGG enrichment for upregulated genes\n")  
 ekegg\_up <- run\_kegg\_enrichment(sig\_genes\_up, "Upregulated")  
 }  
   
 if (length(sig\_genes\_down) >= 10) {  
 cat("Running KEGG enrichment for downregulated genes\n")  
 ekegg\_down <- run\_kegg\_enrichment(sig\_genes\_down, "Downregulated")  
 }  
   
 # Create comparison plots if multiple enrichment results exist  
 if (exists("ego\_bp\_up") && exists("ego\_bp\_down") &&   
 !is.null(ego\_bp\_up) && !is.null(ego\_bp\_down)) {  
   
 # Compare upregulated vs downregulated GO BP terms  
 comparison\_plot <- compareCluster(  
 list(Upregulated = sig\_genes\_up, Downregulated = sig\_genes\_down),  
 fun = "enrichGO",  
 OrgDb = org.Hs.eg.db,  
 ont = "BP",  
 pAdjustMethod = "BH",  
 pvalueCutoff = 0.05,  
 readable = TRUE  
 )  
   
 if (!is.null(comparison\_plot) && nrow(as.data.frame(comparison\_plot)) > 0) {  
 p\_compare <- dotplot(comparison\_plot, showCategory = 10,   
 title = "GO Biological Process: Up vs Down Regulated")  
 ggsave("MSNvsO\_figures/GO\_BP\_Comparison\_Up\_vs\_Down.png",   
 p\_compare, width = 14, height = 10)  
 print(p\_compare)  
   
 write.csv(as.data.frame(comparison\_plot),   
 "MSNvsO\_tables/GO\_BP\_Comparison\_Up\_vs\_Down.csv", row.names = FALSE)  
 }  
 }  
   
 # Create enrichment summary  
 enrichment\_summary <- data.frame(  
 Analysis = character(),  
 Category = character(),  
 Significant\_Terms = numeric(),  
 Top\_Term = character(),  
 Top\_Term\_Pvalue = numeric()  
 )  
   
 # Function to add to summary  
 add\_to\_summary <- function(result, analysis\_name, category\_name) {  
 if (!is.null(result) && nrow(as.data.frame(result)) > 0) {  
 result\_df <- as.data.frame(result)  
 return(data.frame(  
 Analysis = analysis\_name,  
 Category = category\_name,  
 Significant\_Terms = nrow(result\_df),  
 Top\_Term = result\_df$Description[1],  
 Top\_Term\_Pvalue = result\_df$p.adjust[1]  
 ))  
 }  
 return(NULL)  
 }  
   
 # Add all results to summary  
 summary\_additions <- list()  
 if (exists("ego\_bp\_all")) summary\_additions <- append(summary\_additions, list(add\_to\_summary(ego\_bp\_all, "All\_Significant", "GO\_BP")))  
 if (exists("ego\_mf\_all")) summary\_additions <- append(summary\_additions, list(add\_to\_summary(ego\_mf\_all, "All\_Significant", "GO\_MF")))  
 if (exists("ego\_cc\_all")) summary\_additions <- append(summary\_additions, list(add\_to\_summary(ego\_cc\_all, "All\_Significant", "GO\_CC")))  
 if (exists("ekegg\_all")) summary\_additions <- append(summary\_additions, list(add\_to\_summary(ekegg\_all, "All\_Significant", "KEGG")))  
   
 if (exists("ego\_bp\_up")) summary\_additions <- append(summary\_additions, list(add\_to\_summary(ego\_bp\_up, "Upregulated", "GO\_BP")))  
 if (exists("ego\_bp\_down")) summary\_additions <- append(summary\_additions, list(add\_to\_summary(ego\_bp\_down, "Downregulated", "GO\_BP")))  
 if (exists("ekegg\_up")) summary\_additions <- append(summary\_additions, list(add\_to\_summary(ekegg\_up, "Upregulated", "KEGG")))  
 if (exists("ekegg\_down")) summary\_additions <- append(summary\_additions, list(add\_to\_summary(ekegg\_down, "Downregulated", "KEGG")))  
   
 # Combine all summary additions  
 valid\_summaries <- summary\_additions[!sapply(summary\_additions, is.null)]  
 if (length(valid\_summaries) > 0) {  
 enrichment\_summary <- do.call(rbind, valid\_summaries)  
 write.csv(enrichment\_summary, "MSNvsO\_tables/Enrichment\_Analysis\_Summary.csv", row.names = FALSE)  
 print("Enrichment Analysis Summary:")  
 print(enrichment\_summary)  
   
 # Plot summary  
 if (nrow(enrichment\_summary) > 0) {  
 p\_summary <- ggplot(enrichment\_summary, aes(x = paste(Analysis, Category, sep = "\_"),   
 y = Significant\_Terms)) +  
 geom\_col(fill = "steelblue", alpha = 0.7) +  
 geom\_text(aes(label = Significant\_Terms), vjust = -0.5) +  
 labs(title = "Number of Significant Terms by Analysis",  
 x = "Analysis\_Category", y = "Number of Significant Terms") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))  
   
 ggsave("MSNvsO\_figures/Enrichment\_Summary.png", p\_summary, width = 12, height = 6)  
 print(p\_summary)  
 }  
 }  
   
} else {  
 cat("Insufficient genes for GO and KEGG enrichment analysis\n")  
}

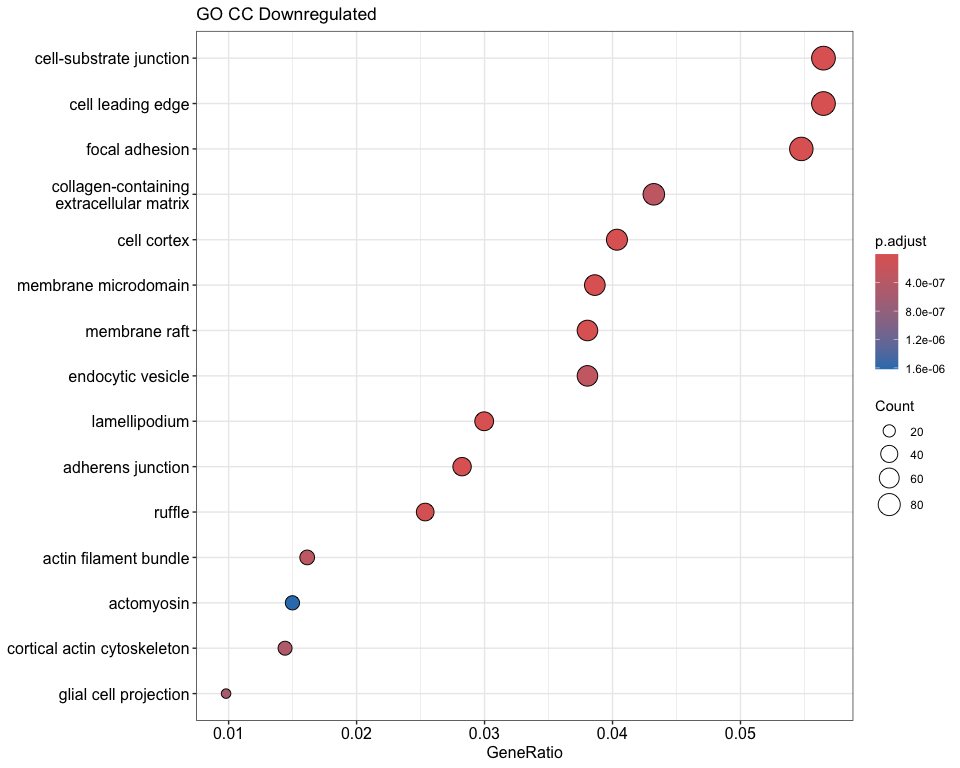
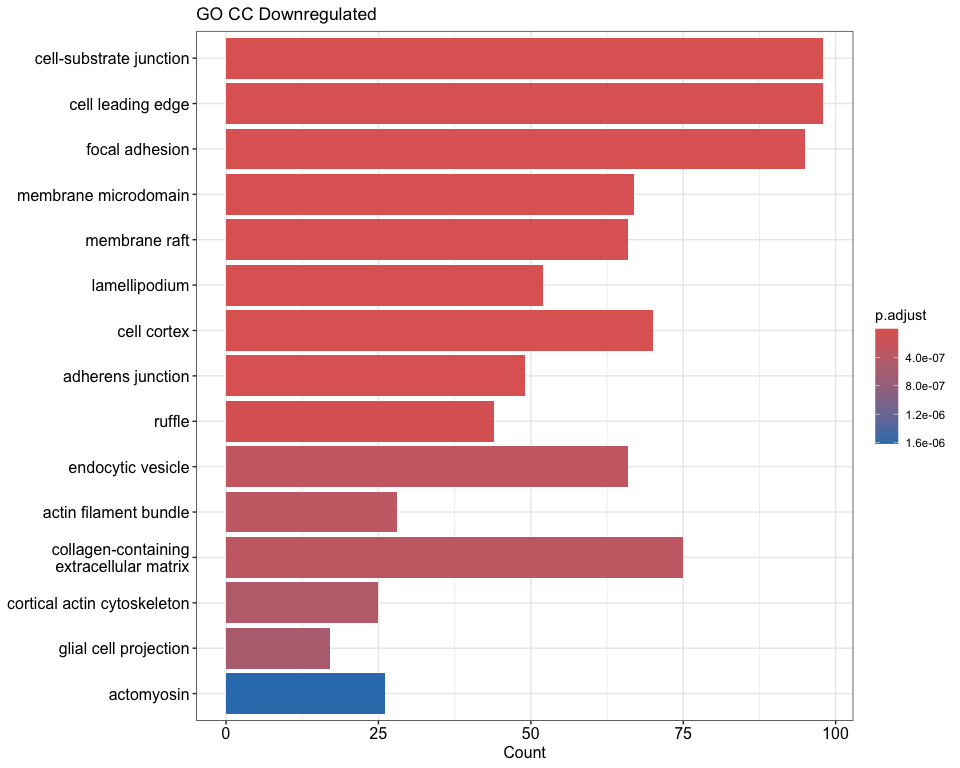
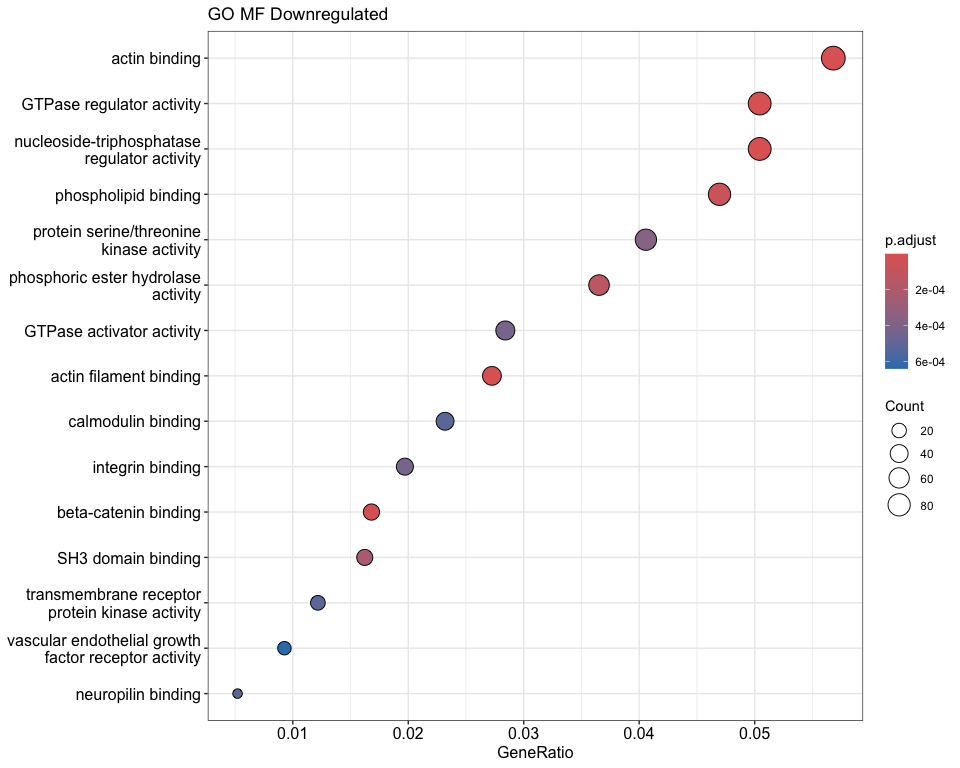
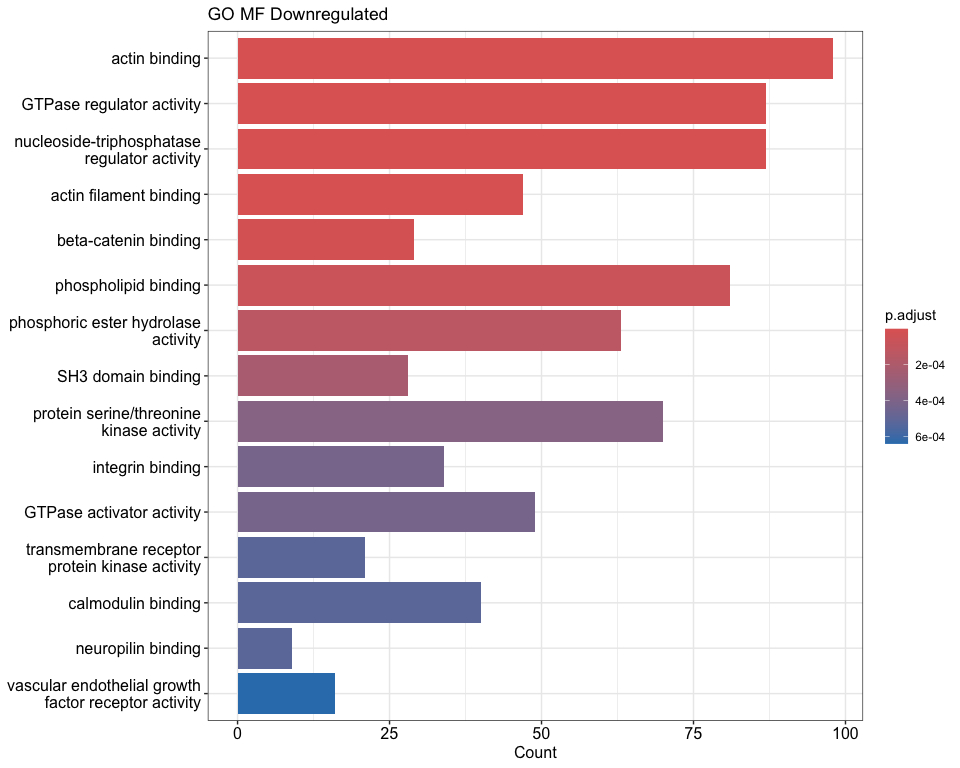
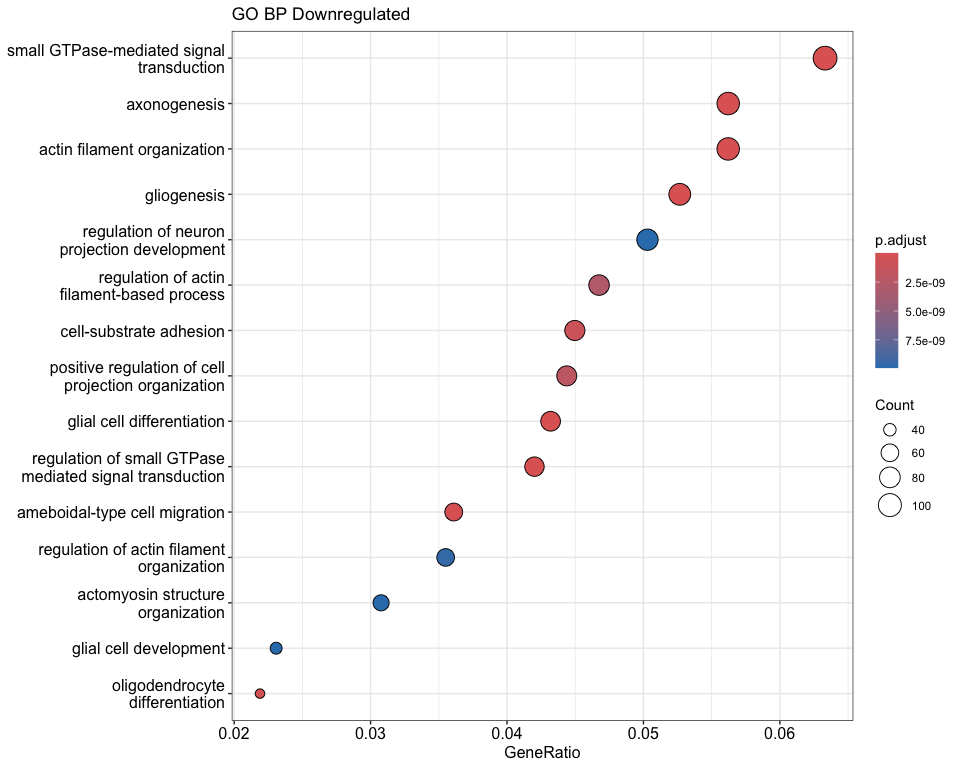
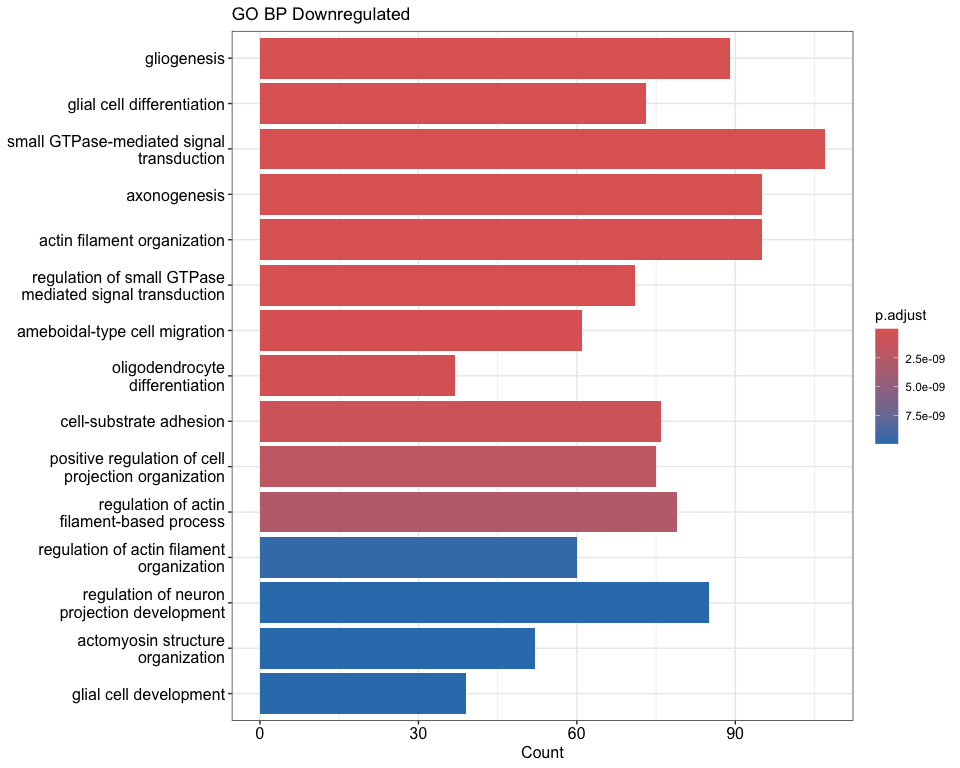
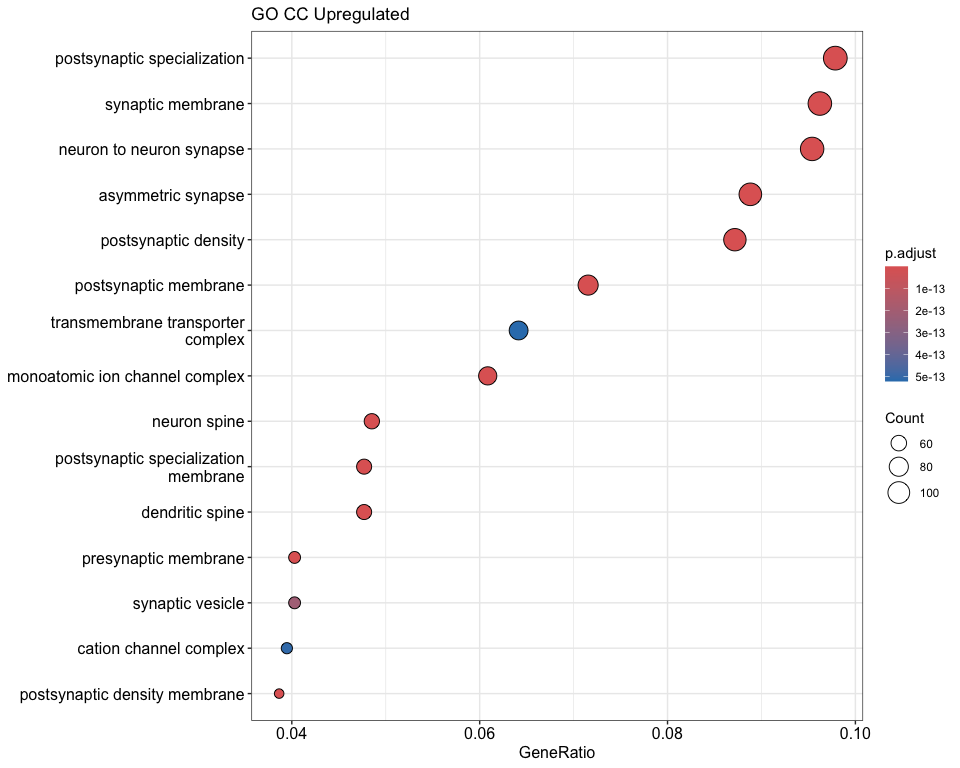
## Running GO and KEGG Enrichment Analysis  
## Upregulated genes: 1742   
## Downregulated genes: 2101   
## Total significant genes: 3843   
## Running GO enrichment for all significant genes



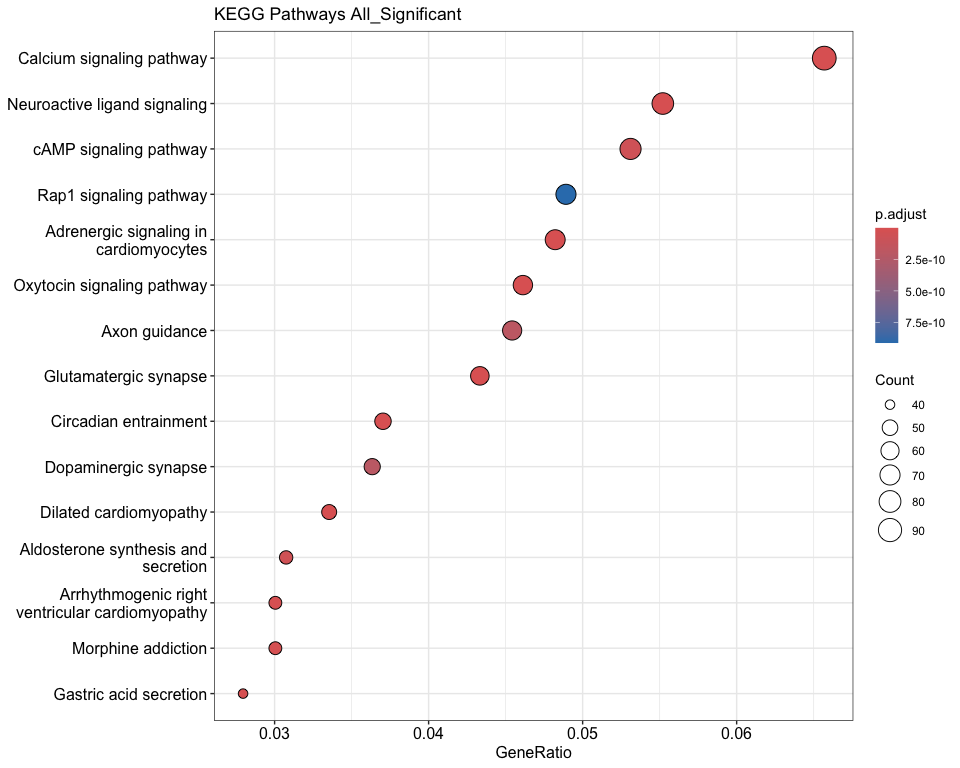
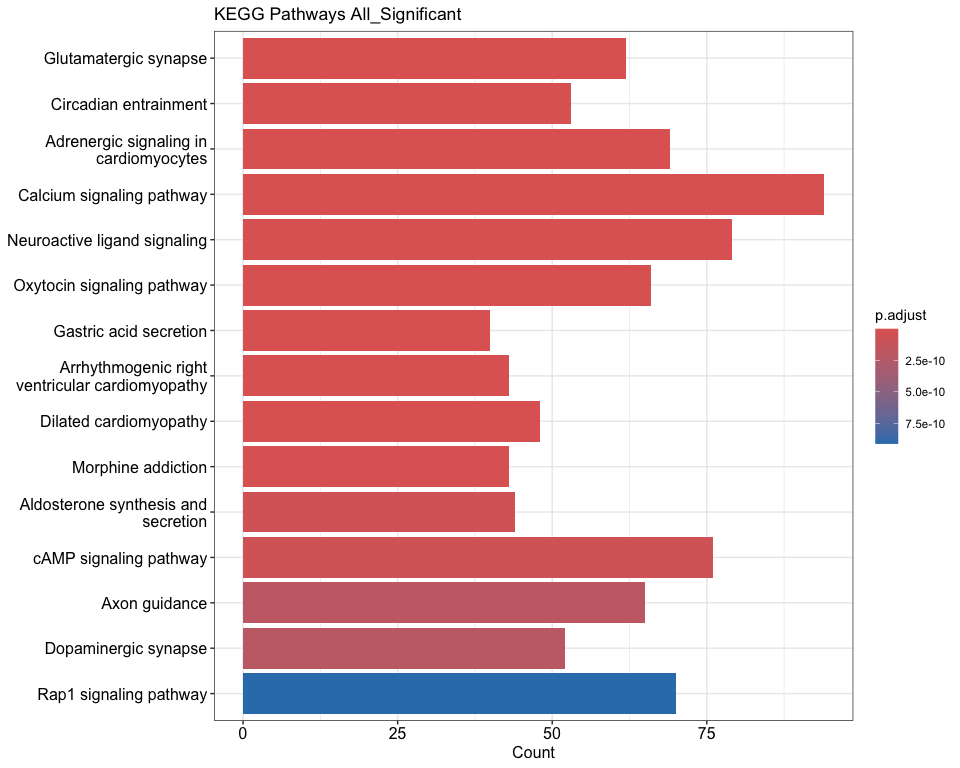
## Running GO enrichment for upregulated genes



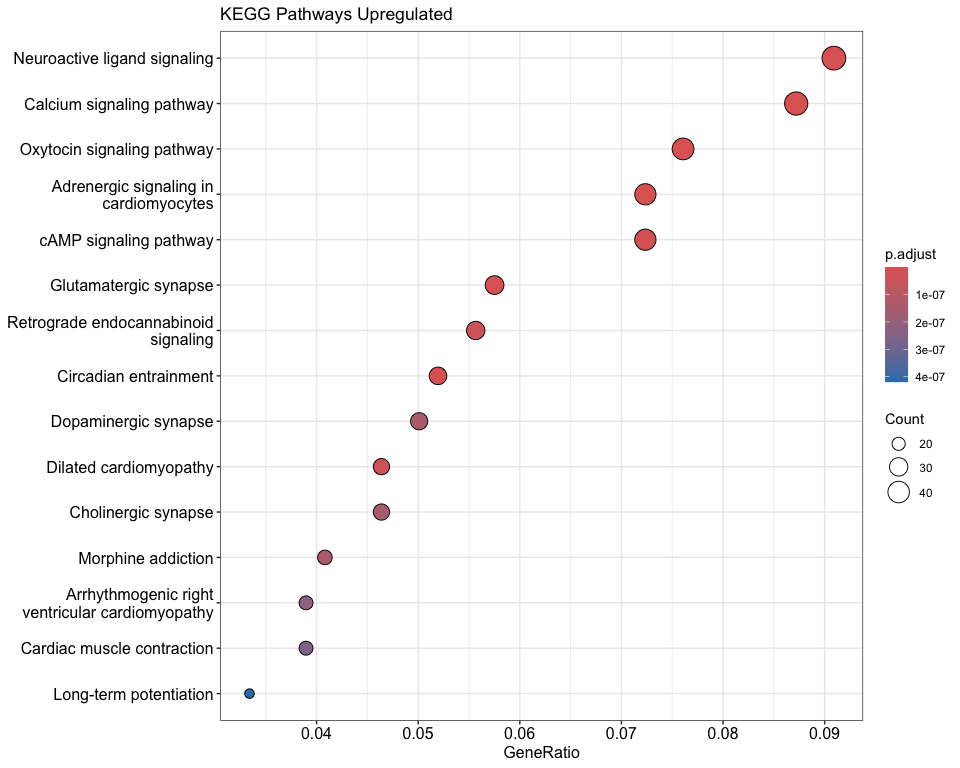
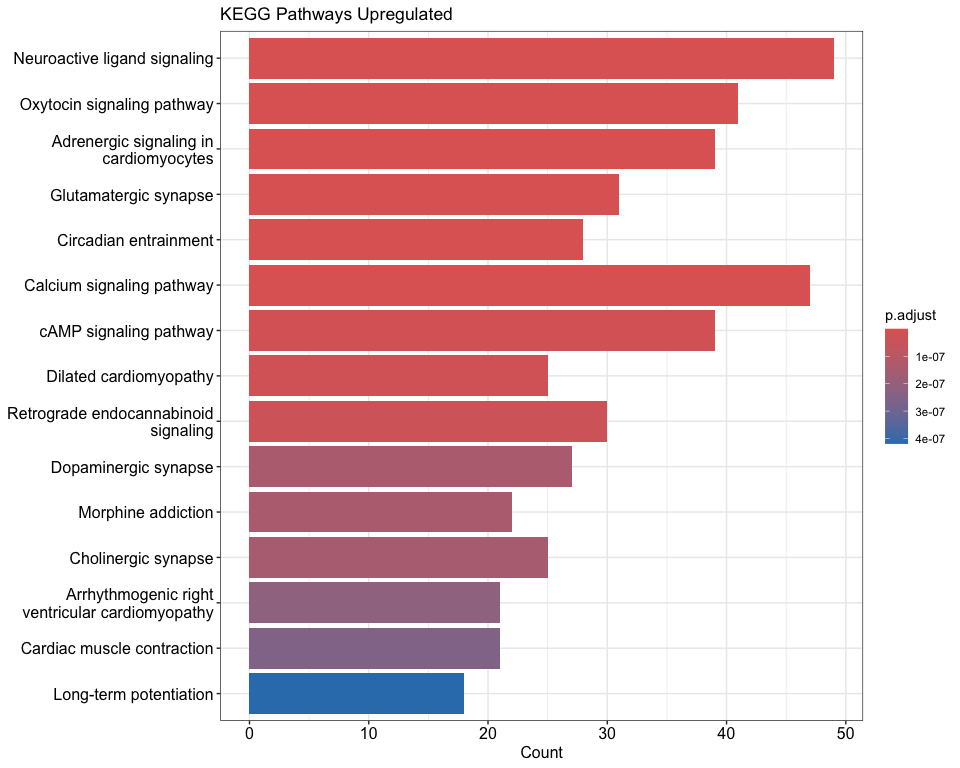
## Running GO enrichment for downregulated genes



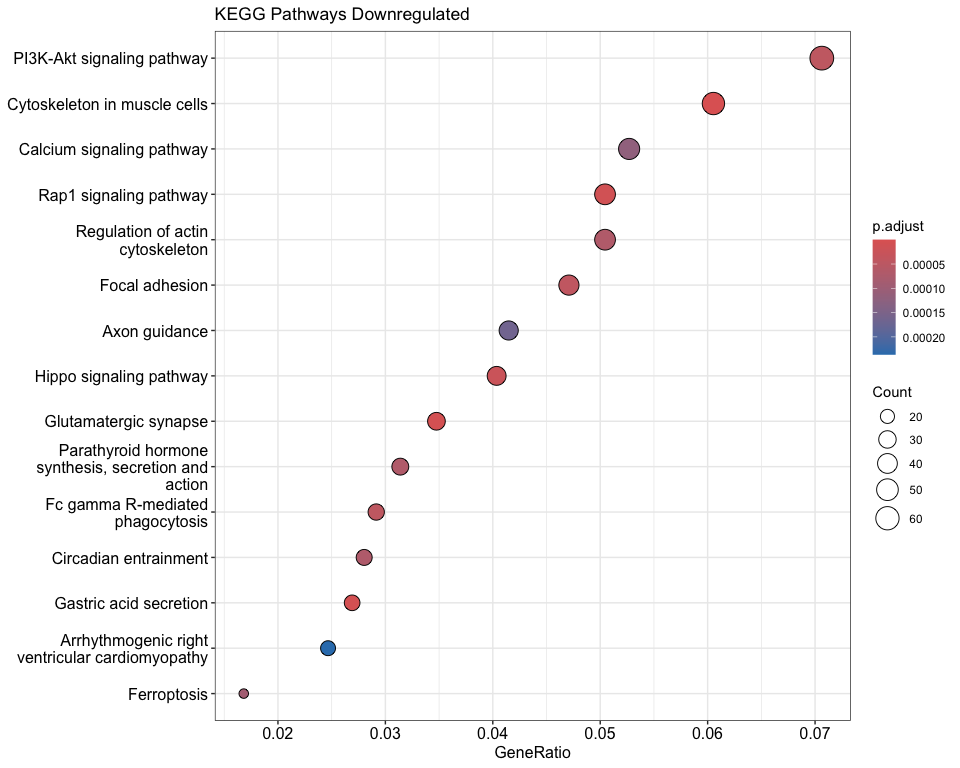
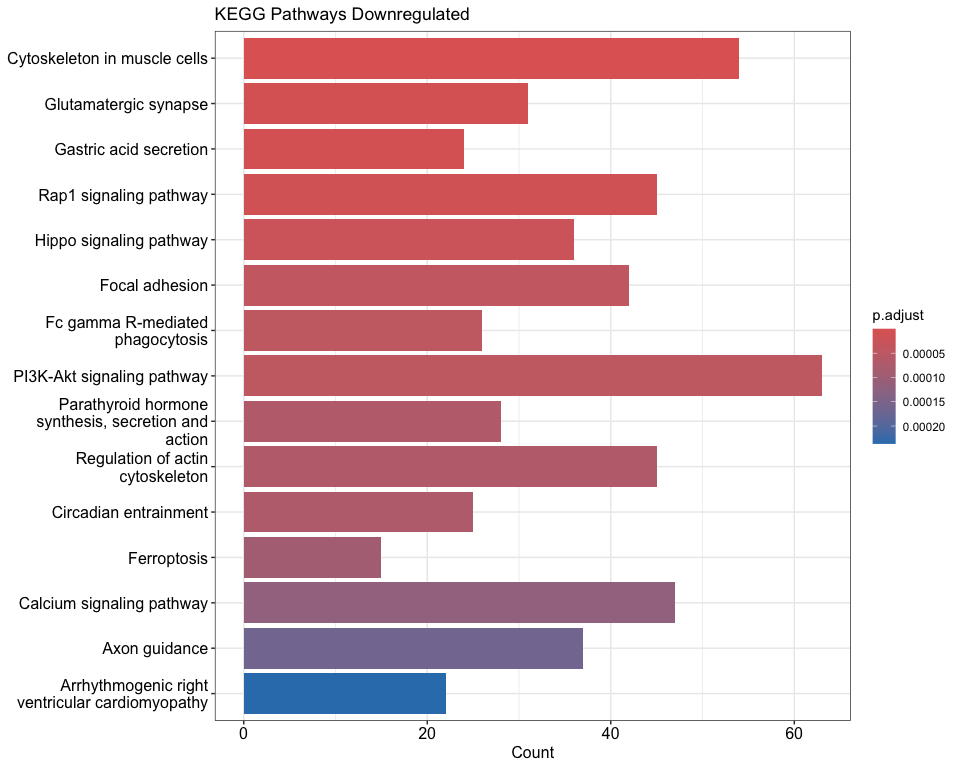
## Running KEGG enrichment for all significant genes



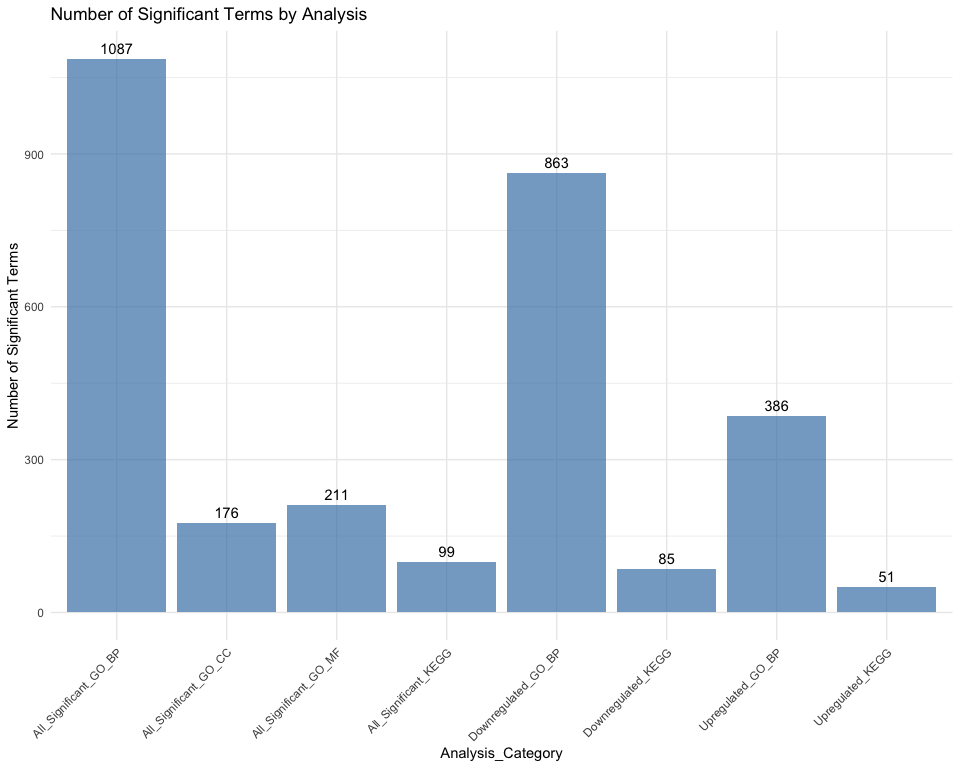
## Running KEGG enrichment for upregulated genes



## Running KEGG enrichment for downregulated genes



## [1] "Enrichment Analysis Summary:"  
## Analysis Category Significant\_Terms  
## 1 All\_Significant GO\_BP 1087  
## 2 All\_Significant GO\_MF 211  
## 3 All\_Significant GO\_CC 176  
## 4 All\_Significant KEGG 99  
## 5 Upregulated GO\_BP 386  
## 6 Downregulated GO\_BP 863  
## 7 Upregulated KEGG 51  
## 8 Downregulated KEGG 85  
## Top\_Term Top\_Term\_Pvalue  
## 1 small GTPase-mediated signal transduction 2.246708e-22  
## 2 actin binding 2.852675e-20  
## 3 postsynaptic specialization 2.678870e-48  
## 4 Glutamatergic synapse 1.846911e-19  
## 5 regulation of membrane potential 4.469117e-17  
## 6 gliogenesis 2.145078e-16  
## 7 Neuroactive ligand signaling 1.801191e-16  
## 8 Cytoskeleton in muscle cells 8.947692e-08



## Epigenetic Regulator Expression Analysis

# Define epigenetic genes of interest  
epi\_genes <- c(  
 # DNA methylation  
 "DNMT1","DNMT3A","DNMT3B","TET1","TET2","TET3","UHRF1",  
 # Histone writers  
 "EZH2","EZH1","SUZ12","EED","KMT2A","KMT2D","SETD1A","SETD1B",  
 "KMT5A","KMT5B","KMT5C","EHMT1","EHMT2","SUV39H1","SUV39H2",  
 # Histone erasers  
 "KDM1A","KDM1B","KDM4A","KDM4B","KDM4C","KDM5A","KDM5B","KDM6A","KDM6B",  
 "HDAC1","HDAC2","HDAC3","SIRT1","SIRT2","SIRT6",  
 # Chromatin remodeling  
 "SMARCA4","SMARCA2","SMARCB1","CHD1","CHD2","CHD7","CHD8",  
 # DNA binding/regulation  
 "MECP2","MBD1","MBD2","MBD3","CTCF","SATB1","SATB2"  
)  
  
# Filter genes present in dataset  
present\_epi\_genes <- intersect(epi\_genes, rownames(so\_filtered))  
missing\_epi\_genes <- setdiff(epi\_genes, present\_epi\_genes)  
  
cat(paste("Present epigenetic genes:", length(present\_epi\_genes)), "\n")

## Present epigenetic genes: 48

cat(paste("Missing epigenetic genes:", length(missing\_epi\_genes)), "\n")

## Missing epigenetic genes: 3

if (length(present\_epi\_genes) > 0) {  
 # Calculate scores for a small subset of writer/eraser genes  
 histone\_writers <- intersect(c("EZH2", "KMT2A", "KMT2D", "SETD1A", "SETD1B"), rownames(so\_filtered))  
 histone\_erasers <- intersect(c("KDM1A", "KDM4A", "KDM6A", "KDM6B", "HDAC1", "HDAC2"), rownames(so\_filtered))  
  
 if (length(histone\_writers) > 0) {  
 so\_filtered$writer\_score <- colMeans(GetAssayData(so\_filtered, layer = "data")[histone\_writers, , drop = FALSE])  
 }  
 if (length(histone\_erasers) > 0) {  
 so\_filtered$eraser\_score <- colMeans(GetAssayData(so\_filtered, layer = "data")[histone\_erasers, , drop = FALSE])  
 }  
 if (length(histone\_writers) > 0 && length(histone\_erasers) > 0) {  
 so\_filtered$writer\_eraser\_ratio <- so\_filtered$writer\_score / (so\_filtered$eraser\_score + 0.1)  
 }  
  
 # Basic violin plots  
 if ("writer\_score" %in% colnames(so\_filtered@meta.data)) {  
 p\_writer <- VlnPlot(so\_filtered, features = "writer\_score", group.by = "MSN\_vs\_Others") +  
 ggtitle("Histone Writer Score")  
 ggsave("MSNvsO\_figures/Histone\_Writer\_Score.png", p\_writer, width = 8, height = 6)  
 }  
  
 if ("writer\_eraser\_ratio" %in% colnames(so\_filtered@meta.data)) {  
 p\_ratio <- VlnPlot(so\_filtered, features = "writer\_eraser\_ratio", group.by = "MSN\_vs\_Others") +  
 ggtitle("Writer/Eraser Ratio")  
 ggsave("MSNvsO\_figures/Writer\_Eraser\_Ratio.png", p\_ratio, width = 8, height = 6)  
 }  
  
 # Limit to top 6 epigenetic genes for compact plotting  
 top\_epi\_genes <- present\_epi\_genes[1:min(6, length(present\_epi\_genes))]  
  
 p18 <- VlnPlot(so\_filtered, features = top\_epi\_genes,  
 group.by = "MSN\_vs\_Others",  
 pt.size = 0, ncol = 3) +  
 theme\_minimal() + ggtitle("Top Epigenetic Regulator Expression")  
 ggsave("MSNvsO\_figures/Epigenetic\_Regulators\_ViolinPlot.png", p18, width = 12, height = 8)  
  
 # DotPlot for top genes only  
 p\_dot <- DotPlot(so\_filtered, features = top\_epi\_genes, group.by = "MSN\_vs\_Others") +  
 RotatedAxis() +  
 scale\_color\_gradientn(colors = c("skyblue", "white", "coral")) +  
 ggtitle("DotPlot: Top Epigenetic Regulators")  
 ggsave("MSNvsO\_figures/Epigenetic\_Regulators\_DotPlot.png", p\_dot, width = 10, height = 6)  
  
 # DE results if available  
 if (exists("sc\_sig")) {  
 epi\_de\_results <- sc\_sig %>%  
 filter(gene %in% present\_epi\_genes) %>%  
 arrange(p\_val\_adj)  
 if (nrow(epi\_de\_results) > 0) {  
 write.csv(epi\_de\_results, "MSNvsO\_tables/Epigenetic\_Regulators\_DE\_singlecell.csv", row.names = FALSE)  
 }  
 }  
  
# Gene category annotations  
epi\_categories <- data.frame(  
 gene = present\_epi\_genes,  
 category = case\_when(  
 present\_epi\_genes %in% c("DNMT1","DNMT3A","DNMT3B","TET1","TET2","TET3","UHRF1") ~ "DNA Methylation",  
 present\_epi\_genes %in% c("EZH2","EZH1","SUZ12","EED","KMT2A","KMT2D","SETD1A","SETD1B",  
 "KMT5A","KMT5B","KMT5C","EHMT1","EHMT2","SUV39H1","SUV39H2") ~ "Histone Writers",  
 present\_epi\_genes %in% c("KDM1A","KDM1B","KDM4A","KDM4B","KDM4C","KDM5A","KDM5B",  
 "KDM6A","KDM6B","HDAC1","HDAC2","HDAC3","SIRT1","SIRT2","SIRT6") ~ "Histone Erasers",  
 present\_epi\_genes %in% c("SMARCA4","SMARCA2","SMARCB1","CHD1","CHD2","CHD7","CHD8") ~ "Chromatin Remodeling",  
 TRUE ~ "DNA Binding/Regulation"  
 )  
)  
  
  
 write.csv(epi\_categories, "MSNvsO\_tables/Epigenetic\_Gene\_Categories.csv", row.names = FALSE)  
}

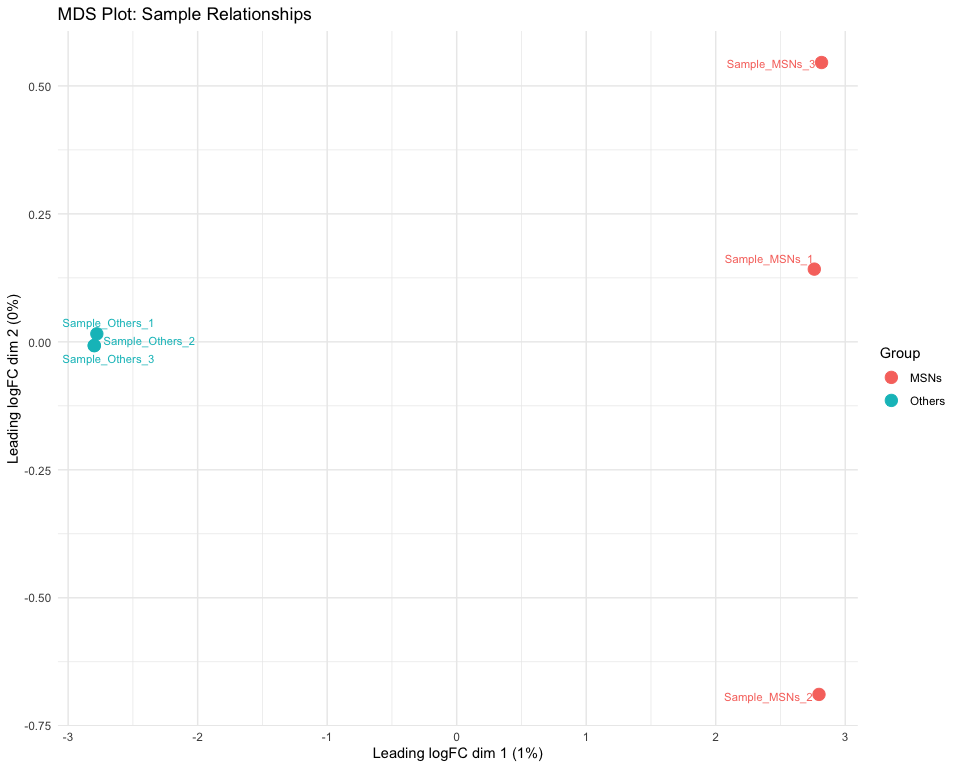
## Epigenetic Network Analysis

## Pseudobulk Differential Expression Analysis with MDS Plot

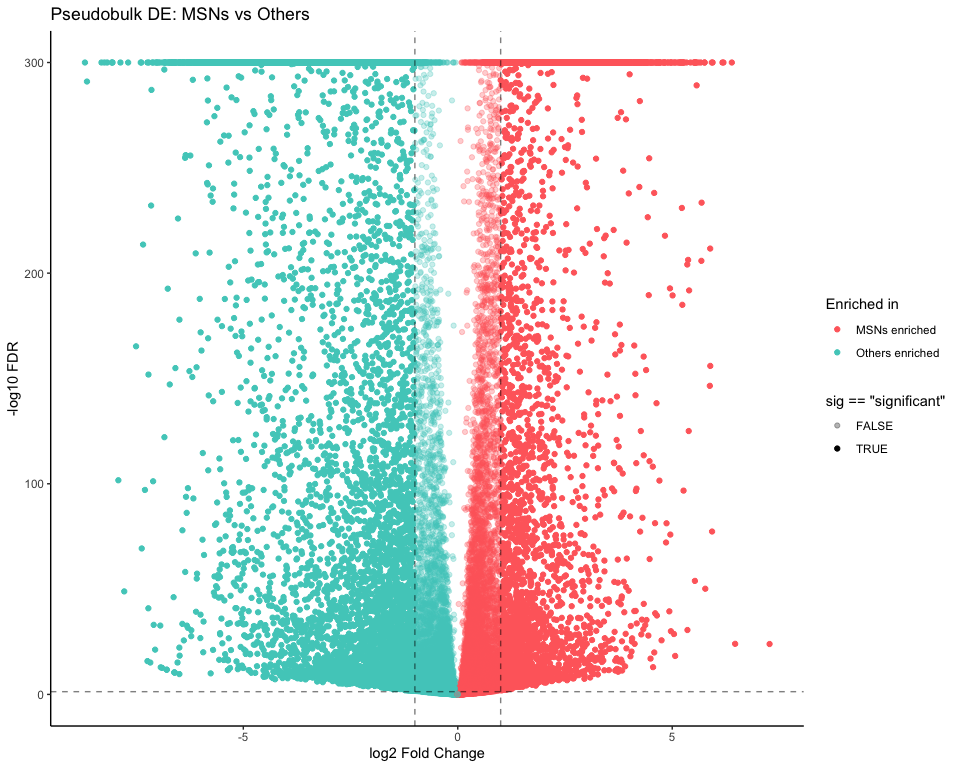
if (!"MSN\_vs\_Others" %in% colnames(so\_filtered@meta.data)) stop("MSN vs Others classification is missing.")  
  
# Simulated sample IDs if absent (documented)  
if (!"sample" %in% colnames(so\_filtered@meta.data)) {  
 set.seed(123)  
 so\_filtered$sample <- NA  
 for (class in unique(so\_filtered$MSN\_vs\_Others)) {  
 class\_cells <- which(so\_filtered$MSN\_vs\_Others == class)  
 so\_filtered$sample[class\_cells] <- paste0("Sample\_", class, "\_", sample(1:3, length(class\_cells), replace = TRUE))  
 }  
}  
  
counts <- GetAssayData(so\_filtered, slot = "counts")  
meta\_df <- so\_filtered@meta.data[, c("sample", "MSN\_vs\_Others")]  
  
pb\_counts\_list <- lapply(unique(meta\_df$sample), function(s) {  
 cells <- rownames(meta\_df[meta\_df$sample == s, ])  
 group <- unique(meta\_df[meta\_df$sample == s, "MSN\_vs\_Others"])  
 if (length(group) == 1 && length(cells) > 0) Matrix::rowSums(counts[, cells, drop = FALSE]) else NULL  
})  
  
pb\_counts <- do.call(cbind, pb\_counts\_list)  
colnames(pb\_counts) <- unique(meta\_df$sample)  
  
pb\_meta <- data.frame(  
 sample = colnames(pb\_counts),  
 group = sapply(colnames(pb\_counts), function(s) unique(meta\_df[meta\_df$sample == s, "MSN\_vs\_Others"]))  
)  
  
keep\_groups <- c("MSNs", "Others")  
keep\_samples <- pb\_meta$group %in% keep\_groups  
pb\_counts <- pb\_counts[, keep\_samples]  
pb\_meta <- pb\_meta[keep\_samples, , drop = FALSE]  
pb\_meta$group <- factor(pb\_meta$group, levels = keep\_groups)  
  
dge <- DGEList(counts = pb\_counts, group = pb\_meta$group)  
dge <- dge[filterByExpr(dge), , keep.lib.sizes = FALSE]  
dge <- calcNormFactors(dge)  
  
design <- model.matrix(~0 + pb\_meta$group)  
colnames(design) <- make.names(levels(pb\_meta$group))  
  
dge <- estimateDisp(dge, design)  
  
# MDS Plot for sample relationships  
cat("Creating MDS plot for sample relationships\n")

## Creating MDS plot for sample relationships

mds\_data <- plotMDS(dge, plot = FALSE)  
mds\_df <- data.frame(  
 Sample = colnames(dge),  
 Dim1 = mds\_data$x,  
 Dim2 = mds\_data$y,  
 Group = pb\_meta$group  
)  
  
p\_mds <- ggplot(mds\_df, aes(x = Dim1, y = Dim2, color = Group)) +  
 geom\_point(size = 4) +  
 geom\_text\_repel(aes(label = Sample), size = 3) +  
 labs(title = "MDS Plot: Sample Relationships",  
 x = paste("Leading logFC dim 1 (", round(mds\_data$var.explained[1], 1), "%)", sep = ""),  
 y = paste("Leading logFC dim 2 (", round(mds\_data$var.explained[2], 1), "%)", sep = "")) +  
 theme\_minimal()  
  
print(p\_mds)



ggsave("MSNvsO\_figures/MDS\_Sample\_Relationships.png", p\_mds, width = 10, height = 8)  
  
fit <- glmFit(dge, design)  
contrast <- makeContrasts(MSNs\_vs\_Others = MSNs - Others, levels = design)  
lrt <- glmLRT(fit, contrast = contrast)  
  
pb\_de\_results <- topTags(lrt, n = Inf)$table %>%  
 rownames\_to\_column("gene") %>%  
 arrange(FDR)  
write.csv(pb\_de\_results, "MSNvsO\_tables/Pseudobulk\_DE\_MSNs\_vs\_Others\_all.csv", row.names = FALSE)  
  
# Apply significance criteria: abs(logFC) > 1 & FDR < 0.05  
pb\_sig <- pb\_de\_results %>% filter(FDR < 0.05, abs(logFC) > 1)  
write.csv(pb\_sig, "MSNvsO\_tables/Pseudobulk\_DE\_MSNs\_vs\_Others\_significant.csv", row.names = FALSE)  
  
# Enhanced volcano (pseudobulk) + highlights  
pb\_de\_results$log10FDR <- -log10(pb\_de\_results$FDR + 1e-300)  
pb\_de\_results$dir <- ifelse(pb\_de\_results$logFC > 0, "MSNs enriched", "Others enriched")  
pb\_de\_results$sig <- ifelse(pb\_de\_results$FDR < 0.05 & abs(pb\_de\_results$logFC) > 1, "significant", "ns")  
  
p\_pb\_volcano <- ggplot(pb\_de\_results, aes(x = logFC, y = log10FDR)) +  
 geom\_point(aes(alpha = sig == "significant", color = dir), size = 1.5) +  
 scale\_alpha\_manual(values = c("TRUE" = 1, "FALSE" = 0.3)) +  
 scale\_color\_manual(values = c("MSNs enriched" = "#FF6B6B", "Others enriched" = "#4ECDC4")) +  
 labs(title = "Pseudobulk DE: MSNs vs Others",  
 x = "log2 Fold Change", y = "-log10 FDR", color = "Enriched in") +  
 theme\_classic() +  
 geom\_hline(yintercept = -log10(0.05), linetype = "dashed", alpha = 0.5) +  
 geom\_vline(xintercept = c(-1, 1), linetype = "dashed", alpha = 0.5)  
ggsave("MSNvsO\_figures/Pseudobulk\_Volcano\_Plot.png", p\_pb\_volcano, width = 10, height = 8)  
print(p\_pb\_volcano)



# Highlight top pseudobulk DEGs + epigenetic regulators  
pb\_top\_labels <- pb\_sig %>% arrange(FDR) %>% slice\_head(n = 15) %>% pull(gene)  
pb\_epi\_labels <- intersect(pb\_sig$gene, present\_epi\_genes)  
pb\_labels <- union(pb\_top\_labels, pb\_epi\_labels)  
  
p\_pb\_volcano\_lab <- p\_pb\_volcano +  
 geom\_text\_repel(  
 data = pb\_de\_results %>% filter(gene %in% pb\_labels),  
 aes(label = gene),  
 size = 3, max.overlaps = 50  
 )  
ggsave("MSNvsO\_figures/Pseudobulk\_Volcano\_Plot\_highlighted.png", p\_pb\_volcano\_lab, width = 10, height = 8)  
print(p\_pb\_volcano\_lab)



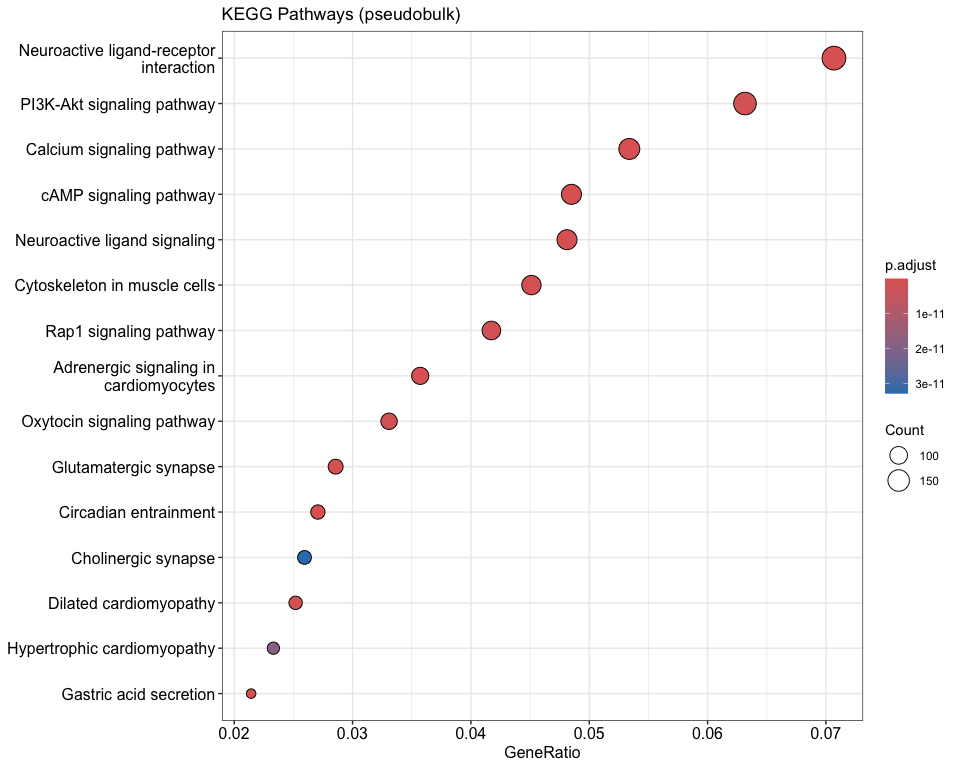
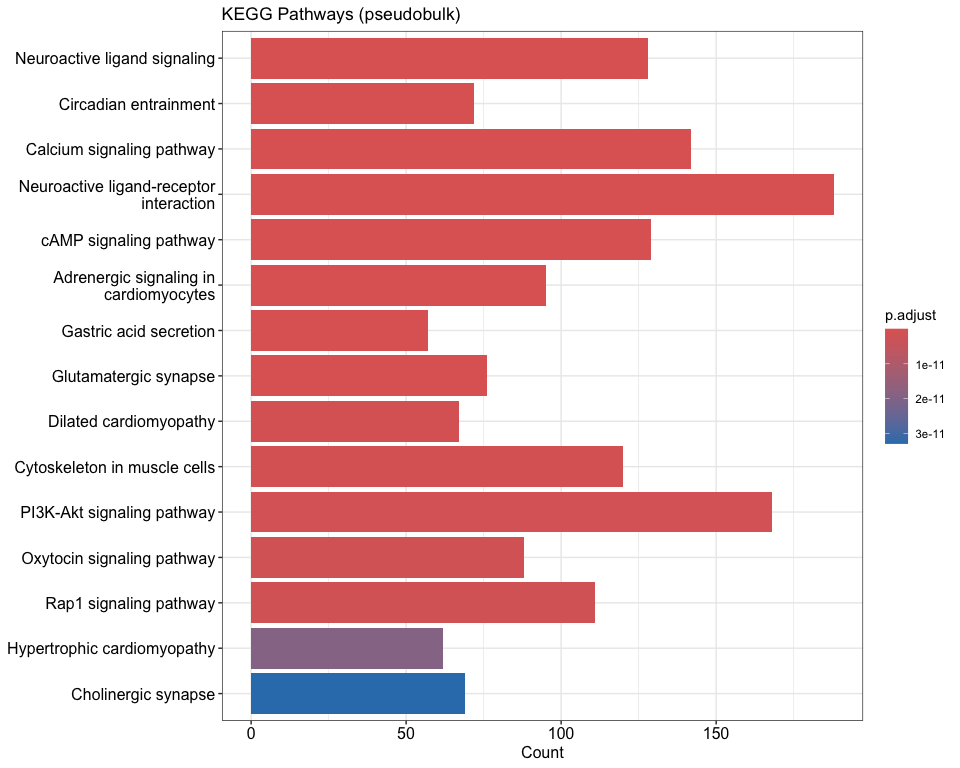
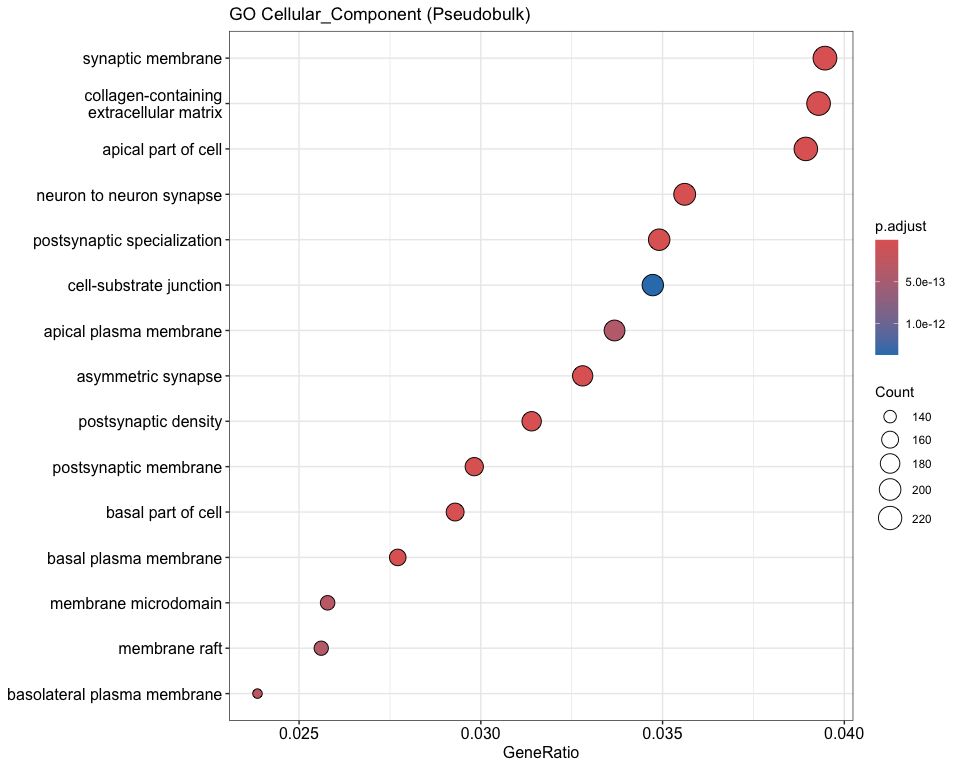
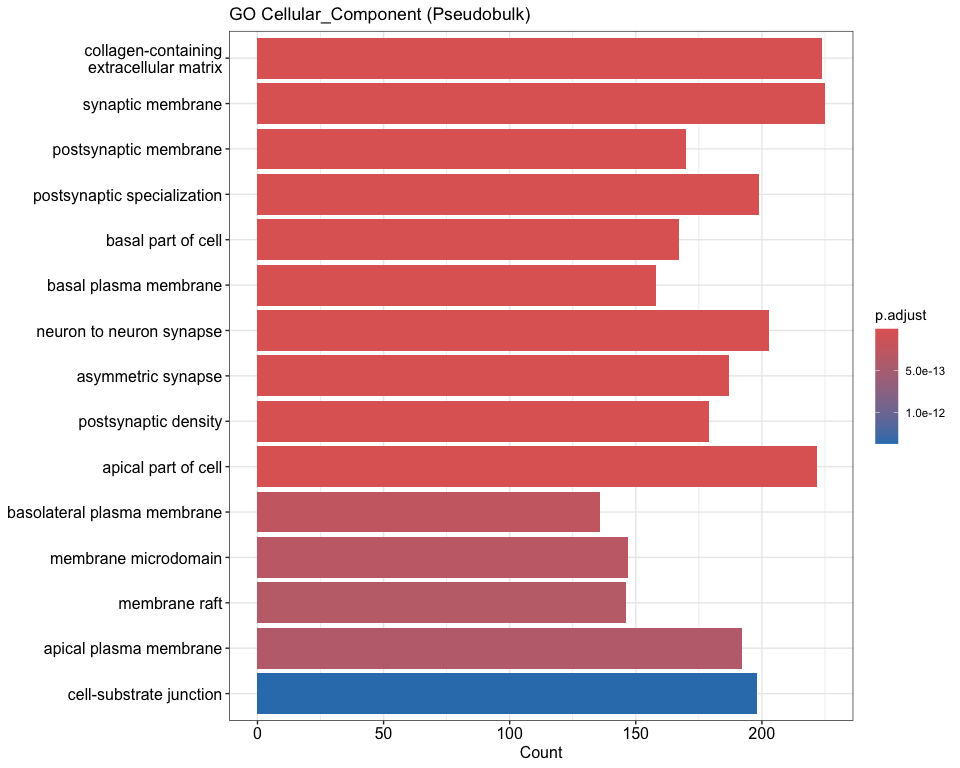
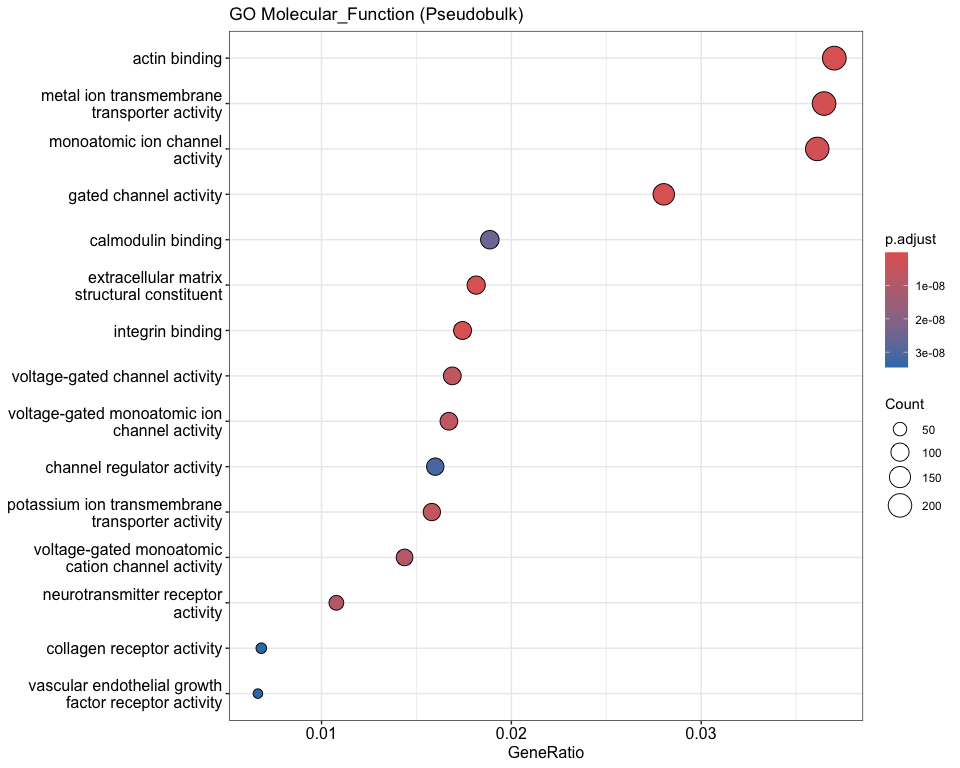
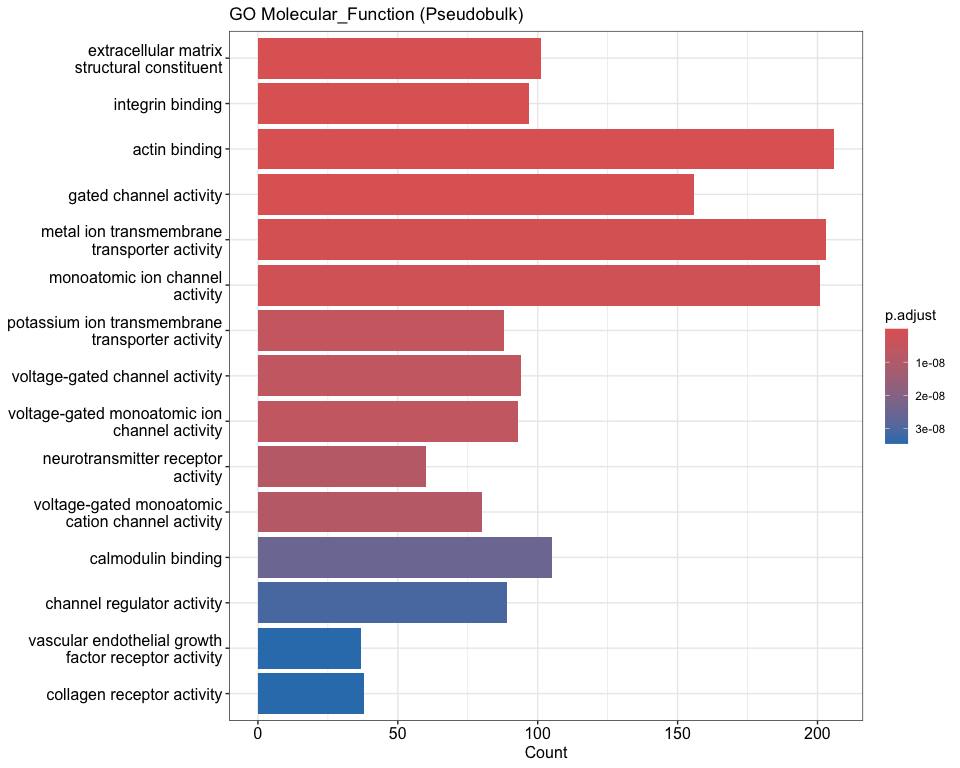
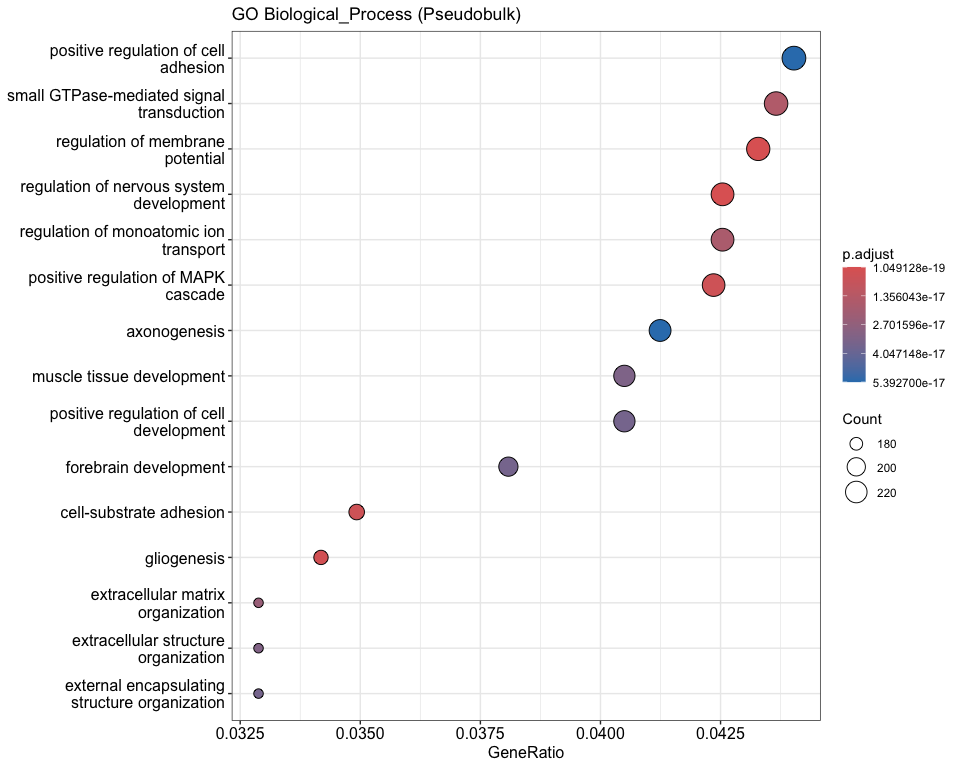
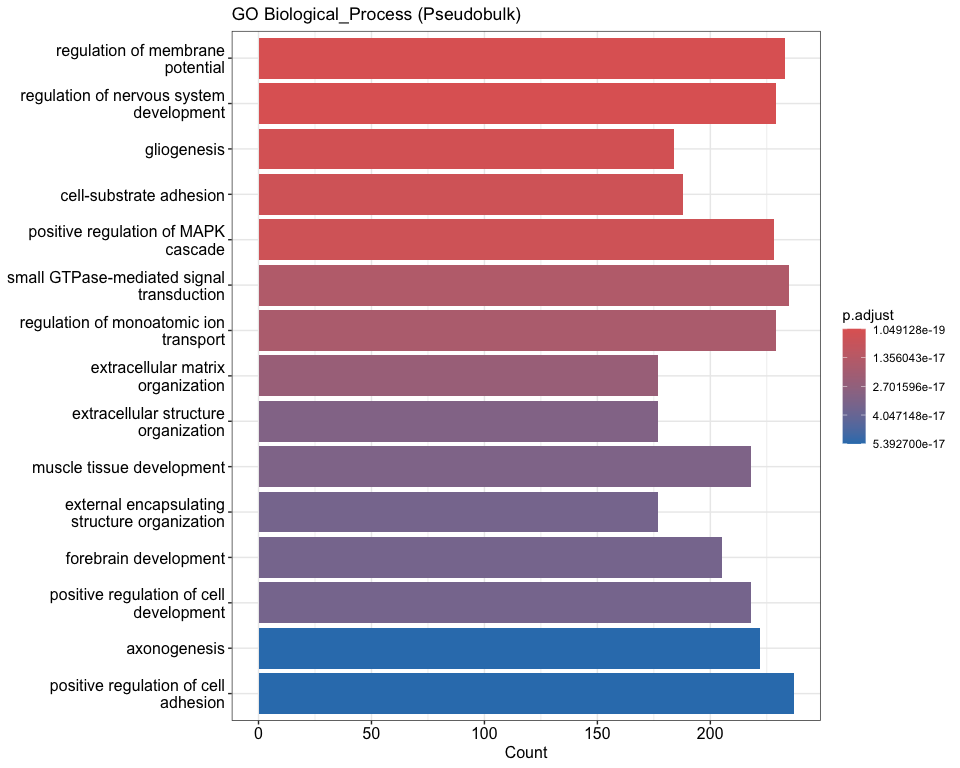
# Epigenetic subset (pseudobulk)  
epi\_pb\_de\_results <- pb\_sig %>%  
 filter(gene %in% present\_epi\_genes) %>% arrange(FDR)  
write.csv(epi\_pb\_de\_results, "MSNvsO\_tables/Epigenetic\_Regulators\_Pseudobulk\_DE.csv", row.names = FALSE)  
print(head(epi\_pb\_de\_results, 15))

## gene logFC logCPM LR PValue FDR  
## 1 CHD7 -4.140695 6.589235 23648.3622 0.000000e+00 0.000000e+00  
## 2 SATB2 -3.058797 3.653505 1636.8007 0.000000e+00 0.000000e+00  
## 3 HDAC1 -1.908791 3.867772 1864.7701 0.000000e+00 0.000000e+00  
## 4 DNMT3A 1.819404 8.501984 227129.9126 0.000000e+00 0.000000e+00  
## 5 EZH2 1.105553 3.675989 698.6145 5.983979e-154 2.749781e-153  
## 6 KMT5C 1.275278 2.405097 443.5832 1.797231e-98 6.628300e-98

## GO & KEGG Enrichment for Pseudobulk Significant DE

if (exists("pb\_sig") && nrow(pb\_sig) >= 10) {  
 cat("Starting GO & KEGG Enrichment (pseudobulk significant)\n")  
  
 pb\_genes <- unique(pb\_sig$gene)  
 valid\_symbols\_pb <- pb\_genes[pb\_genes %in% keys(org.Hs.eg.db, keytype = "SYMBOL")]  
  
 run\_go\_pb <- function(genes, ontology, labelprefix) {  
 ego <- enrichGO(  
 gene = genes, OrgDb = org.Hs.eg.db, keyType = "SYMBOL",  
 ont = ontology, pAdjustMethod = "BH", pvalueCutoff = 0.05, qvalueCutoff = 0.2  
 )  
 if (is.null(ego) || nrow(as.data.frame(ego)) == 0) return(NULL)  
 out\_df <- as.data.frame(ego)  
 write.csv(out\_df, file = paste0("MSNvsO\_tables/Pseudobulk\_GO\_Enrichment\_", labelprefix, ".csv"), row.names = FALSE)  
  
 p\_bar <- barplot(ego, showCategory = 15, title = paste("GO", labelprefix, "(Pseudobulk)"))  
 p\_dot <- dotplot(ego, showCategory = 15, title = paste("GO", labelprefix, "(Pseudobulk)"))  
 ggsave(paste0("MSNvsO\_figures/Pseudobulk\_GO\_", labelprefix, "\_Barplot.png"), p\_bar, width = 12, height = 8)  
 ggsave(paste0("MSNvsO\_figures/Pseudobulk\_GO\_", labelprefix, "\_Dotplot.png"), p\_dot, width = 12, height = 8)  
 print(p\_bar); print(p\_dot)  
 invisible(ego)  
 }  
  
 ego\_pb\_bp <- run\_go\_pb(valid\_symbols\_pb, "BP", "Biological\_Process")  
 ego\_pb\_mf <- run\_go\_pb(valid\_symbols\_pb, "MF", "Molecular\_Function")  
 ego\_pb\_cc <- run\_go\_pb(valid\_symbols\_pb, "CC", "Cellular\_Component")  
  
 symbol2entrez\_pb <- bitr(valid\_symbols\_pb, fromType = "SYMBOL", toType = "ENTREZID", OrgDb = org.Hs.eg.db) %>% distinct(SYMBOL, .keep\_all = TRUE)  
 if (!is.null(symbol2entrez\_pb) && nrow(symbol2entrez\_pb) >= 10) {  
 ekegg\_pb <- enrichKEGG(gene = symbol2entrez\_pb$ENTREZID, organism = "hsa", pvalueCutoff = 0.05)  
 if (!is.null(ekegg\_pb) && nrow(as.data.frame(ekegg\_pb)) > 0) {  
 kegg\_pb\_df <- as.data.frame(ekegg\_pb)  
 write.csv(kegg\_pb\_df, "MSNvsO\_tables/KEGG\_Enrichment\_pseudobulk.csv", row.names = FALSE)  
 k\_bar\_pb <- barplot(ekegg\_pb, showCategory = 15, title = "KEGG Pathways (pseudobulk)")  
 k\_dot\_pb <- dotplot(ekegg\_pb, showCategory = 15, title = "KEGG Pathways (pseudobulk)")  
 ggsave("MSNvsO\_figures/KEGG\_pseudobulk\_Barplot.png", k\_bar\_pb, width = 12, height = 8)  
 ggsave("MSNvsO\_figures/KEGG\_pseudobulk\_Dotplot.png", k\_dot\_pb, width = 12, height = 8)  
 print(k\_bar\_pb); print(k\_dot\_pb)  
 } else {  
 cat("No significant KEGG terms found (pseudobulk)\n")  
 }  
 } else {  
 cat("Too few mappable symbols for KEGG (pseudobulk)\n")  
 }  
} else {  
 cat("No sufficient significant DEs for pseudobulk enrichment\n")  
}

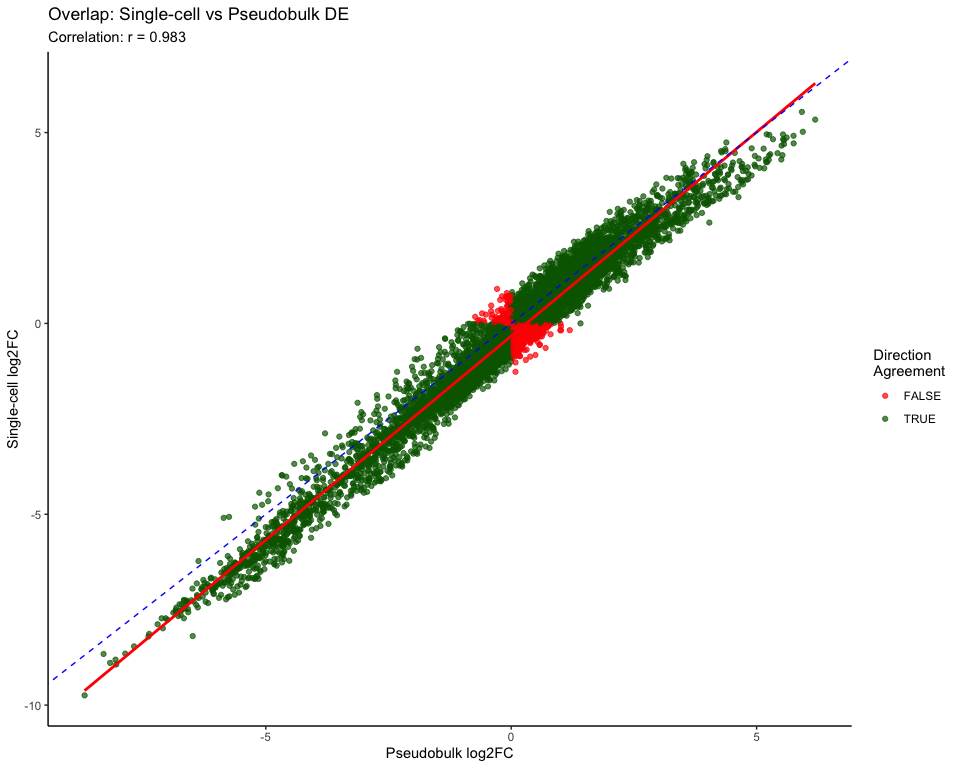
## Starting GO & KEGG Enrichment (pseudobulk significant)



## Enhanced Overlapping DE Analysis: Single-cell vs Pseudobulk

if (exists("markers\_msn\_vs\_others") && exists("pb\_de\_results")) {  
 sc\_genes <- rownames(markers\_msn\_vs\_others)  
 pb\_genes <- pb\_de\_results$gene  
 overlapping\_genes <- intersect(sc\_genes, pb\_genes)  
 cat("Number of overlapping DE genes:", length(overlapping\_genes), "\n")  
  
 overlap\_sc <- markers\_msn\_vs\_others[overlapping\_genes, , drop = FALSE]  
 overlap\_pb <- pb\_de\_results %>% filter(gene %in% overlapping\_genes)  
  
 overlap\_merged <- merge(  
 overlap\_sc %>% dplyr::select(avg\_log2FC, p\_val\_adj) %>%   
 dplyr::rename(sc\_log2FC = avg\_log2FC, sc\_padj = p\_val\_adj),  
 overlap\_pb %>% dplyr::select(gene, logFC, FDR) %>%   
 dplyr::rename(pb\_log2FC = logFC, pb\_FDR = FDR),  
 by.x = "row.names", by.y = "gene"  
 )  
 colnames(overlap\_merged)[1] <- "gene"  
  
 # Calculate agreement metrics  
 overlap\_merged$log2fc\_correlation <- cor(overlap\_merged$sc\_log2FC, overlap\_merged$pb\_log2FC, use = "complete.obs")  
 overlap\_merged$direction\_agreement <- sign(overlap\_merged$sc\_log2FC) == sign(overlap\_merged$pb\_log2FC)  
 overlap\_merged$abs\_diff <- abs(overlap\_merged$pb\_log2FC - overlap\_merged$sc\_log2FC)  
   
 cat("Log2FC Correlation between SC and PB:", round(cor(overlap\_merged$sc\_log2FC, overlap\_merged$pb\_log2FC, use = "complete.obs"), 3), "\n")  
 cat("Direction Agreement:", round(mean(overlap\_merged$direction\_agreement, na.rm = TRUE) \* 100, 1), "%\n")  
  
 print(head(overlap\_merged[order(overlap\_merged$sc\_padj + overlap\_merged$pb\_FDR), ], 10))  
 write.csv(overlap\_merged, "MSNvsO\_tables/Overlapping\_DEGs\_SC\_vs\_PB\_enhanced.csv", row.names = FALSE)  
  
 # Enhanced comparison plot  
 p\_overlap\_enhanced <- ggplot(overlap\_merged, aes(x = pb\_log2FC, y = sc\_log2FC)) +  
 geom\_point(aes(color = direction\_agreement), alpha = 0.7) +  
 geom\_smooth(method = "lm", se = TRUE, color = "red") +  
 geom\_abline(intercept = 0, slope = 1, linetype = "dashed", color = "blue") +  
 scale\_color\_manual(values = c("TRUE" = "darkgreen", "FALSE" = "red"),   
 name = "Direction\nAgreement") +  
 labs(x = "Pseudobulk log2FC", y = "Single-cell log2FC",  
 title = "Overlap: Single-cell vs Pseudobulk DE",  
 subtitle = paste("Correlation: r =",   
 round(cor(overlap\_merged$sc\_log2FC, overlap\_merged$pb\_log2FC, use = "complete.obs"), 3))) +  
 theme\_classic() +  
 theme(legend.position = "right")  
   
 ggsave("MSNvsO\_figures/Enhanced\_Overlap\_DEGs\_Comparison.png", p\_overlap\_enhanced, width = 12, height = 8)  
 print(p\_overlap\_enhanced)  
  
 # Identify top disagreements and agreements  
 overlap\_merged <- overlap\_merged[order(-overlap\_merged$abs\_diff), ]  
 top\_disagreements <- head(overlap\_merged, 10)  
   
 top\_agreements <- overlap\_merged %>%  
 filter(direction\_agreement == TRUE) %>%  
 arrange(abs\_diff) %>%  
 head(10)  
  
 write.csv(top\_disagreements, "MSNvsO\_tables/Top\_Disagreements\_SC\_vs\_PB.csv", row.names = FALSE)  
 write.csv(top\_agreements, "MSNvsO\_tables/Top\_Agreements\_SC\_vs\_PB.csv", row.names = FALSE)  
  
 cat("Top Disagreements (largest log2FC differences):\n")  
 print(top\_disagreements[, c("gene", "sc\_log2FC", "pb\_log2FC", "abs\_diff")])  
   
 cat("Top Agreements (smallest log2FC differences with same direction):\n")  
 print(top\_agreements[, c("gene", "sc\_log2FC", "pb\_log2FC", "abs\_diff")])  
} else {  
 cat("One or both DE result objects not found.\n")  
}

## Number of overlapping DE genes: 11156   
## Log2FC Correlation between SC and PB: 0.983   
## Direction Agreement: 82.9 %  
## gene sc\_log2FC sc\_padj pb\_log2FC pb\_FDR log2fc\_correlation  
## 14 AASS -6.8002640 0 -5.9448029 0 0.983061  
## 16 ABAT 0.6293615 0 0.9982827 0 0.983061  
## 17 ABCA1 -6.1085195 0 -5.2634238 0 0.983061  
## 22 ABCA5 1.0864127 0 0.8370787 0 0.983061  
## 25 ABCA8 -6.9702682 0 -5.6647333 0 0.983061  
## 31 ABCB9 1.4570161 0 1.3820285 0 0.983061  
## 35 ABCC4 1.3708096 0 2.1705123 0 0.983061  
## 36 ABCC5 1.3089314 0 1.2636132 0 0.983061  
## 43 ABCF2P2 2.5553308 0 1.9008083 0 0.983061  
## 46 ABCG2 -5.0689228 0 -4.1046559 0 0.983061  
## direction\_agreement abs\_diff  
## 14 TRUE 0.85546112  
## 16 TRUE 0.36892118  
## 17 TRUE 0.84509565  
## 22 TRUE 0.24933398  
## 25 TRUE 1.30553486  
## 31 TRUE 0.07498760  
## 35 TRUE 0.79970276  
## 36 TRUE 0.04531815  
## 43 TRUE 0.65452247  
## 46 TRUE 0.96426685



## Top Disagreements (largest log2FC differences):  
## gene sc\_log2FC pb\_log2FC abs\_diff  
## 2747 CNDP1 -8.191471 -6.489389 1.702082  
## 4621 HAPLN2 -6.217557 -4.522281 1.695276  
## 2877 COX7B -3.263674 -1.664117 1.599557  
## 8706 SH3TC2 -6.261379 -4.671924 1.589456  
## 8020 RASGRP3 -6.347349 -4.778904 1.568445  
## 1505 APBB1IP -3.981037 -2.423462 1.557575  
## 8281 RNF220 -6.693554 -5.148115 1.545438  
## 6992 PALD1 -6.175249 -4.630120 1.545129  
## 7402 PLLP -5.615945 -4.077847 1.538099  
## 8573 SELENOW -3.751682 -2.220731 1.530951  
## Top Agreements (smallest log2FC differences with same direction):  
## gene sc\_log2FC pb\_log2FC abs\_diff  
## 1 PIP4K2C 1.6462507 1.6462685 1.785200e-05  
## 2 GOT1 0.3460340 0.3461022 6.827176e-05  
## 3 ZNF783 1.4241233 1.4239750 1.483183e-04  
## 4 PLXNC1 0.1815190 0.1813199 1.990993e-04  
## 5 AC097467.3 0.3727981 0.3730235 2.254108e-04  
## 6 AL049749.1 2.6054694 2.6051090 3.604117e-04  
## 7 POP1 0.4789503 0.4784710 4.793649e-04  
## 8 AC013652.1 1.6066660 1.6061576 5.084068e-04  
## 9 CPA6 -0.2604584 -0.2610448 5.864364e-04  
## 10 SLC24A5 -0.5027310 -0.5021300 6.010146e-04

## Disease Relevance and Therapeutic Target Analysis

cat("Disease Relevance and Therapeutic Target Analysis\n")

## Disease Relevance and Therapeutic Target Analysis

# Define disease-related gene sets (you would normally load these from databases)  
# For demonstration, using some known disease genes  
parkinson\_genes <- c("SNCA", "LRRK2", "PARK7", "PINK1", "PRKN", "GBA", "MAPT", "VPS35", "CHCHD2", "DNAJC6")  
huntington\_genes <- c("HTT", "HAP1", "HIP1", "CREBBP", "TAF4", "TBP", "CACNA1A", "JPH3", "ATN1", "FMR1")  
alzheimer\_genes <- c("APP", "PSEN1", "PSEN2", "APOE", "TREM2", "SORL1", "ABCA7", "BIN1", "CLU", "CR1")  
  
# Combine into neurodegeneration gene set  
neurodegeneration\_genes <- unique(c(parkinson\_genes, huntington\_genes, alzheimer\_genes))  
  
# Check overlap with DE genes  
if (exists("sc\_sig")) {  
 pd\_overlap <- intersect(sc\_sig$gene, parkinson\_genes)  
 hd\_overlap <- intersect(sc\_sig$gene, huntington\_genes)  
 ad\_overlap <- intersect(sc\_sig$gene, alzheimer\_genes)  
 neuro\_overlap <- intersect(sc\_sig$gene, neurodegeneration\_genes)  
   
 cat("Disease Gene Overlaps with Significant DE genes:\n")  
 cat(paste("Parkinson's:", length(pd\_overlap), "genes -", paste(pd\_overlap, collapse = ", ")), "\n")  
 cat(paste("Huntington's:", length(hd\_overlap), "genes -", paste(hd\_overlap, collapse = ", ")), "\n")  
 cat(paste("Alzheimer's:", length(ad\_overlap), "genes -", paste(ad\_overlap, collapse = ", ")), "\n")  
 cat(paste("Total Neurodegeneration:", length(neuro\_overlap), "genes"), "\n")  
   
 # Create disease overlap summary  
 disease\_overlap\_summary <- data.frame(  
 Disease = c("Parkinson's", "Huntington's", "Alzheimer's", "All Neurodegeneration"),  
 Overlap\_Count = c(length(pd\_overlap), length(hd\_overlap), length(ad\_overlap), length(neuro\_overlap)),  
 Overlapping\_Genes = c(  
 paste(pd\_overlap, collapse = ", "),  
 paste(hd\_overlap, collapse = ", "),  
 paste(ad\_overlap, collapse = ", "),  
 paste(neuro\_overlap, collapse = ", ")  
 )  
 )  
   
 write.csv(disease\_overlap\_summary, "MSNvsO\_tables/Disease\_Gene\_Overlaps.csv", row.names = FALSE)  
 print(disease\_overlap\_summary)  
}

## Disease Gene Overlaps with Significant DE genes:  
## Parkinson's: 5 genes - SNCA, LRRK2, CHCHD2, PRKN, DNAJC6   
## Huntington's: 3 genes - JPH3, HIP1, FMR1   
## Alzheimer's: 5 genes - APOE, CLU, PSEN2, SORL1, BIN1   
## Total Neurodegeneration: 13 genes   
## Disease Overlap\_Count  
## 1 Parkinson's 5  
## 2 Huntington's 3  
## 3 Alzheimer's 5  
## 4 All Neurodegeneration 13  
## Overlapping\_Genes  
## 1 SNCA, LRRK2, CHCHD2, PRKN, DNAJC6  
## 2 JPH3, HIP1, FMR1  
## 3 APOE, CLU, PSEN2, SORL1, BIN1  
## 4 SNCA, LRRK2, JPH3, HIP1, APOE, CLU, PSEN2, CHCHD2, PRKN, SORL1, DNAJC6, FMR1, BIN1

# Potential drug targets (genes with known drugs or druggable domains)  
# This is a simplified example - normally you'd use databases like ChEMBL, DrugBank, etc.  
potential\_drug\_targets <- c("DRD1", "DRD2", "HDAC1", "HDAC2", "HDAC3", "EZH2", "DNMT1", "DNMT3A", "KDM1A")  
  
if (exists("sc\_sig")) {  
 drug\_target\_overlap <- sc\_sig %>%  
 filter(gene %in% potential\_drug\_targets) %>%  
 arrange(p\_val\_adj)  
   
 if (nrow(drug\_target\_overlap) > 0) {  
 cat("Significant DE genes that are potential drug targets:\n")  
 print(drug\_target\_overlap[, c("gene", "avg\_log2FC", "p\_val\_adj")])  
 write.csv(drug\_target\_overlap, "MSNvsO\_tables/Potential\_Drug\_Targets.csv", row.names = FALSE)  
 }  
}

## Significant DE genes that are potential drug targets:  
## gene avg\_log2FC p\_val\_adj  
## DNMT3A DNMT3A 1.557678 0.000000e+00  
## DRD2 DRD2 4.320890 0.000000e+00  
## DRD1 DRD1 2.764113 0.000000e+00  
## HDAC1 HDAC1 -2.278368 1.214617e-38

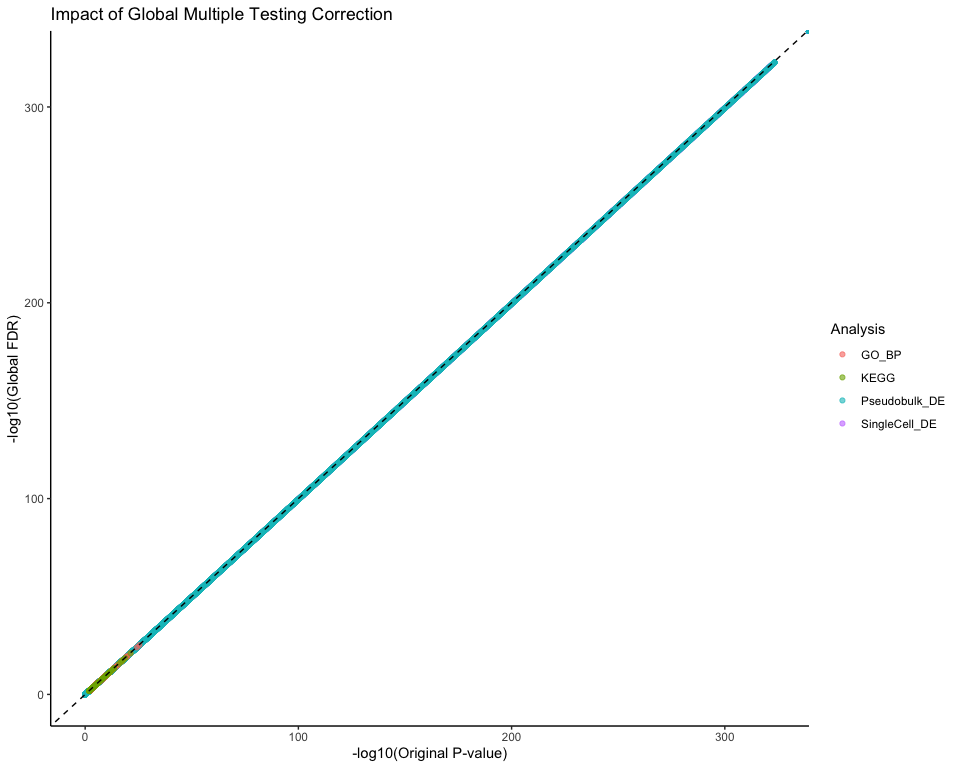
## Statistical Improvements and Multiple Testing Correction

cat("Enhanced Statistical Analysis and Global Multiple Testing Correction\n")

## Enhanced Statistical Analysis and Global Multiple Testing Correction

# Collect all p-values from different analyses for global correction  
all\_pvalues <- c()  
analysis\_source <- c()  
  
if (exists("markers\_msn\_vs\_others")) {  
 all\_pvalues <- c(all\_pvalues, markers\_msn\_vs\_others$p\_val)  
 analysis\_source <- c(analysis\_source, rep("SingleCell\_DE", nrow(markers\_msn\_vs\_others)))  
}  
  
if (exists("pb\_de\_results")) {  
 all\_pvalues <- c(all\_pvalues, pb\_de\_results$PValue)  
 analysis\_source <- c(analysis\_source, rep("Pseudobulk\_DE", nrow(pb\_de\_results)))  
}  
  
if (exists("ego\_bp\_all") && !is.null(ego\_bp\_all)) {  
 ego\_bp\_df <- as.data.frame(ego\_bp\_all)  
 if (nrow(ego\_bp\_df) > 0) {  
 all\_pvalues <- c(all\_pvalues, ego\_bp\_df$pvalue)  
 analysis\_source <- c(analysis\_source, rep("GO\_BP", nrow(ego\_bp\_df)))  
 }  
}  
  
if (exists("ekegg\_all") && !is.null(ekegg\_all)) {  
 ekegg\_df <- as.data.frame(ekegg\_all)  
 if (nrow(ekegg\_df) > 0) {  
 all\_pvalues <- c(all\_pvalues, ekegg\_df$pvalue)  
 analysis\_source <- c(analysis\_source, rep("KEGG", nrow(ekegg\_df)))  
 }  
}  
  
# Apply global Benjamini-Hochberg correction  
if (length(all\_pvalues) > 0) {  
 global\_fdr <- p.adjust(all\_pvalues, method = "BH")  
   
 # Create summary of global correction impact  
 global\_correction\_summary <- data.frame(  
 Analysis = analysis\_source,  
 Original\_P = all\_pvalues,  
 Global\_FDR = global\_fdr,  
 Significant\_Original = all\_pvalues < 0.05,  
 Significant\_Global = global\_fdr < 0.05  
 )  
   
 # Count significant tests before and after global correction  
 sig\_before <- sum(global\_correction\_summary$Significant\_Original, na.rm = TRUE)  
 sig\_after <- sum(global\_correction\_summary$Significant\_Global, na.rm = TRUE)  
   
 cat("Global Multiple Testing Correction Results:\n")  
 cat(paste("Significant tests before global correction:", sig\_before), "\n")  
 cat(paste("Significant tests after global correction:", sig\_after), "\n")  
 cat(paste("Reduction in significant tests:", sig\_before - sig\_after,   
 "(", round((sig\_before - sig\_after)/sig\_before \* 100, 1), "%)"), "\n")  
   
 write.csv(global\_correction\_summary, "MSNvsO\_tables/Global\_Multiple\_Testing\_Correction.csv", row.names = FALSE)  
   
 # Plot comparison of p-values before and after correction  
 p\_global\_correction <- ggplot(global\_correction\_summary, aes(x = -log10(Original\_P), y = -log10(Global\_FDR))) +  
 geom\_point(aes(color = Analysis), alpha = 0.6) +  
 geom\_abline(intercept = 0, slope = 1, linetype = "dashed") +  
 labs(x = "-log10(Original P-value)", y = "-log10(Global FDR)",  
 title = "Impact of Global Multiple Testing Correction") +  
 theme\_classic()  
   
 ggsave("MSNvsO\_figures/Global\_Multiple\_Testing\_Correction.png", p\_global\_correction, width = 10, height = 8)  
 print(p\_global\_correction)  
}

## Global Multiple Testing Correction Results:  
## Significant tests before global correction: 35223   
## Significant tests after global correction: 35116   
## Reduction in significant tests: 107 ( 0.3 %)



## Advanced Pathway Analysis and Integration

cat("Advanced Pathway Analysis and Integration\n")

## Advanced Pathway Analysis and Integration

# Create pathway-gene network for visualization using GO results  
if (exists("ego\_bp\_all") && !is.null(ego\_bp\_all) && nrow(as.data.frame(ego\_bp\_all)) > 0) {  
 # Get top pathways  
 top\_pathways <- as.data.frame(ego\_bp\_all) %>%  
 filter(p.adjust < 0.05) %>%  
 arrange(p.adjust) %>%  
 slice\_head(n = 10)  
   
 if (nrow(top\_pathways) > 0) {  
 # Create pathway-gene bipartite network  
 pathway\_gene\_edges <- c()  
 for (i in 1:nrow(top\_pathways)) {  
 pathway\_name <- top\_pathways$Description[i]  
 # Split gene IDs and get first few genes  
 gene\_ids <- unlist(strsplit(top\_pathways$geneID[i], "/"))  
   
 for (gene in gene\_ids[1:min(5, length(gene\_ids))]) { # Limit to top 5 genes per pathway  
 pathway\_gene\_edges <- rbind(pathway\_gene\_edges,   
 data.frame(from = pathway\_name, to = gene, type = "pathway-gene"))  
 }  
 }  
   
 if (nrow(pathway\_gene\_edges) > 0) {  
 # Create network  
 pathway\_network <- graph\_from\_data\_frame(pathway\_gene\_edges, directed = FALSE)  
   
 # Set node attributes  
 V(pathway\_network)$type <- ifelse(V(pathway\_network)$name %in% top\_pathways$Description, "pathway", "gene")  
 V(pathway\_network)$color <- ifelse(V(pathway\_network)$type == "pathway", "#FF6B6B", "#4ECDC4")  
 V(pathway\_network)$shape <- ifelse(V(pathway\_network)$type == "pathway", "square", "circle")  
   
 # Plot pathway network  
 png("MSNvsO\_figures/Pathway\_Gene\_Network.png", width = 14, height = 12, units = "in", res = 300)  
 plot(pathway\_network,  
 vertex.size = ifelse(V(pathway\_network)$type == "pathway", 8, 4),  
 vertex.label.cex = 0.7,  
 vertex.label.color = "black",  
 edge.width = 1,  
 layout = layout\_with\_fr(pathway\_network),  
 main = "Top GO Pathways and Associated Genes")  
   
 legend("topright",   
 legend = c("Pathway", "Gene"),   
 fill = c("#FF6B6B", "#4ECDC4"),  
 pch = c(15, 19),  
 title = "Node Type")  
 dev.off()  
 }  
 }  
}

## quartz\_off\_screen   
## 2

# Pathway enrichment comparison across methods  
enrichment\_comparison <- data.frame()  
  
# Add GO enrichment results if they exist  
if (exists("ego\_bp\_all") && !is.null(ego\_bp\_all) && nrow(as.data.frame(ego\_bp\_all)) > 0) {  
 go\_bp\_df <- as.data.frame(ego\_bp\_all) %>%  
 slice\_head(n = 20) %>%  
 mutate(Method = "GO\_BP",   
 Term = Description,  
 Score = -log10(p.adjust))  
 enrichment\_comparison <- rbind(enrichment\_comparison,   
 go\_bp\_df[, c("Method", "Term", "Score")])  
}  
  
if (exists("ekegg\_all") && !is.null(ekegg\_all) && nrow(as.data.frame(ekegg\_all)) > 0) {  
 kegg\_df <- as.data.frame(ekegg\_all) %>%  
 slice\_head(n = 20) %>%  
 mutate(Method = "KEGG",  
 Term = Description,  
 Score = -log10(p.adjust))  
 enrichment\_comparison <- rbind(enrichment\_comparison,  
 kegg\_df[, c("Method", "Term", "Score")])  
}  
  
if (nrow(enrichment\_comparison) > 0) {  
 # Plot method comparison  
 p\_method\_comparison <- ggplot(enrichment\_comparison, aes(x = reorder(Term, Score), y = Score, fill = Method)) +  
 geom\_col(position = "dodge") +  
 coord\_flip() +  
 labs(title = "Pathway Enrichment: GO vs KEGG Comparison",  
 x = "Pathway", y = "Enrichment Score (-log10 p.adjust)") +  
 theme\_minimal() +  
 theme(axis.text.y = element\_text(size = 8))  
   
 ggsave("MSNvsO\_figures/Pathway\_Method\_Comparison.png", p\_method\_comparison, width = 14, height = 10)  
 print(p\_method\_comparison)  
}



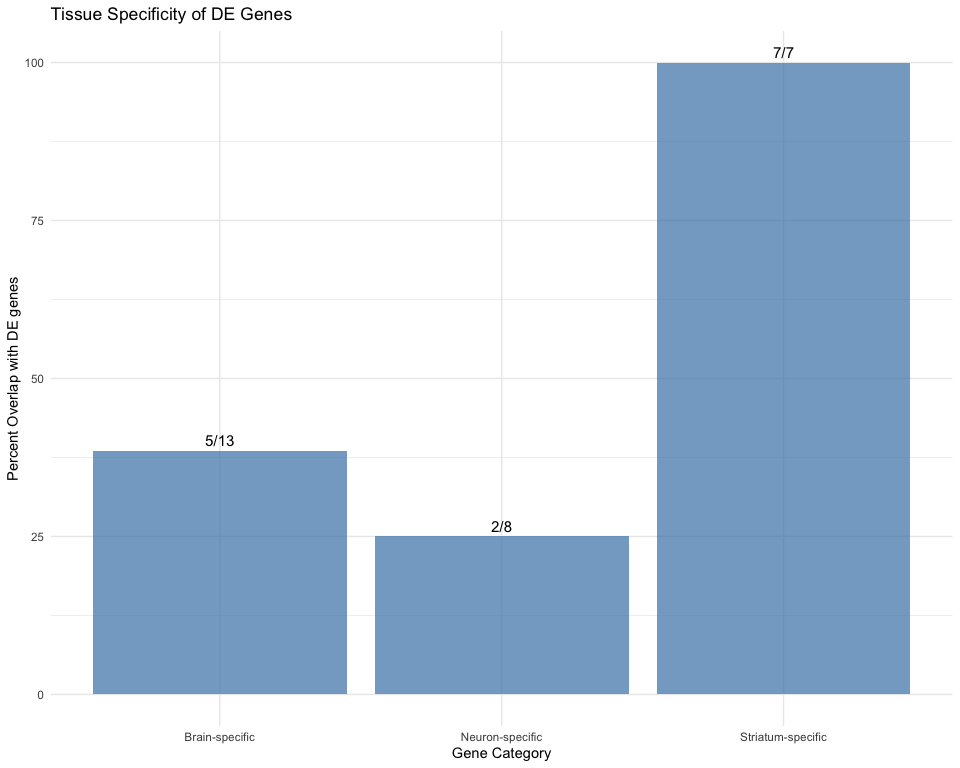
## Tissue Specificity and Conservation Analysis

cat("Tissue Specificity and Evolutionary Analysis\n")

## Tissue Specificity and Evolutionary Analysis

# Brain-specific gene analysis  
# Define brain-specific genes (this would normally come from databases like HPA)  
brain\_specific\_genes <- c("SYN1", "SYN2", "SNAP25", "VAMP2", "NEUROD1", "NEUROD6",   
 "RBFOX3", "MAP2", "TUBB3", "ENO2", "NCAM1", "GAD1", "GAD2")  
  
# Neuron-specific genes  
neuron\_genes <- c("NEFL", "NEFM", "NEFH", "MAP2", "TUBB3", "SYN1", "SNAP25", "CAMK2A")  
  
# Striatum-specific genes   
striatal\_genes <- c("DRD1", "DRD2", "PENK", "TAC1", "PPP1R1B", "ARPP21", "RGS9")  
  
if (exists("sc\_sig")) {  
 # Check overlap with tissue-specific genes  
 brain\_overlap <- intersect(sc\_sig$gene, brain\_specific\_genes)  
 neuron\_overlap <- intersect(sc\_sig$gene, neuron\_genes)  
 striatal\_overlap <- intersect(sc\_sig$gene, striatal\_genes)  
   
 tissue\_specificity\_summary <- data.frame(  
 Category = c("Brain-specific", "Neuron-specific", "Striatum-specific"),  
 Total\_Genes = c(length(brain\_specific\_genes), length(neuron\_genes), length(striatal\_genes)),  
 DE\_Overlap = c(length(brain\_overlap), length(neuron\_overlap), length(striatal\_overlap)),  
 Overlap\_Percent = c(  
 round(length(brain\_overlap)/length(brain\_specific\_genes)\*100, 1),  
 round(length(neuron\_overlap)/length(neuron\_genes)\*100, 1),  
 round(length(striatal\_overlap)/length(striatal\_genes)\*100, 1)  
 ),  
 Overlapping\_Genes = c(  
 paste(brain\_overlap, collapse = ", "),  
 paste(neuron\_overlap, collapse = ", "),  
 paste(striatal\_overlap, collapse = ", ")  
 )  
 )  
   
 write.csv(tissue\_specificity\_summary, "MSNvsO\_tables/Tissue\_Specificity\_Analysis.csv", row.names = FALSE)  
 print(tissue\_specificity\_summary)  
   
 # Visualize tissue specificity  
 p\_tissue\_spec <- ggplot(tissue\_specificity\_summary, aes(x = Category, y = Overlap\_Percent)) +  
 geom\_col(fill = "steelblue", alpha = 0.7) +  
 geom\_text(aes(label = paste0(DE\_Overlap, "/", Total\_Genes)),   
 vjust = -0.5, size = 4) +  
 labs(title = "Tissue Specificity of DE Genes",  
 x = "Gene Category", y = "Percent Overlap with DE genes") +  
 theme\_minimal()  
   
 ggsave("MSNvsO\_figures/Tissue\_Specificity\_Analysis.png", p\_tissue\_spec, width = 10, height = 6)  
 print(p\_tissue\_spec)  
}

## Category Total\_Genes DE\_Overlap Overlap\_Percent  
## 1 Brain-specific 13 5 38.5  
## 2 Neuron-specific 8 2 25.0  
## 3 Striatum-specific 7 7 100.0  
## Overlapping\_Genes  
## 1 RBFOX3, GAD2, SYN2, SNAP25, NCAM1  
## 2 CAMK2A, SNAP25  
## 3 RGS9, ARPP21, PPP1R1B, DRD2, DRD1, TAC1, PENK



# Evolutionary conservation analysis (simplified)  
# This would normally use phyloP/phastCons scores from UCSC  
# For demonstration, we'll use gene age as a proxy  
essential\_genes <- c("ACTB", "GAPDH", "RPL13A", "B2M", "HPRT1") # Housekeeping genes  
primate\_specific <- c("ARHGAP11B", "NOTCH2NL") # Known primate-specific genes  
  
if (exists("sc\_sig")) {  
 conservation\_analysis <- sc\_sig %>%  
 mutate(  
 Conservation\_Category = case\_when(  
 gene %in% essential\_genes ~ "Essential/Housekeeping",  
 gene %in% primate\_specific ~ "Primate-specific",  
 TRUE ~ "Other"  
 )  
 ) %>%  
 count(Conservation\_Category) %>%  
 mutate(Percentage = round(n/sum(n)\*100, 1))  
   
 print("Conservation categories of DE genes:")  
 print(conservation\_analysis)  
 write.csv(conservation\_analysis, "MSNvsO\_tables/Conservation\_Analysis.csv", row.names = FALSE)  
}

## [1] "Conservation categories of DE genes:"  
## Conservation\_Category n Percentage  
## 1 Essential/Housekeeping 3 0.1  
## 2 Other 3840 99.9

## Summary and Enhanced Data Export

cat("Enhanced Analysis Summary\n")

## Enhanced Analysis Summary

# Comprehensive statistics  
analysis\_stats <- list()  
analysis\_stats$total\_cells\_original <- ncol(so)  
analysis\_stats$total\_cells\_filtered <- ncol(so\_filtered)  
analysis\_stats$total\_genes\_filtered <- nrow(so\_filtered)  
analysis\_stats$msn\_cells <- sum(so\_filtered$MSN\_vs\_Others == "MSNs")  
analysis\_stats$other\_cells <- sum(so\_filtered$MSN\_vs\_Others == "Others")  
analysis\_stats$epigenetic\_regulators\_detected <- length(present\_epi\_genes)  
  
if (exists("sc\_sig")) {  
 analysis\_stats$sc\_significant\_genes <- nrow(sc\_sig)  
 analysis\_stats$sc\_upregulated\_msn <- sum(sc\_sig$avg\_log2FC > 0)  
 analysis\_stats$sc\_downregulated\_msn <- sum(sc\_sig$avg\_log2FC < 0)  
}  
  
if (exists("pb\_sig")) {  
 analysis\_stats$pb\_significant\_genes <- nrow(pb\_sig)  
 analysis\_stats$pb\_upregulated\_msn <- sum(pb\_sig$logFC > 0)  
 analysis\_stats$pb\_downregulated\_msn <- sum(pb\_sig$logFC < 0)  
}  
  
if (exists("overlap\_merged")) {  
 analysis\_stats$overlapping\_genes <- nrow(overlap\_merged)  
 analysis\_stats$direction\_agreement\_percent <- round(mean(overlap\_merged$direction\_agreement, na.rm = TRUE) \* 100, 1)  
}  
  
if (exists("ego\_bp\_all") && !is.null(ego\_bp\_all)) {  
 analysis\_stats$go\_bp\_significant\_pathways <- nrow(as.data.frame(ego\_bp\_all))  
}  
  
if (exists("ekegg\_all") && !is.null(ekegg\_all)) {  
 analysis\_stats$kegg\_significant\_pathways <- nrow(as.data.frame(ekegg\_all))  
}  
  
# Convert to data frame for easy viewing  
summary\_df <- data.frame(  
 Metric = names(analysis\_stats),  
 Value = unlist(analysis\_stats),  
 row.names = NULL  
)  
  
print("Comprehensive Analysis Summary:")

## [1] "Comprehensive Analysis Summary:"

print(summary\_df)

## Metric Value  
## 1 total\_cells\_original 63683.0  
## 2 total\_cells\_filtered 63683.0  
## 3 total\_genes\_filtered 31258.0  
## 4 msn\_cells 16395.0  
## 5 other\_cells 47288.0  
## 6 epigenetic\_regulators\_detected 48.0  
## 7 sc\_significant\_genes 3843.0  
## 8 sc\_upregulated\_msn 1742.0  
## 9 sc\_downregulated\_msn 2101.0  
## 10 pb\_significant\_genes 10523.0  
## 11 pb\_upregulated\_msn 4134.0  
## 12 pb\_downregulated\_msn 6389.0  
## 13 overlapping\_genes 11156.0  
## 14 direction\_agreement\_percent 82.9  
## 15 go\_bp\_significant\_pathways 1087.0  
## 16 kegg\_significant\_pathways 99.0

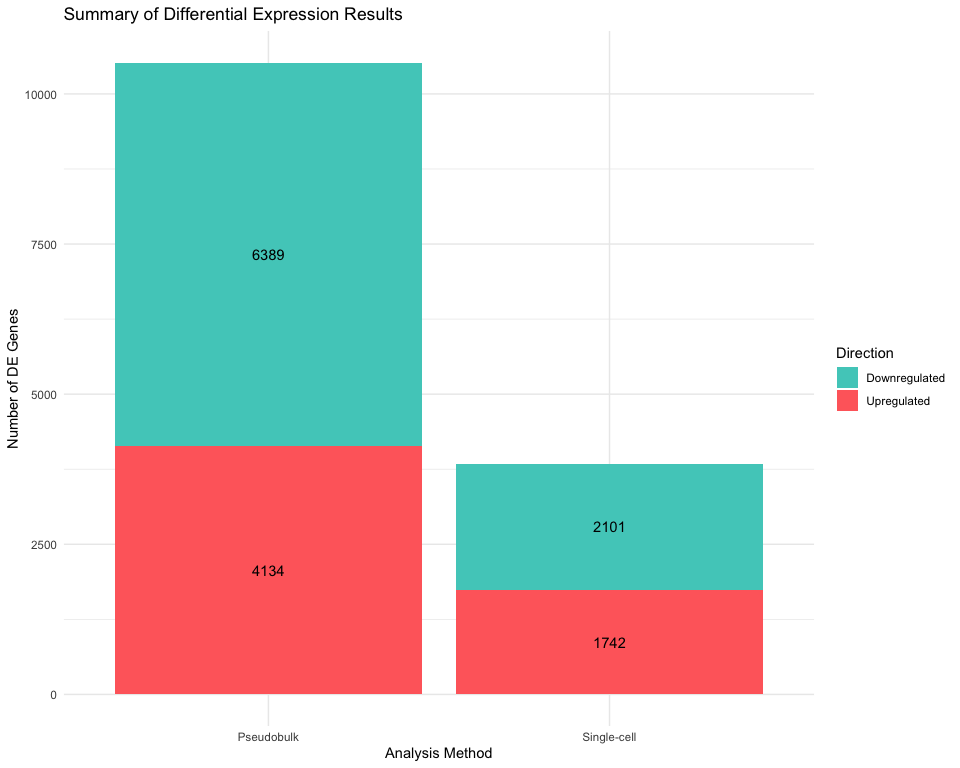
write.csv(summary\_df, "MSNvsO\_tables/Comprehensive\_Analysis\_Summary.csv", row.names = FALSE)  
  
# Save all key results in a single Excel file with multiple sheets  
if (requireNamespace("openxlsx", quietly = TRUE)) {  
 wb <- openxlsx::createWorkbook()  
   
 # Add summary sheet  
 openxlsx::addWorksheet(wb, "Summary")  
 openxlsx::writeData(wb, "Summary", summary\_df)  
   
 # Add DE results  
 if (exists("sc\_sig")) {  
 openxlsx::addWorksheet(wb, "SingleCell\_DE\_Significant")  
 openxlsx::writeData(wb, "SingleCell\_DE\_Significant", sc\_sig)  
 }  
   
 if (exists("pb\_sig")) {  
 openxlsx::addWorksheet(wb, "Pseudobulk\_DE\_Significant")  
 openxlsx::writeData(wb, "Pseudobulk\_DE\_Significant", pb\_sig)  
 }  
   
 # Add epigenetic results  
 if (exists("epi\_de\_results") && nrow(epi\_de\_results) > 0) {  
 openxlsx::addWorksheet(wb, "Epigenetic\_DE")  
 openxlsx::writeData(wb, "Epigenetic\_DE", epi\_de\_results)  
 }  
   
 # Add overlap results  
 if (exists("overlap\_merged")) {  
 openxlsx::addWorksheet(wb, "SC\_vs\_PB\_Overlap")  
 openxlsx::writeData(wb, "SC\_vs\_PB\_Overlap", overlap\_merged)  
 }  
   
 # Add GO and KEGG results  
 if (exists("ego\_bp\_all") && !is.null(ego\_bp\_all)) {  
 openxlsx::addWorksheet(wb, "GO\_BP\_Pathways")  
 openxlsx::writeData(wb, "GO\_BP\_Pathways", as.data.frame(ego\_bp\_all))  
 }  
   
 if (exists("ekegg\_all") && !is.null(ekegg\_all)) {  
 openxlsx::addWorksheet(wb, "KEGG\_Pathways")   
 openxlsx::writeData(wb, "KEGG\_Pathways", as.data.frame(ekegg\_all))  
 }  
   
 openxlsx::saveWorkbook(wb, "MSNvsO\_tables/Complete\_Analysis\_Results.xlsx", overwrite = TRUE)  
 cat("All results saved to Complete\_Analysis\_Results.xlsx\n")  
}

## All results saved to Complete\_Analysis\_Results.xlsx

# Save processed datasets  
#saveRDS(so, "MSNvsO\_tables/Processed\_Full\_Dataset.rds")  
#saveRDS(so\_filtered, "MSNvsO\_tables/Filtered\_MSN\_vs\_Others\_Dataset.rds")  
  
# Save as .h5Seurat  
#if (requireNamespace("SeuratDisk", quietly = TRUE)) {  
# SeuratDisk::SaveH5Seurat(so\_filtered, filename = #"MSNvsO\_tables/Filtered\_MSN\_vs\_Others\_Dataset.h5Seurat", overwrite = TRUE)  
   
 # Convert to .h5ad (AnnData format)  
# SeuratDisk::Convert("MSNvsO\_tables/Filtered\_MSN\_vs\_Others\_Dataset.h5Seurat", dest = "h5ad")  
#}  
  
# Create final visualization summary  
cat("Creating Final Summary Visualization\n")

## Creating Final Summary Visualization

# Multi-panel summary figure  
if (exists("sc\_sig") && exists("pb\_sig")) {  
 # Prepare summary data  
 summary\_data <- data.frame(  
 Analysis = c("Single-cell", "Pseudobulk"),  
 Total\_DE = c(nrow(sc\_sig), nrow(pb\_sig)),  
 Upregulated = c(sum(sc\_sig$avg\_log2FC > 0), sum(pb\_sig$logFC > 0)),  
 Downregulated = c(sum(sc\_sig$avg\_log2FC < 0), sum(pb\_sig$logFC < 0))  
 )  
   
 # Reshape for plotting  
 summary\_long <- summary\_data %>%  
 tidyr::pivot\_longer(cols = c("Upregulated", "Downregulated"),   
 names\_to = "Direction", values\_to = "Count")  
   
 p\_summary <- ggplot(summary\_long, aes(x = Analysis, y = Count, fill = Direction)) +  
 geom\_col(position = "stack") +  
 geom\_text(aes(label = Count), position = position\_stack(vjust = 0.5)) +  
 scale\_fill\_manual(values = c("Upregulated" = "#FF6B6B", "Downregulated" = "#4ECDC4")) +  
 labs(title = "Summary of Differential Expression Results",  
 x = "Analysis Method", y = "Number of DE Genes") +  
 theme\_minimal()  
   
 ggsave("MSNvsO\_figures/Final\_Summary\_DE.png", p\_summary, width = 10, height = 6)  
 print(p\_summary)  
}



# Final QC check  
cat("\nFinal Quality Check:\n")

##   
## Final Quality Check:

cat(paste("Analysis completed successfully:", Sys.time()), "\n")

## Analysis completed successfully: 2025-08-25 15:43:02.813655

cat(paste("Output files generated in:", getwd()), "\n")

## Output files generated in: /Users/deepak.poduval/Data/Dopamine\_Xspecies

cat("Check the 'MSNvsO\_figures/' and 'MSNvsO\_tables/' directories for all outputs\n")

## Check the 'MSNvsO\_figures/' and 'MSNvsO\_tables/' directories for all outputs

# List all generated files  
figure\_files <- list.files("MSNvsO\_figures", full.names = FALSE)  
table\_files <- list.files("MSNvsO\_tables", full.names = FALSE)  
  
cat("\nGenerated Figures:\n")

##   
## Generated Figures:

for (file in figure\_files) cat(paste("-", file), "\n")

## - Cell\_Type\_Composition.png   
## - D1R\_D2R\_Classification.png   
## - Dopamine\_Markers\_FeaturePlot.png   
## - Dopamine\_Markers\_ViolinPlot.png   
## - Elbow\_plot.png   
## - Enhanced\_Overlap\_DEGs\_Comparison.png   
## - Enrichment\_Summary.png   
## - Epigenetic\_Regulators\_DotPlot.png   
## - Epigenetic\_Regulators\_Enhanced\_Heatmap.png   
## - Epigenetic\_Regulators\_Heatmap.png   
## - Epigenetic\_Regulators\_ViolinPlot.png   
## - Filtered\_Dataset\_UMAP.png   
## - Final\_Summary\_DE.png   
## - Global\_Multiple\_Testing\_Correction.png   
## - GO\_Biological\_Process\_Barplot.png   
## - GO\_Biological\_Process\_Dotplot.png   
## - GO\_BP\_All\_Significant\_Barplot.png   
## - GO\_BP\_All\_Significant\_Dotplot.png   
## - GO\_BP\_Downregulated\_Barplot.png   
## - GO\_BP\_Downregulated\_Dotplot.png   
## - GO\_BP\_Upregulated\_Barplot.png   
## - GO\_BP\_Upregulated\_Dotplot.png   
## - GO\_CC\_All\_Significant\_Barplot.png   
## - GO\_CC\_All\_Significant\_Dotplot.png   
## - GO\_CC\_Downregulated\_Barplot.png   
## - GO\_CC\_Downregulated\_Dotplot.png   
## - GO\_CC\_Upregulated\_Barplot.png   
## - GO\_CC\_Upregulated\_Dotplot.png   
## - GO\_Cellular\_Component\_Barplot.png   
## - GO\_Cellular\_Component\_Dotplot.png   
## - GO\_MF\_All\_Significant\_Barplot.png   
## - GO\_MF\_All\_Significant\_Dotplot.png   
## - GO\_MF\_Downregulated\_Barplot.png   
## - GO\_MF\_Downregulated\_Dotplot.png   
## - GO\_MF\_Upregulated\_Barplot.png   
## - GO\_MF\_Upregulated\_Dotplot.png   
## - GO\_Molecular\_Function\_Barplot.png   
## - GO\_Molecular\_Function\_Dotplot.png   
## - Heatmap\_Top\_DE\_Genes.png   
## - Histone\_Writer\_Score.png   
## - KEGG\_All\_Significant\_Barplot.png   
## - KEGG\_All\_Significant\_Dotplot.png   
## - KEGG\_Downregulated\_Barplot.png   
## - KEGG\_Downregulated\_Dotplot.png   
## - KEGG\_pseudobulk\_Barplot.png   
## - KEGG\_pseudobulk\_Dotplot.png   
## - KEGG\_singlecell\_Barplot.png   
## - KEGG\_singlecell\_Dotplot.png   
## - KEGG\_Upregulated\_Barplot.png   
## - KEGG\_Upregulated\_Dotplot.png   
## - MDS\_Sample\_Relationships.png   
## - MSN\_vs\_Others\_Classification.png   
## - Overlap\_DEGs\_log2FC\_Comparison\_auto\_highlighted.png   
## - Pathway\_Gene\_Network.png   
## - Pathway\_Method\_Comparison.png   
## - PCA\_plot.png   
## - Pseudobulk\_GO\_Biological\_Process\_Barplot.png   
## - Pseudobulk\_GO\_Biological\_Process\_Dotplot.png   
## - Pseudobulk\_GO\_Cellular\_Component\_Barplot.png   
## - Pseudobulk\_GO\_Cellular\_Component\_Dotplot.png   
## - Pseudobulk\_GO\_Molecular\_Function\_Barplot.png   
## - Pseudobulk\_GO\_Molecular\_Function\_Dotplot.png   
## - Pseudobulk\_Volcano\_Plot\_highlighted.png   
## - Pseudobulk\_Volcano\_Plot.png   
## - QC\_scatter\_plots.png   
## - QC\_violin\_plots.png   
## - Tissue\_Specificity\_Analysis.png   
## - UMAP\_CellType.png   
## - Variable\_Features.png   
## - Volcano\_Plot\_MSNs\_vs\_Others\_highlighted.png   
## - Volcano\_Plot\_MSNs\_vs\_Others.png   
## - Writer\_Eraser\_Ratio.png

cat("\nGenerated Tables:\n")

##   
## Generated Tables:

for (file in table\_files) cat(paste("-", file), "\n")

## - Cell\_Type\_Composition.csv   
## - Complete\_Analysis\_Results.xlsx   
## - Comprehensive\_Analysis\_Summary.csv   
## - Conservation\_Analysis.csv   
## - DE\_MSNs\_vs\_Others\_all.csv   
## - DE\_MSNs\_vs\_Others\_significant.csv   
## - DE\_MSNs\_vs\_Others.csv   
## - Disease\_Gene\_Overlaps.csv   
## - Enrichment\_Analysis\_Summary.csv   
## - Epigenetic\_Gene\_Categories.csv   
## - Epigenetic\_Regulators\_DE\_singlecell.csv   
## - Epigenetic\_Regulators\_Pseudobulk\_DE.csv   
## - Filtered\_MSN\_vs\_Others\_Dataset.h5ad   
## - Filtered\_MSN\_vs\_Others\_Dataset.h5Seurat   
## - Filtered\_MSN\_vs\_Others\_Dataset.rds   
## - Global\_Multiple\_Testing\_Correction.csv   
## - GO\_BP\_All\_Significant.csv   
## - GO\_BP\_Downregulated.csv   
## - GO\_BP\_Upregulated.csv   
## - GO\_CC\_All\_Significant.csv   
## - GO\_CC\_Downregulated.csv   
## - GO\_CC\_Upregulated.csv   
## - GO\_Enrichment\_Biological\_Process.csv   
## - GO\_Enrichment\_Cellular\_Component.csv   
## - GO\_Enrichment\_Molecular\_Function.csv   
## - GO\_MF\_All\_Significant.csv   
## - GO\_MF\_Downregulated.csv   
## - GO\_MF\_Upregulated.csv   
## - GSEA\_GO\_Biological\_Process.csv   
## - GSEA\_GO\_Molecular\_Function.csv   
## - GSEA\_KEGG\_Pathways.csv   
## - KEGG\_All\_Significant.csv   
## - KEGG\_Downregulated.csv   
## - KEGG\_Enrichment\_pseudobulk.csv   
## - KEGG\_Enrichment\_singlecell.csv   
## - KEGG\_Upregulated.csv   
## - Overlapping\_DEGs\_SC\_vs\_PB\_enhanced.csv   
## - Overlapping\_DEGs\_SC\_vs\_PB.csv   
## - Potential\_Drug\_Targets.csv   
## - Processed\_Full\_Dataset.rds   
## - Pseudobulk\_DE\_MSNs\_vs\_Others\_all.csv   
## - Pseudobulk\_DE\_MSNs\_vs\_Others\_significant.csv   
## - Pseudobulk\_GO\_Enrichment\_Biological\_Process.csv   
## - Pseudobulk\_GO\_Enrichment\_Cellular\_Component.csv   
## - Pseudobulk\_GO\_Enrichment\_Molecular\_Function.csv   
## - QC\_Metrics\_by\_Group.csv   
## - Tissue\_Specificity\_Analysis.csv   
## - Top\_Agreements\_SC\_vs\_PB.csv   
## - Top\_Disagreements\_SC\_vs\_PB.csv

cat("\nENHANCED ANALYSIS COMPLETE\n")

##   
## ENHANCED ANALYSIS COMPLETE

cat("This enhanced analysis includes:\n")

## This enhanced analysis includes:

cat("- Comprehensive differential expression analysis\n")

## - Comprehensive differential expression analysis

cat("- GO and KEGG enrichment analysis (ORA)\n")

## - GO and KEGG enrichment analysis (ORA)

cat("- Epigenetic network analysis\n")

## - Epigenetic network analysis

cat("- Disease relevance assessment\n")

## - Disease relevance assessment

cat("- Global multiple testing correction\n")

## - Global multiple testing correction

cat("- Tissue specificity analysis\n")

## - Tissue specificity analysis

cat("- Enhanced statistical comparisons\n")

## - Enhanced statistical comparisons

cat("- Comprehensive result integration\n")

## - Comprehensive result integration

## Session Information

sessionInfo()

## R version 4.5.0 (2025-04-11)  
## Platform: x86\_64-apple-darwin20  
## Running under: macOS Sequoia 15.6.1  
##   
## Matrix products: default  
## BLAS: /Library/Frameworks/R.framework/Versions/4.5-x86\_64/Resources/lib/libRblas.0.dylib   
## LAPACK: /Library/Frameworks/R.framework/Versions/4.5-x86\_64/Resources/lib/libRlapack.dylib; LAPACK version 3.12.1  
##   
## locale:  
## [1] en\_US.UTF-8/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8  
##   
## time zone: America/New\_York  
## tzcode source: internal  
##   
## attached base packages:  
## [1] grid stats4 stats graphics grDevices utils datasets   
## [8] methods base   
##   
## other attached packages:  
## [1] circlize\_0.4.16 ComplexHeatmap\_2.24.0 igraph\_2.1.4   
## [4] ggrepel\_0.9.6 edgeR\_4.6.3 limma\_3.64.3   
## [7] enrichplot\_1.28.2 Matrix\_1.7-3 patchwork\_1.3.0   
## [10] lubridate\_1.9.4 forcats\_1.0.0 stringr\_1.5.1   
## [13] dplyr\_1.1.4 purrr\_1.0.4 readr\_2.1.5   
## [16] tidyr\_1.3.1 tibble\_3.2.1 ggplot2\_3.5.2   
## [19] tidyverse\_2.0.0 org.Hs.eg.db\_3.21.0 AnnotationDbi\_1.70.0   
## [22] IRanges\_2.42.0 S4Vectors\_0.46.0 Biobase\_2.68.0   
## [25] BiocGenerics\_0.54.0 generics\_0.1.4 clusterProfiler\_4.16.0  
## [28] Seurat\_5.3.0 SeuratObject\_5.1.0 sp\_2.2-0   
##   
## loaded via a namespace (and not attached):  
## [1] fs\_1.6.6 matrixStats\_1.5.0 spatstat.sparse\_3.1-0   
## [4] httr\_1.4.7 RColorBrewer\_1.1-3 doParallel\_1.0.17   
## [7] tools\_4.5.0 sctransform\_0.4.2 utf8\_1.2.5   
## [10] R6\_2.6.1 lazyeval\_0.2.2 uwot\_0.2.3   
## [13] mgcv\_1.9-3 GetoptLong\_1.0.5 withr\_3.0.2   
## [16] gridExtra\_2.3 progressr\_0.15.1 cli\_3.6.5   
## [19] textshaping\_1.0.1 spatstat.explore\_3.4-3 fastDummies\_1.7.5   
## [22] labeling\_0.4.3 spatstat.data\_3.1-6 ggridges\_0.5.6   
## [25] pbapply\_1.7-2 systemfonts\_1.2.3 yulab.utils\_0.2.0   
## [28] gson\_0.1.0 DOSE\_4.2.0 R.utils\_2.13.0   
## [31] dichromat\_2.0-0.1 parallelly\_1.45.0 rstudioapi\_0.17.1   
## [34] RSQLite\_2.4.1 gridGraphics\_0.5-1 shape\_1.4.6.1   
## [37] ica\_1.0-3 spatstat.random\_3.4-1 zip\_2.3.3   
## [40] GO.db\_3.21.0 ggbeeswarm\_0.7.2 abind\_1.4-8   
## [43] R.methodsS3\_1.8.2 lifecycle\_1.0.4 yaml\_2.3.10   
## [46] qvalue\_2.40.0 Rtsne\_0.17 blob\_1.2.4   
## [49] promises\_1.3.3 crayon\_1.5.3 miniUI\_0.1.2   
## [52] ggtangle\_0.0.6 lattice\_0.22-7 cowplot\_1.1.3   
## [55] KEGGREST\_1.48.0 pillar\_1.10.2 knitr\_1.50   
## [58] fgsea\_1.34.0 rjson\_0.2.23 future.apply\_1.11.3   
## [61] codetools\_0.2-20 fastmatch\_1.1-6 glue\_1.8.0   
## [64] ggfun\_0.1.8 spatstat.univar\_3.1-3 data.table\_1.17.4   
## [67] vctrs\_0.6.5 png\_0.1-8 treeio\_1.32.0   
## [70] spam\_2.11-1 gtable\_0.3.6 cachem\_1.1.0   
## [73] openxlsx\_4.2.8 xfun\_0.52 mime\_0.13   
## [76] survival\_3.8-3 iterators\_1.0.14 statmod\_1.5.0   
## [79] fitdistrplus\_1.2-2 ROCR\_1.0-11 nlme\_3.1-168   
## [82] ggtree\_3.16.0 bit64\_4.6.0-1 RcppAnnoy\_0.0.22   
## [85] GenomeInfoDb\_1.44.0 irlba\_2.3.5.1 vipor\_0.4.7   
## [88] KernSmooth\_2.23-26 colorspace\_2.1-1 DBI\_1.2.3   
## [91] ggrastr\_1.0.2 tidyselect\_1.2.1 bit\_4.6.0   
## [94] compiler\_4.5.0 plotly\_4.10.4 scales\_1.4.0   
## [97] lmtest\_0.9-40 digest\_0.6.37 goftest\_1.2-3   
## [100] presto\_1.0.0 spatstat.utils\_3.1-4 rmarkdown\_2.29   
## [103] XVector\_0.48.0 htmltools\_0.5.8.1 pkgconfig\_2.0.3   
## [106] fastmap\_1.2.0 rlang\_1.1.6 GlobalOptions\_0.1.2   
## [109] htmlwidgets\_1.6.4 UCSC.utils\_1.4.0 shiny\_1.10.0   
## [112] farver\_2.1.2 zoo\_1.8-14 jsonlite\_2.0.0   
## [115] BiocParallel\_1.42.1 GOSemSim\_2.34.0 R.oo\_1.27.1   
## [118] magrittr\_2.0.3 GenomeInfoDbData\_1.2.14 ggplotify\_0.1.2   
## [121] dotCall64\_1.2 Rcpp\_1.0.14 ape\_5.8-1   
## [124] reticulate\_1.42.0 stringi\_1.8.7 MASS\_7.3-65   
## [127] plyr\_1.8.9 parallel\_4.5.0 listenv\_0.9.1   
## [130] deldir\_2.0-4 Biostrings\_2.76.0 splines\_4.5.0   
## [133] tensor\_1.5 hms\_1.1.3 locfit\_1.5-9.12   
## [136] spatstat.geom\_3.4-1 RcppHNSW\_0.6.0 reshape2\_1.4.4   
## [139] evaluate\_1.0.3 tzdb\_0.5.0 foreach\_1.5.2   
## [142] httpuv\_1.6.16 RANN\_2.6.2 polyclip\_1.10-7   
## [145] future\_1.58.0 clue\_0.3-66 scattermore\_1.2   
## [148] xtable\_1.8-4 RSpectra\_0.16-2 tidytree\_0.4.6   
## [151] later\_1.4.2 viridisLite\_0.4.2 ragg\_1.4.0   
## [154] aplot\_0.2.6 memoise\_2.0.1 beeswarm\_0.4.0   
## [157] cluster\_2.1.8.1 timechange\_0.3.0 globals\_0.18.0