

AD_Gene_Prediction

2025-05-09

```
# install.packages(c("tidyverse", "randomForest", "xgboost", "h2o", "neuralnet", "caret", "pROC", "clusterProf")  
# if (!requireNamespace("BiocManager", quietly=TRUE)) # install.packages("BiocManager")  
# BiocManager::install(c("GEOquery", "SNPRelate", "org.Hs.eg.db", "AnnotationDbi", "disgenet2r"))
```

1. Load libraries

```
library(tidyverse)  
library(randomForest)  
library(xgboost)  
library(h2o)  
library(neuralnet)  
library(caret)  
library(pROC)  
library(GEOquery)  
library(SNPRelate)  
library(org.Hs.eg.db)  
library(AnnotationDbi)  
library(disgenet2r)  
library(clusterProfiler)  
library(ggplot2)
```

```
# Initialize H2O  
h2o.init(max_mem_size = "4G", nthreads = -1)
```

```
## Connection successful!  
##  
## R is connected to the H2O cluster:  
##   H2O cluster uptime:      10 hours 10 minutes  
##   H2O cluster timezone:    America/Chicago  
##   H2O data parsing timezone: UTC  
##   H2O cluster version:     3.44.0.3  
##   H2O cluster version age:  1 year, 4 months and 19 days  
##   H2O cluster name:        H2O_started_from_R_yamunanunavath_dcw451  
##   H2O cluster total nodes:  1  
##   H2O cluster total memory: 3.76 GB  
##   H2O cluster total cores:  8  
##   H2O cluster allowed cores: 8  
##   H2O cluster healthy:      TRUE  
##   H2O Connection ip:        localhost  
##   H2O Connection port:      54321  
##   H2O Connection proxy:     NA
```

```
##      H2O Internal Security:      FALSE
##      R Version:                  R version 4.4.3 (2025-02-28)
```

Prepare SNP-based Features

```
# 1.1 Read filtered GWAS SNP hits
snps_filt <- read.delim(
  "ad_gwas_snps_filtered.tsv",
  sep      = "\t",
  header   = TRUE,
  stringsAsFactors = FALSE
)

# 1.2 Expand reported_genes to long table
snps_long <- snps_filt %>%
  filter(!is.na(reported_genes), reported_genes != "") %>%
  separate_rows(reported_genes, sep = "\\s*") %>%
  rename(gene = reported_genes) %>%
  select(rsID, chromosome, position, gene, p_value, effect_size)

# 1.3 Summarize per-gene SNP features
gene_snps <- snps_long %>%
  group_by(gene) %>%
  summarise(
    min_p      = min(p_value,      na.rm=TRUE),
    mean_beta  = mean(effect_size, na.rm=TRUE),
    snp_count  = n(),
    top_snp    = rsID[which.min(p_value)],
    .groups    = "drop"
  ) %>%
  arrange(min_p)

cat("Number of genes with SNP features:", nrow(gene_snps), "\n")
```

```
## Number of genes with SNP features: 771
```

```
print(head(gene_snps, 10))
```

```
## # A tibble: 10 x 5
##   gene      min_p mean_beta snp_count top_snp
##   <chr>      <dbl>    <dbl>    <int> <chr>
## 1 NR        1. e-307      1.08      525 rs10119
## 2 APOE       2 e-303      4.23       87 rs429358
## 3 TOMM40    8 e-220      3.42       60 rs157580
## 4 PVRL2     1 e-152      5.56       40 rs6859
## 5 APOC1     8 e-149      2.65       26 rs4420638
## 6 BCAM      2 e-143      4.48       21 rs28399637
## 7 BCL3      9 e-116      5.38       16 rs2075650
## 8 EXOC3L2   9 e-116      7.10        6 rs2075650
## 9 PVR       9 e-116      6.44        7 rs2075650
## 10 APOC1P1  1 e- 97      1.40       11 rs157595
```

Expression Data: GSE33000 (Training)

```
# Download & Extract
gset_list <- getGEO("GSE33000", GSEMatrix=TRUE, AnnotGPL=TRUE)
gset33000 <- gset_list[[1]]
expr_raw <- exprs(gset33000)
pheno <- pData(gset33000)
gpl33000 <- getGEO(annotation(gset33000))

# Collapse Probes → Gene Symbols
# Build probe→Entrez lookup (force probe to character)
plat_tbl <- Table(gpl33000) %>%
  as_tibble() %>%
  transmute(
    probe = as.character(ID),
    entrez = as.character(EntrezGeneID)
  )

expr_entrez <- expr_raw %>%
  as.data.frame() %>%
  rownames_to_column("probe") %>%
  left_join(plat_tbl, by="probe") %>%
  drop_na(entrez) %>%
  pivot_longer(-c(probe, entrez), names_to="sample", values_to="expr") %>%
  group_by(entrez, sample) %>%
  summarise(expr = median(expr), .groups="drop") %>%
  pivot_wider(names_from=sample, values_from=expr)

# 5.3 Map Entrez → SYMBOL and collapse duplicates by median
library(dplyr)
library(tibble)

# Map Entrez IDs to gene symbols
symbols <- mapIds(
  org.Hs.eg.db,
  keys      = expr_entrez$entrez,      # use the existing column
  column    = "SYMBOL",
  keytype   = "ENTREZID",
  multiVals = "first"
)

expr_gene <- expr_entrez %>%
  # add symbol column, drop unmapped
  mutate(symbol = symbols[entrez]) %>%
  filter(!is.na(symbol)) %>%
  # now collapse any duplicate symbols by taking median across samples
  group_by(symbol) %>%
  summarise(across(-entrez, median, na.rm=TRUE), .groups="drop") %>%
  # move symbol into rownames
  column_to_rownames("symbol")

cat("Gene-level expression dims:", dim(expr_gene), "\n")
```

```
## Gene-level expression dims: 19867 624
```

```
# Compute Differential Expression (GSE33000)

labels      <- sub(
  "^disease status: ?",
  "",
  pheno$disease status:ch2`,
  ignore.case = TRUE
)
ad_samps    <- labels == "Alzheimer's disease"
control_samps <- labels == "non-demented"

de_stats <- tibble(
  gene      = rownames(expr_gene),
  log2FC    = rowMeans(expr_gene[, ad_samps, drop = FALSE]) -
    rowMeans(expr_gene[, control_samps, drop = FALSE]),
  de_p      = apply(expr_gene, 1, function(x)
    t.test(x[ad_samps], x[control_samps])$p.value)
)
cat("Computed DE for", nrow(de_stats), "genes\n")
```

```
## Computed DE for 19867 genes
```

Load Known AD Genes & Build Feature Table

```
diseases <- read.delim("DISEASES_Summary_GDA_RGD_HUMAN_C0002395.tsv",
  sep="\t", header=TRUE, stringsAsFactors=FALSE)

alz_genes <- unique(diseases$Gene)

features <- gene_snps %>%
  inner_join(de_stats, by="gene") %>%
  mutate(
    mean_beta = if_else(is.na(mean_beta), 0, mean_beta),
    label      = factor(if_else(gene %in% alz_genes, "Known", "Novel"),
      levels=c("Known", "Novel"))
  ) %>%
  drop_na(min_p, mean_beta, snp_count, log2FC, de_p)

table(features$label)

##
## Known Novel
##    14    567
```

Train/Test Split & Balance

```
set.seed(42)
df_bal      <- downSample(x=features %>% select(-label),
                          y=features$label, yname="label")
train_idx   <- createDataPartition(df_bal$label, p=0.8, list=FALSE)
train_df    <- df_bal[train_idx, ]
test_df     <- df_bal[-train_idx, ]
```

Baseline ML: RF & XGB

```
ctrl <- trainControl(method="cv", number=5, classProbs=TRUE, summaryFunction=twoClassSummary)

# Random Forest
rf_cv <- train(label ~ min_p+mean_beta+snp_count+log2FC+de_p,
               data=train_df, method="rf", metric="ROC",
               tuneGrid=data.frame(mtry=2:4), trControl=ctrl)
rf_auc <- roc(test_df$label, predict(rf_cv, test_df, type="prob"))$auc
cat("RF Test AUC:", round(rf_auc,3), "\n")
```

```
## RF Test AUC: 1
```

```
# XGBoost
xgb_grid <- expand.grid(nrounds=c(50,100), max_depth=c(2,4,6),
                        eta=c(0.01,0.1), gamma=0, colsample_bytree=1,
                        min_child_weight=1, subsample=1)
xgb_cv <- train(label~min_p+mean_beta+snp_count+log2FC+de_p,
                 data=train_df, method="xgbTree", metric="ROC",
                 tuneGrid=xgb_grid, trControl=ctrl)
```

[illegible]


```
# Baseline DNN
```

```
hf_train <- as.h2o(train_df)
```

```
## |
```

```
hf_valid <- as.h2o(test_df)
```

```
## |
```

```
y <- "label"; x_vars <- setdiff(names(train_df), y)
```

```
dl_model <- h2o.deeplearning(x=x_vars, y=y,  
  training_frame=hf_train, validation_frame=hf_valid,  
  hidden=c(64,32), activation="RectifierWithDropout",  
  input_dropout_ratio=0.2, hidden_dropout_ratios=c(0.4,0.3),  
  epochs=50, stopping_metric="AUC", stopping_rounds=5,  
  stopping_tolerance=1e-3, seed=42  
)
```

```
## |
```

```
cat("DL Val AUC:", round(h2o.auc(h2o.performance(dl_model,hf_valid)),3), "\n")
```

```
## DL Val AUC: 1
```

```
# Hyperparameter Grid
```

```
hyper_params <- list(  
  hidden          = list(c(32,16), c(16,8), c(8,4)),  
  input_dropout_ratio = c(0,0.1),  
  hidden_dropout_ratios = list(c(0.1,0.1), c(0.2,0.2)),  
  rate            = c(0.001,0.01)  
)
```

```
grid <- h2o.grid("deeplearning", grid_id="dl_grid",  
  x=x_vars, y=y, training_frame=hf_train, validation_frame=hf_valid,  
  distribution="bernoulli", activation="RectifierWithDropout",  
  hyper_params=hyper_params, stopping_metric="AUC",  
  stopping_rounds=5, stopping_tolerance=1e-3, seed=42  
)
```

```
## |
```

```
perf_grid <- h2o.getGrid("dl_grid", sort_by="auc", decreasing=TRUE)  
best_model <- h2o.getModel(perf_grid@model_ids[[1]])  
cat("Best DL Val AUC:", round(h2o.auc(h2o.performance(best_model,hf_valid)),3), "\n")
```

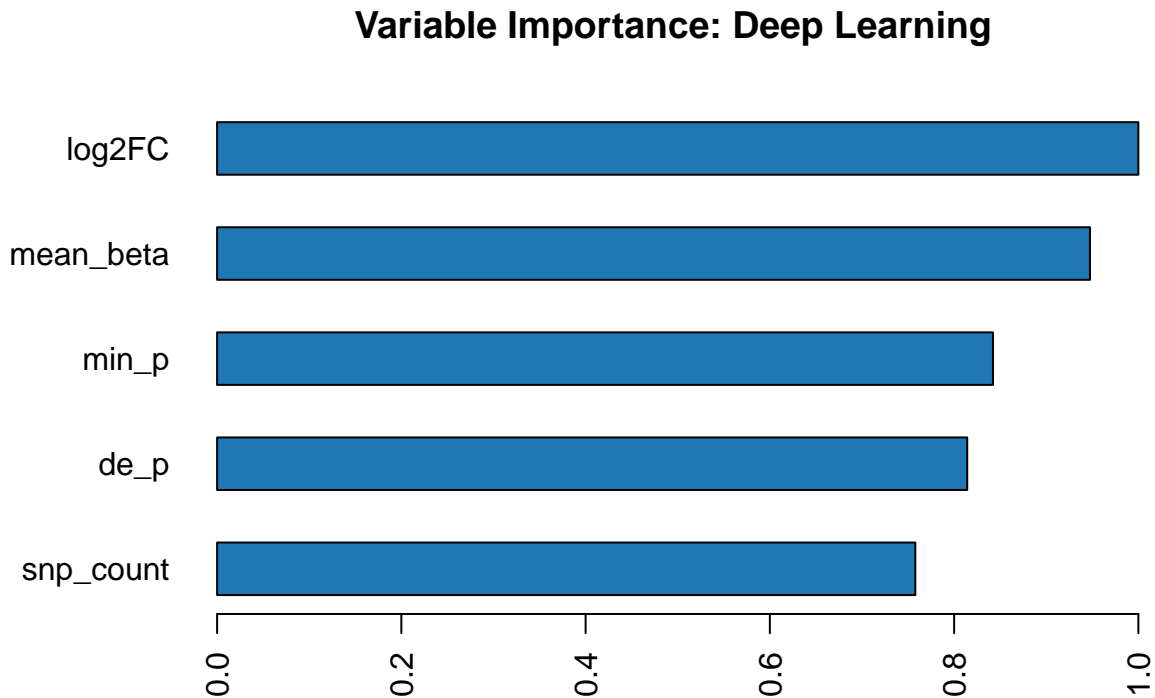
```
## Best DL Val AUC: 1
```

Interpretation & Save

```
# Variable Importance
varimp_df <- as.data.frame(h2o.varimp(best_model))
print(varimp_df)
```

```
##   variable relative_importance scaled_importance percentage
## 1  log2FC           1.0000000         1.0000000  0.2292617
## 2 mean_beta        0.9474829         0.9474829  0.2172216
## 3   min_p          0.8421984         0.8421984  0.1930839
## 4    de_p          0.8142172         0.8142172  0.1866688
## 5 snp_count        0.7579283         0.7579283  0.1737640
```

```
h2o.varimp_plot(best_model, num_of_features=10)
```



```
# Save objects for validation
save(gene_snps, alz_genes, best_model, file="pipeline_objects.RData")
```

Independent Validation: GSE48350


```

# 1 Download & platform
gset2      <- getGEO("GSE48350", GSEMatrix=TRUE, AnnotGPL=TRUE)[[1]]
expr2_raw  <- exprs(gset2); pheno2 <- pData(gset2)
gp12       <- getGEO(annotation(gset2)); plat2_df <- as.data.frame(Table(gp12))

# 2 Collapse probes to SYMBOL
plat2_tbl <- plat2_df %>%
  transmute(
    probe = as.character(ID),
    entrez = as.character(ENTREZ_GENE_ID)
  )

expr2_entrez <- expr2_raw %>%
  as.data.frame() %>% rownames_to_column("probe") %>%
  left_join(plat2_tbl, by="probe") %>% drop_na(entrez) %>%
  pivot_longer(-c(probe,entrez), names_to="sample", values_to="expr") %>%
  group_by(entrez,sample) %>% summarise(expr=median(expr),.groups="drop") %>%
  pivot_wider(names_from=sample, values_from=expr)

symbols2    <- mapIds(org.Hs.eg.db, keys=expr2_entrez$entrez,
                      column="SYMBOL", keytype="ENTREZID", multiVals="first")

expr2_gene <- expr2_entrez %>%
  pivot_longer(-entrez, names_to="sample", values_to="expr") %>%
  mutate(symbol=symbols2[entrez]) %>%
  filter(!is.na(symbol)) %>%
  group_by(symbol,sample) %>%
  summarise(expr=median(expr),.groups="drop") %>%
  pivot_wider(names_from=sample, values_from=expr) %>%
  column_to_rownames("symbol")

# 3 DE stats
labels2 <- sub(".*\\s*", "", pheno2$characteristics_ch1)
ad2     <- labels2=="AA"; ctrl2 <- labels2=="C"

de_stats2 <- map_dfr(rownames(expr2_gene), function(g) {
  adv <- as.numeric(expr2_gene[g,ad2]); cv <- as.numeric(expr2_gene[g,ctrl2])
  if(sum(!is.na(adv))<2||sum(!is.na(cv))<2) return(NULL)
  tibble(gene=g, log2FC=mean(adv,na.rm=TRUE)-mean(cv,na.rm=TRUE), de_p=t.test(adv,cv)$p.value)
})

# 4 Build & score
load("pipeline_objects.RData") # loads gene_snps, alz_genes, best_model

features2 <- gene_snps %>%
  inner_join(de_stats2, by="gene") %>%
  mutate(mean_beta=coalesce(mean_beta,0),
         label=factor(if_else(gene%in%alz_genes,"Known","Novel"),c("Known","Novel"))) %>%
  drop_na(min_p,mean_beta,snp_count,log2FC,de_p)

hf_new <- as.h2o(features2)

```

|

|

```
perf_new <- h2o.performance(best_model, newdata=hf_new)
auc_new <- h2o.auc(perf_new)
cat("Independent Test AUC:", round(auc_new,3), "\n")
```

```
## Independent Test AUC: 0.706
```

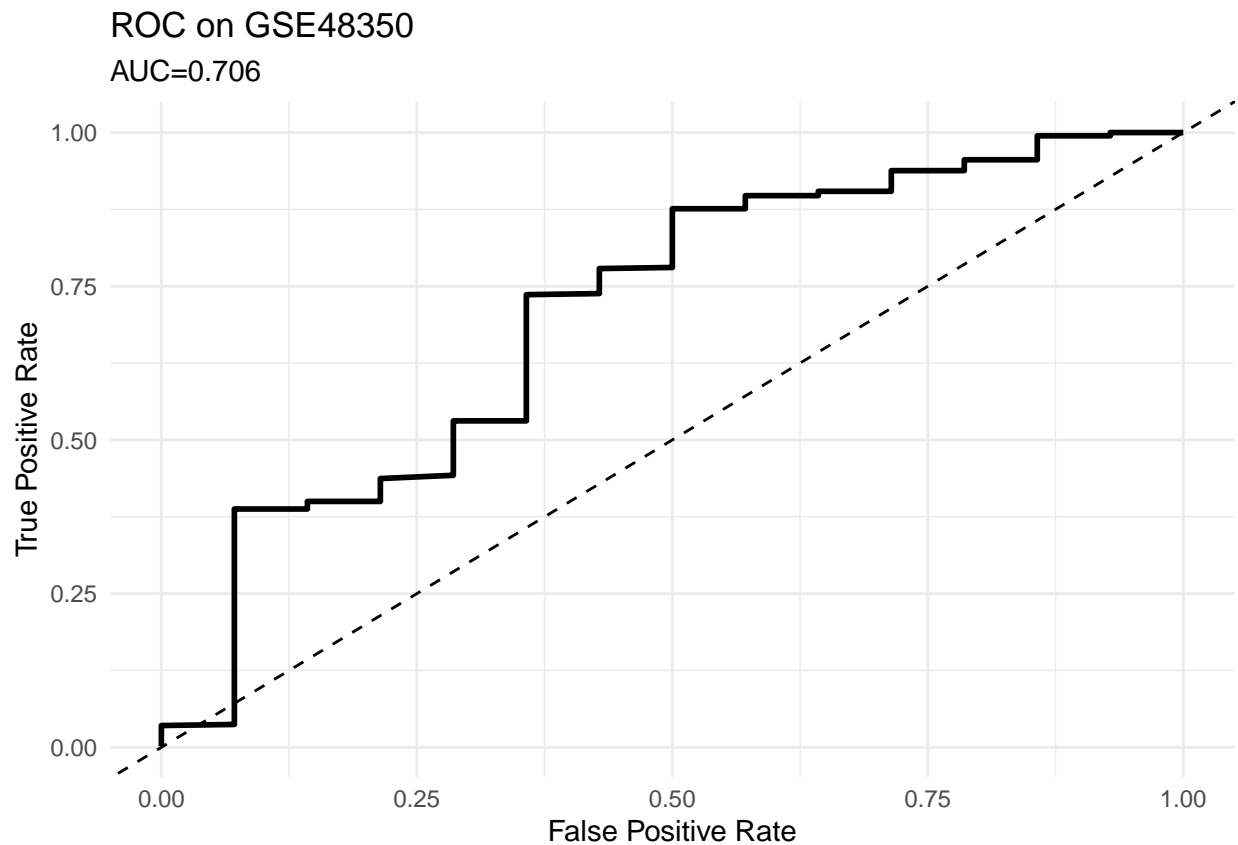
Plots

```
metrics_df <- as.data.frame(h2o.metric(perf_new))
roc_df <- metrics_df %>% select(fpr, tpr)
pr_df <- metrics_df %>% select(recall, precision)
probs_new <- as.data.frame(h2o.predict(best_model, hf_new))
```

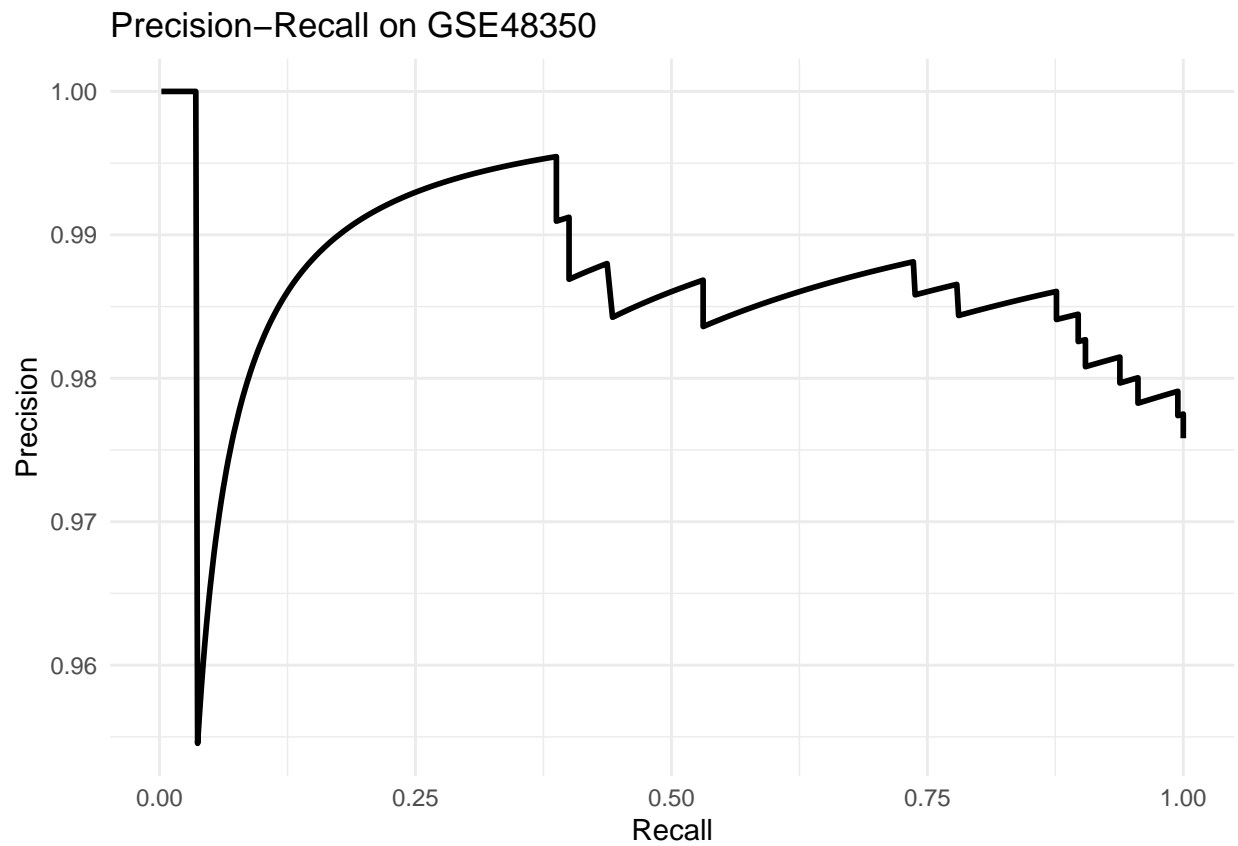
```
## |
```

```
features2 <- features2 %>% mutate(score = probs_new$Known)
```

```
# ROC
ggplot(roc_df, aes(fpr, tpr)) +
  geom_line(size=1) + geom_abline(linetype="dashed") +
  labs(title="ROC on GSE48350", subtitle=paste0("AUC=",round(auc_new,3)),
       x="False Positive Rate", y="True Positive Rate") +
  theme_minimal()
```



```
# PR
ggplot(pr_df, aes(recall, precision)) +
  geom_line(size=1) +
  labs(title="Precision-Recall on GSE48350", x="Recall", y="Precision") +
  theme_minimal()
```

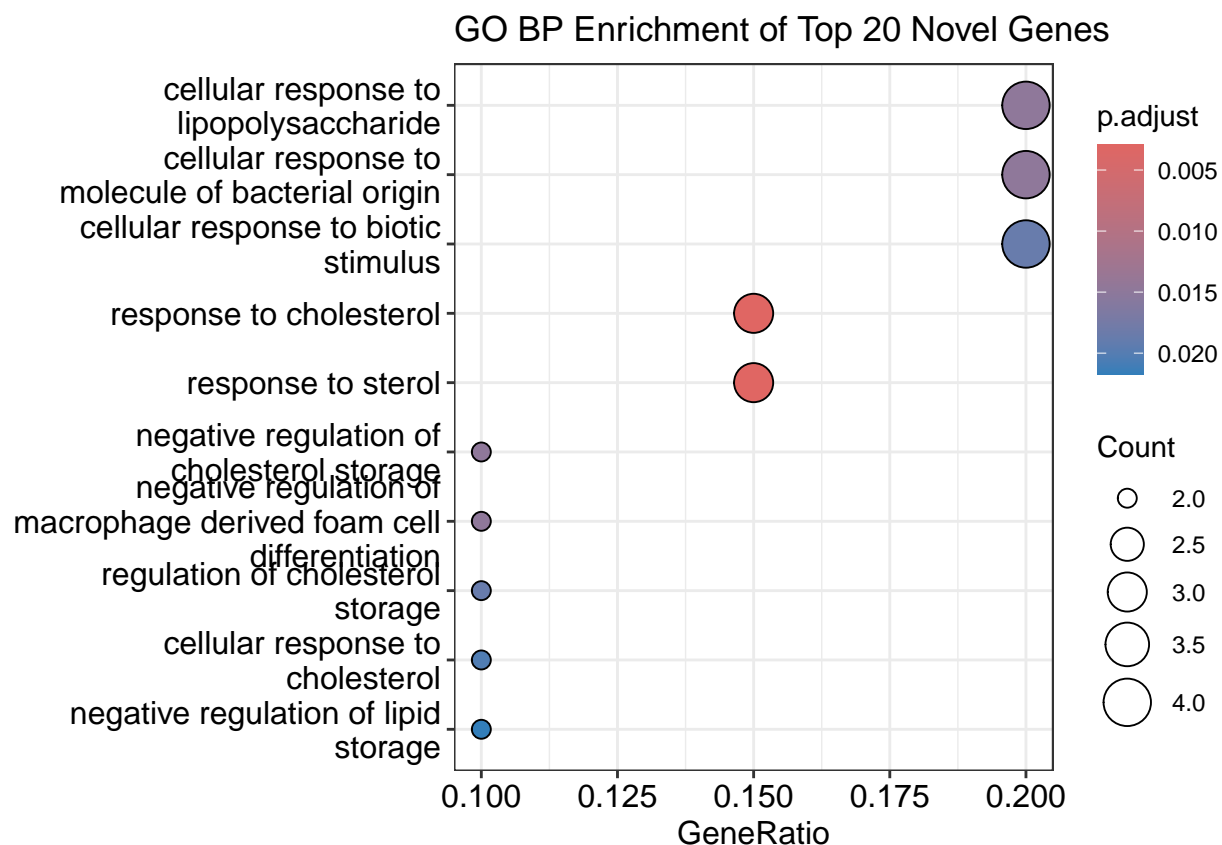


```
# Density
ggplot(features2, aes(score, fill=label)) +
  geom_density(alpha=0.5) +
  labs(title="Predicted P(Known) Density", x="Score", y="Density") +
  theme_minimal()
```



GO Enrichment & Top Novel Hits

```
ranked_novel <- features2 %>%  
  filter(label=="Novel") %>%  
  arrange(desc(score)) %>%  
  slice_head(n=20)  
  
ego <- enrichGO(gene=ranked_novel$gene, OrgDb=org.Hs.eg.db,  
  keyType="SYMBOL", ont="BP",  
  pAdjustMethod="BH", pvalueCutoff=0.05)  
  
dotplot(ego, showCategory=10) +  
  labs(title="GO BP Enrichment of Top 20 Novel Genes")
```



```
print(ranked_novel$gene)
```

```
## [1] "AGBL2" "CCR5" "PACIN3" "SPI1" "GLIS3" "ABCA1"
## [7] "SDF2L1" "NR1H3" "PPP1R13L" "ARRDC4" "CREB3L1" "DGKZ"
## [13] "LILRB2" "NR2F2" "OSBPL7" "RRAS2" "EGR1" "ERCC1"
## [19] "CCRL2" "SMOX"
```

Statistical Rigor: AUC Confidence Intervals, Calibration & Permutation Test

```
library(pROC)
library(dplyr)
library(ggplot2)

# 1 Extract predictions and true labels
pred_probs <- as.data.frame(h2o.predict(best_model, hf_new))$Known
```

```
## |
```

```
true_labels <- factor(features2$label, levels = c("Novel", "Known"))
```

```

# 2 ROC object and 95% CI for AUC
roc_obj <- roc(true_labels, pred_probs, levels = c("Novel","Known"), direction = "<")
ci_auc <- ci_auc(roc_obj)
cat("AUC (95% CI):",
    round(ci_auc[1],3), "to", round(ci_auc[3],3), "\n")

```

```
## AUC (95% CI): 0.549 to 0.862
```

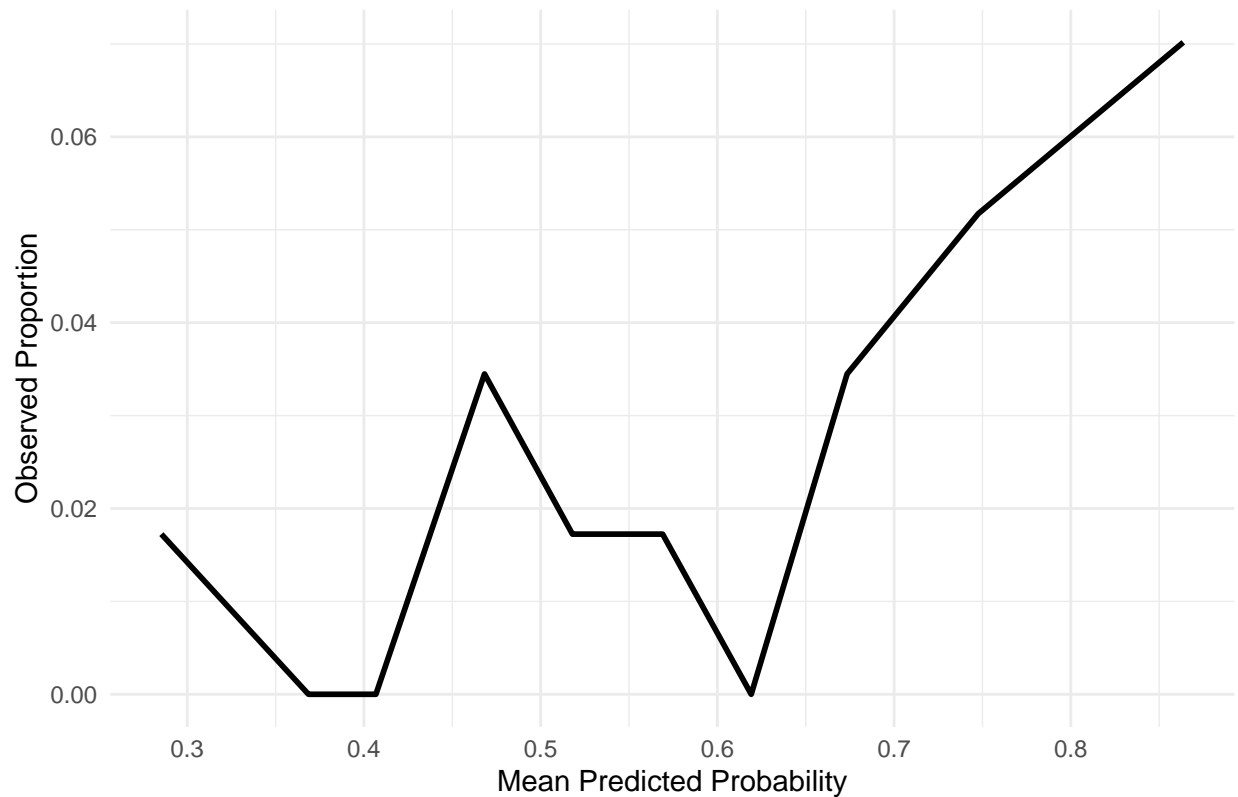
```

# 3 Calibration plot
calib_df <- tibble(
  obs = as.numeric(true_labels == "Known"),
  pred = pred_probs
) %>%
  mutate(decile = ntile(pred, 10)) %>%
  group_by(decile) %>%
  summarise(
    mean_pred = mean(pred),
    obs_rate = mean(obs),
    .groups = "drop"
  )

ggplot(calib_df, aes(mean_pred, obs_rate)) +
  geom_line(size = 1) +
  geom_abline(linetype = "dashed") +
  labs(
    title = "Calibration Plot",
    x = "Mean Predicted Probability",
    y = "Observed Proportion"
  ) +
  theme_minimal()

```

Calibration Plot



```
# 4 Brier score
```

```
brier_full <- mean((pred_probs - as.numeric(true_labels == "Known"))^2)
cat("Brier score:", round(brier_full,4), "\n")
```

```
## Brier score: 0.3244
```

```
# 5 Permutation test for AUC
```

```
set.seed(42)
perm_aucs <- replicate(1000, {
  perm_labels <- sample(true_labels)
  auc( roc(perm_labels, pred_probs, levels=c("Novel","Known"), direction="<") )
})
p_value <- mean(perm_aucs >= auc(roc_obj))
cat("Permutation p-value for AUC >", round(auc(roc_obj),3), ":", round(p_value,4), "\n")
```

```
## Permutation p-value for AUC > 0.706 : 0.001
```

```
# 6 Plot permutation null distribution
```

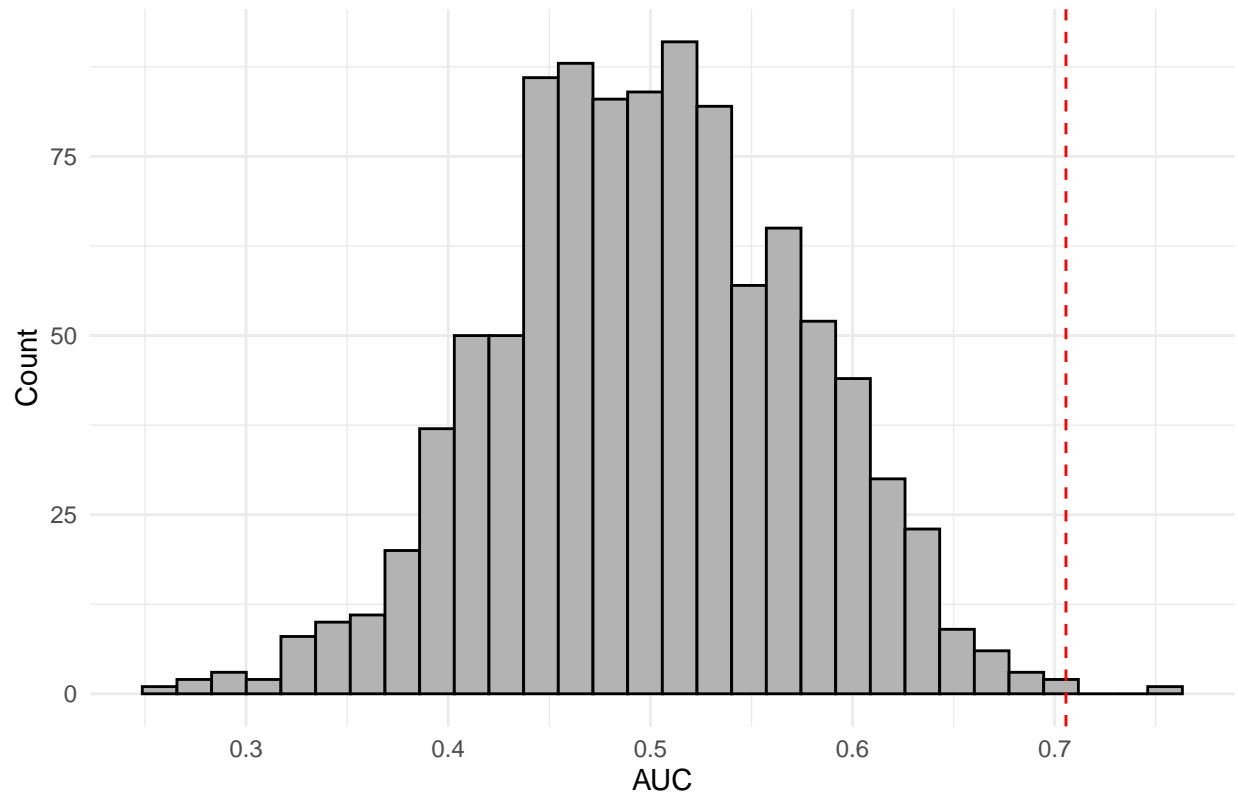
```
perm_df <- tibble(auc = perm_aucs)
ggplot(perm_df, aes(auc)) +
  geom_histogram(bins = 30, fill = "grey70", color = "black") +
  geom_vline(xintercept = auc(roc_obj), color = "red", linetype = "dashed") +
  labs(
    title = "Permutation Null Distribution of AUC",
```

```

x      = "AUC",
y      = "Count"
) +
theme_minimal()

```

Permutation Null Distribution of AUC



Improve Calibration via Platt Scaling or Isotonic Regression

```

# A.1 Platt Scaling (Logistic Calibration)

# assuming pred_probs and true_labels from before

# 1. Fit a logistic regression on your validation set
cal_data <- tibble(
  obs = as.numeric(true_labels == "Known"),
  raw = pred_probs
)
platt <- glm(obs ~ raw, data = cal_data, family = binomial)

# 2. Get calibrated probabilities
cal_data <- cal_data %>%
  mutate(cal_prob = predict(platt, newdata = cal_data, type = "response"))

# 3. Calibration plot after Platt
calib_platt <- cal_data %>%
  mutate(decile = ntile(cal_prob, 10)) %>%
  group_by(decile) %>%

```

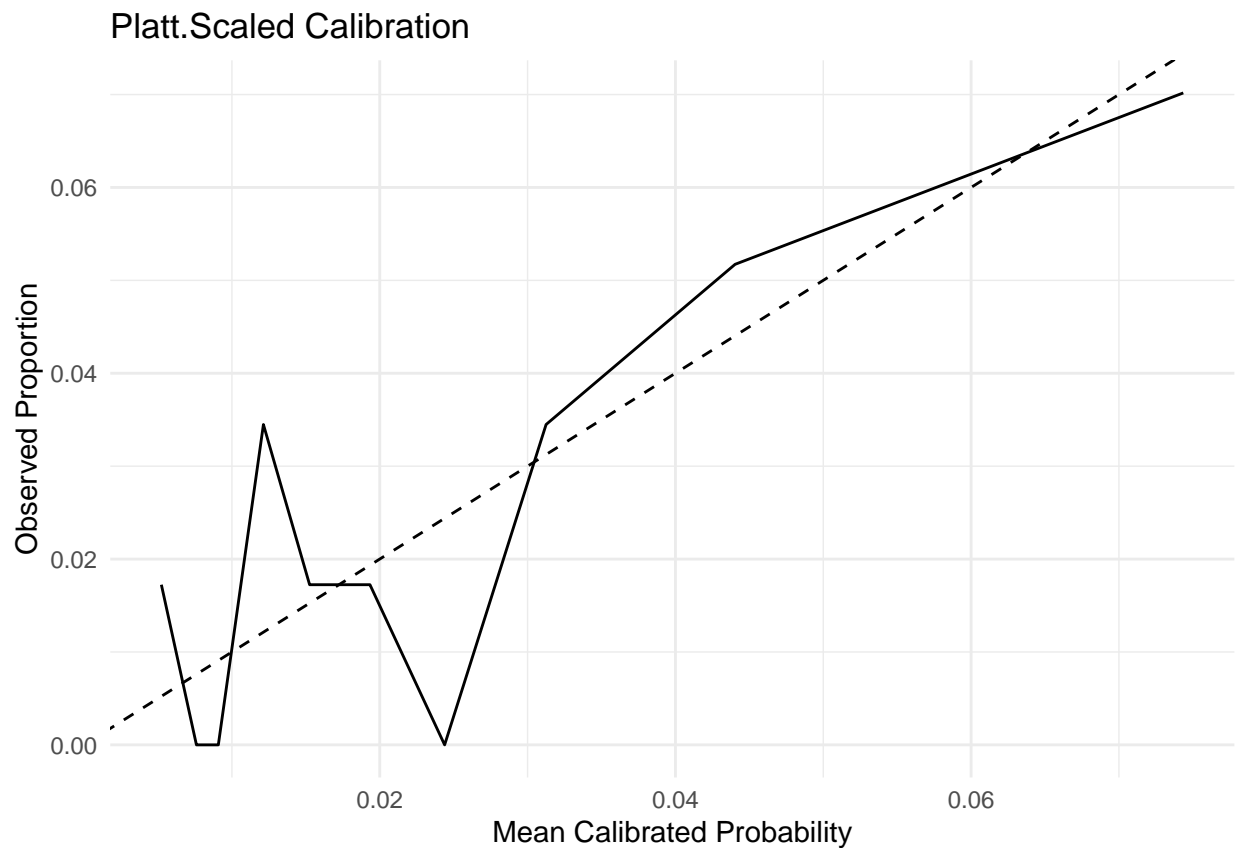


```

summarise(mean_pred = mean(cal_prob), obs_rate = mean(obs)) %>%
ggplot(aes(mean_pred, obs_rate)) +
  geom_line() + geom_abline(linetype="dashed") +
  labs(
    title = "Platt-Scaled Calibration",
    x     = "Mean Calibrated Probability",
    y     = "Observed Proportion"
  ) +
  theme_minimal()

print(calib_platt)

```



```

options(repos = c(CRAN = "https://cran.rstudio.com/"))
install.packages("isotone")

```

```

##
## The downloaded binary packages are in
## /var/folders/cy/7ww6_nh51zj22qj4p7wwf2080000gn/T//RtmpJHtAVi/downloaded_packages

```

```

library(isotone)

# Fit isotonic model
iso_fit <- isoreg(cal_data$raw, cal_data$obs)

```

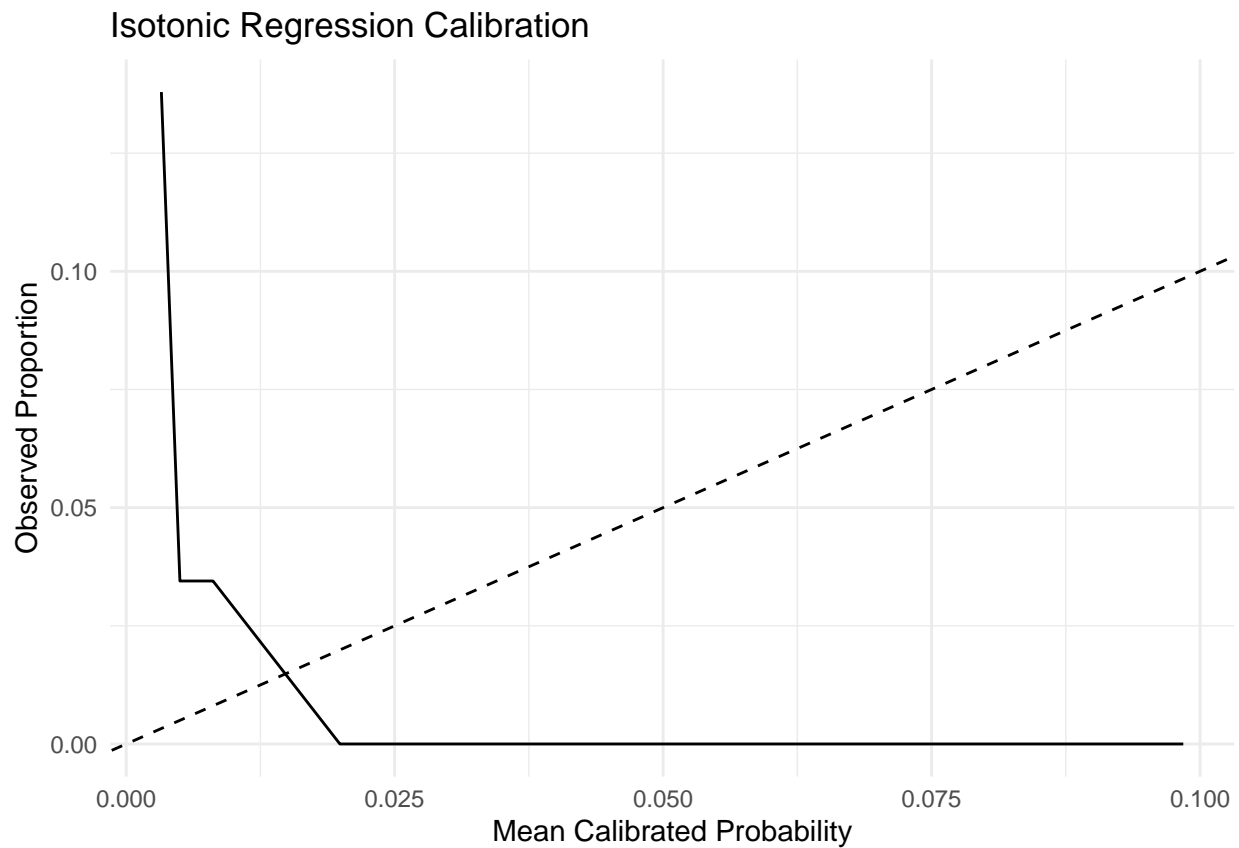
```

# Predict via monotonic interpolation
cal_data <- cal_data %>%
  mutate(iso_prob = {
    approx(x = iso_fit$x, y = iso_fit$yf, xout = raw, rule = 2)$y
  })

# Calibration plot after isotonic
calib_iso <- cal_data %>%
  mutate(decile = ntile(iso_prob, 10)) %>%
  group_by(decile) %>%
  summarise(mean_pred = mean(iso_prob), obs_rate = mean(obs)) %>%
  ggplot(aes(mean_pred, obs_rate)) +
    geom_line() + geom_abline(linetype="dashed") +
    labs(
      title = "Isotonic Regression Calibration",
      x = "Mean Calibrated Probability",
      y = "Observed Proportion"
    ) +
    theme_minimal()

print(calib_iso)

```



```

# Load required libraries
library(tidyverse)

```

```

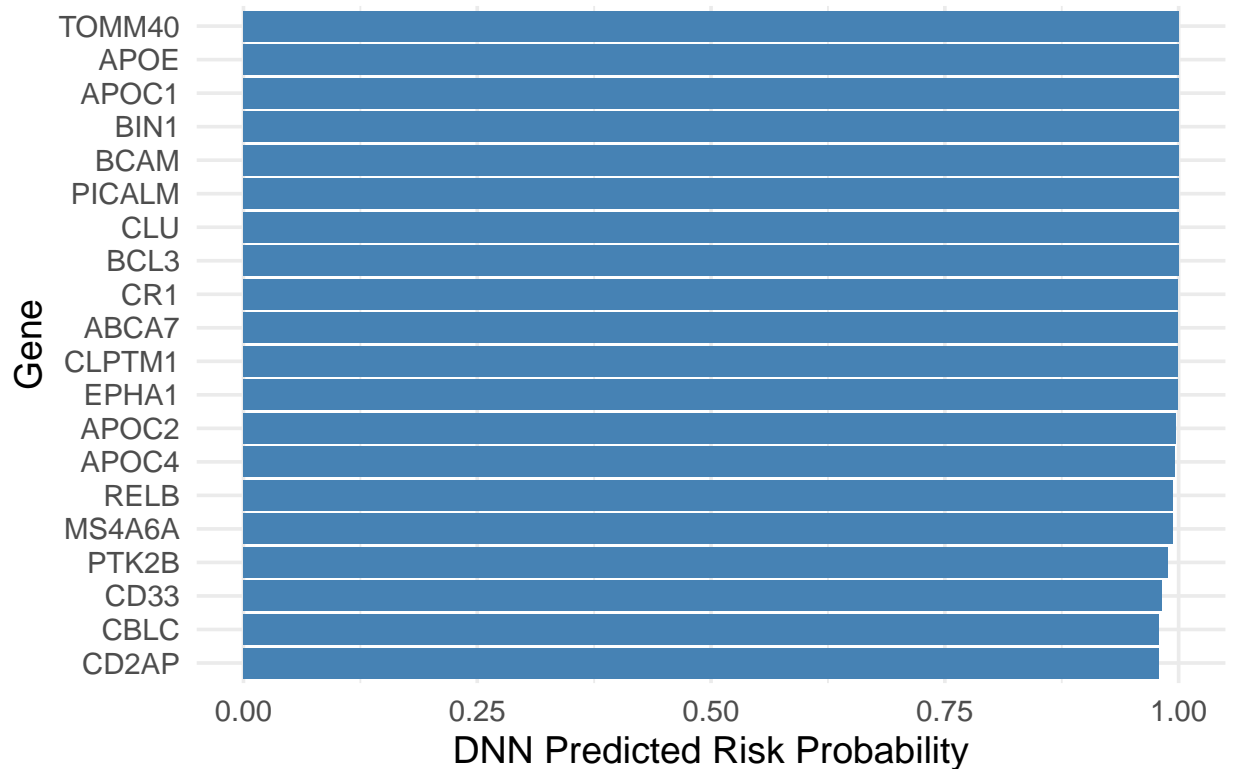
# 1. Read in the gene ranking CSV (ensure the working directory is set appropriately)
df <- read.csv("AD_gene_ranking.csv", stringsAsFactors = FALSE)

# 2. Select the top 20 genes by DNN-predicted risk probability
top20 <- df %>%
  arrange(desc(dnn_prob)) %>%
  slice_head(n = 20)

# 3. Bar plot of the top 20 risk-associated genes
ggplot(top20, aes(x = reorder(gene, dnn_prob), y = dnn_prob)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(
    title = "Top 20 Predicted Risk-Associated Genes for Alzheimer's Disease",
    x = "Gene",
    y = "DNN Predicted Risk Probability"
  ) +
  theme_minimal(base_size = 14)

```

Top 20 Predicted Risk-Associated Genes for Alzheimer



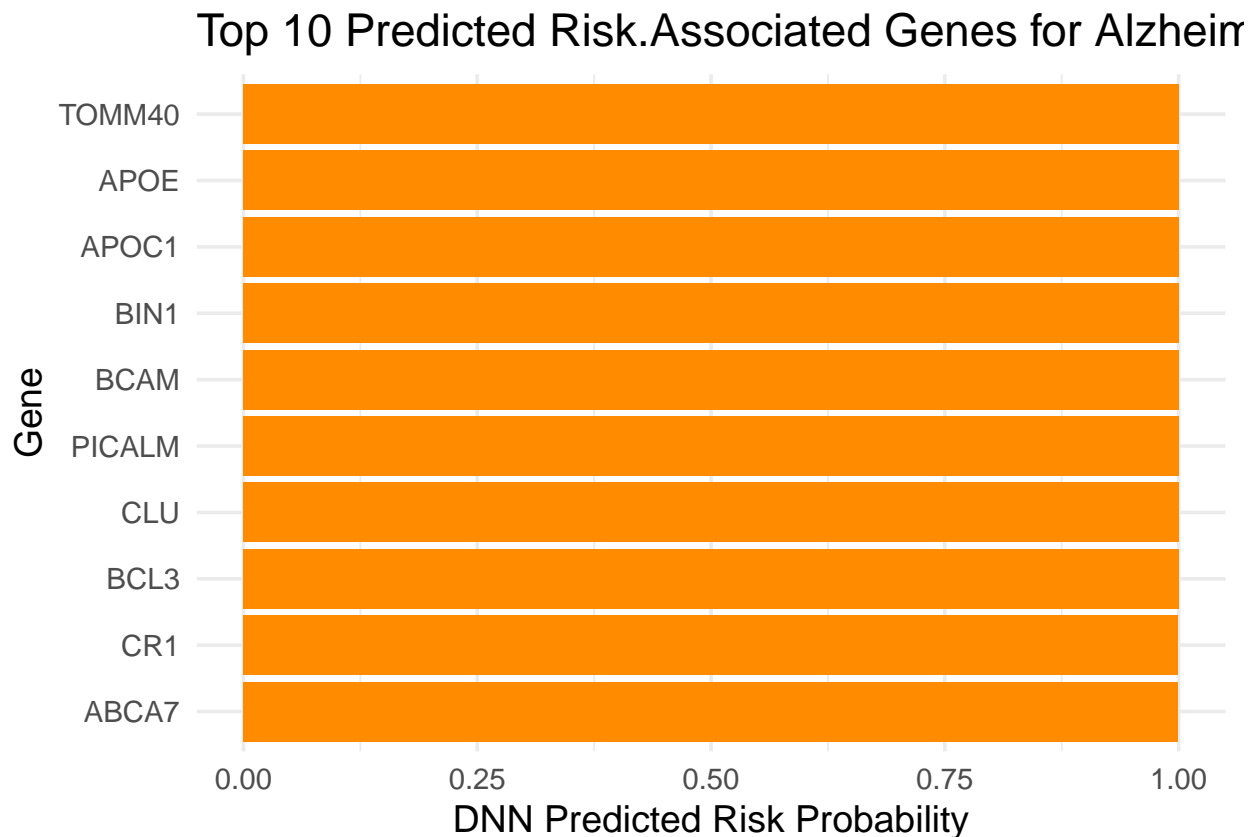
```

# 4. Save the plot to file
ggsave("top20_AD_risk_genes.png", width = 8, height = 6, dpi = 300)

# 5. If you want the top 10 only, simply adjust n:
top10 <- top20 %>% slice_head(n = 10)

```

```
ggplot(top10, aes(x = reorder(gene, dnn_prob), y = dnn_prob)) +
  geom_col(fill = "darkorange") +
  coord_flip() +
  labs(
    title = "Top 10 Predicted Risk-Associated Genes for Alzheimer's Disease",
    x = "Gene",
    y = "DNN Predicted Risk Probability"
  ) +
  theme_minimal(base_size = 14)
```



```
ggsave("top10_AD_risk_genes.png", width = 6, height = 4, dpi = 300)
```

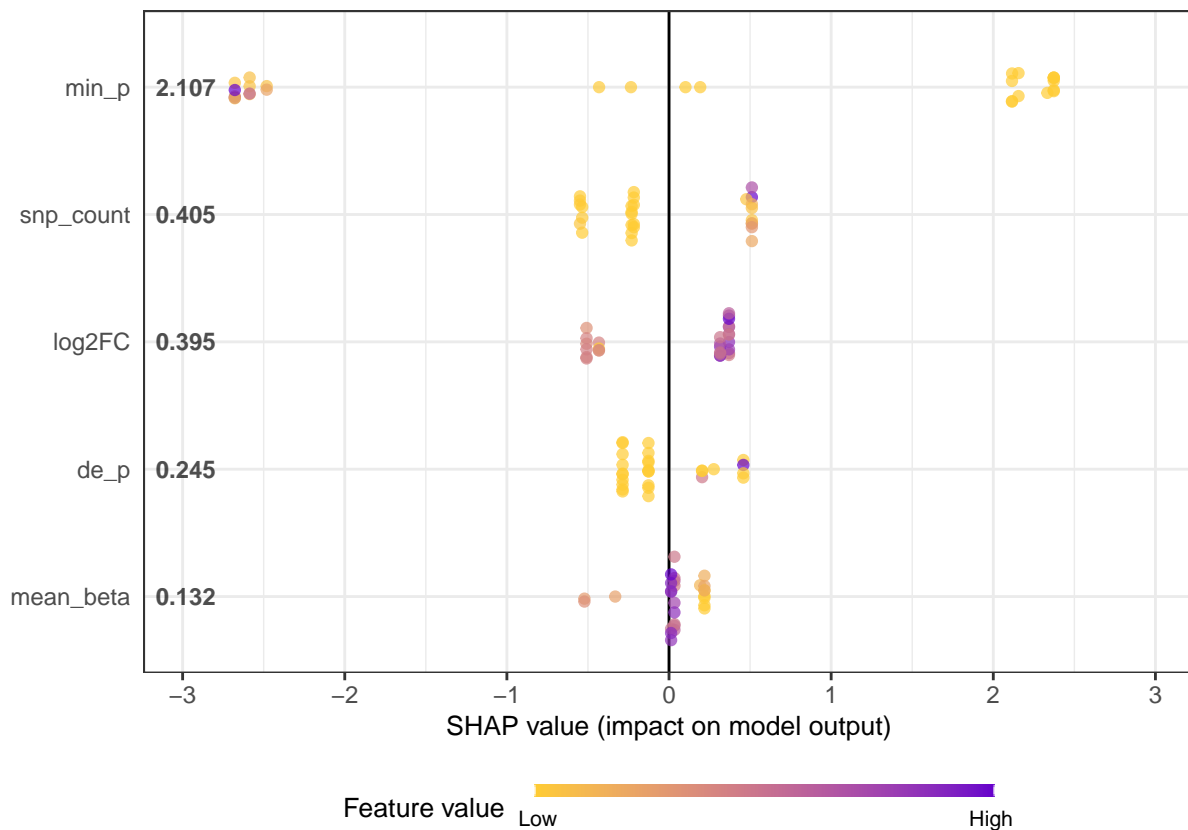
SHAP for Interpretability:

```
# Load required libraries
library(SHAPforxgboost)
library(xgboost)

# Prepare data (already run, but included for completeness)
xgb_data <- df_bal %>% select(min_p, mean_beta, snp_count, log2FC, de_p)
xgb_label <- ifelse(df_bal$label == "Known", 1, 0)
xgb_matrix <- xgb.DMatrix(data = as.matrix(xgb_data), label = xgb_label)
```

```
# Train XGBoost model
xgb_model <- xgb.train(params = list(max_depth = 4, eta = 0.1, objective = "binary:logistic"),
                      data = xgb_matrix, nrounds = 100)

# Alternative SHAP summary plot
shap.plot.summary.wrap1(model = xgb_model, X = as.matrix(xgb_data))
```



```
# Save plot
ggsave("shap_summary.png", width = 8, height = 6, dpi = 300)
```

```
library(smotefamily)

# 1) pull out predictors + outcome
X <- features %>% select(min_p, mean_beta, snp_count, log2FC, de_p) %>% as.data.frame()
y <- features$label

# 2) SMOTE wants y as a numeric factor: 0/1 (not "Known"/"Novel")
y_num <- ifelse(y=="Known", 1, 0)

# 3) run SMOTE: K=5 neighbors, dup_size controls amount of synthetic oversampling
sm <- SMOTE(X, y_num, K = 5, dup_size = 1)

# 4) extract the balanced data
smote_data <- sm$data
# sm$data comes back with all your predictors plus a column named "class" for the new 0/1 labels
```

```
# 5) rename and recast
colnames(smote_data)[ncol(smote_data)] <- "label_num"
smote_data$label <- factor(ifelse(smote_data$label_num==1, "Known", "Novel"),
                           levels = c("Known", "Novel"))

# now you can use `smote_data` in place of `df_bal` or `features`:
xgb_data <- smote_data %>% select(min_p, mean_beta, snp_count, log2FC, de_p)
xgb_label <- ifelse(smote_data$label=="Known", 1, 0)
xgb_matrix <- xgb.DMatrix(data=as.matrix(xgb_data), label=xgb_label)
```

```
head(smote_data)
```

```
##      min_p mean_beta snp_count      log2FC      de_p label_num label
## 1  1e-11  0.950000         1 -0.11797859 7.292071e-21         1 Known
## 2 2e-303  4.227199        87  0.12096133 2.442046e-14         1 Known
## 3  5e-17  2.870045        13  0.11644901 6.717150e-40         1 Known
## 4  5e-24  4.847332         6  0.26783154 6.736926e-30         1 Known
## 5  8e-26  3.037249        18  0.18027604 2.276487e-43         1 Known
## 6  1e-54  3.349342        24  0.06404468 1.551215e-11         1 Known
```

```
library(clusterProfiler)
library(org.Hs.eg.db)

entrez_ids <- mapIds(org.Hs.eg.db, keys = ranked_novel$gene,
                    column = "ENTREZID", keytype = "SYMBOL", multiVals = "first")
entrez_ids <- entrez_ids[!is.na(entrez_ids)]

cat("Number of Entrez IDs:", length(entrez_ids), "\n")
```

```
## Number of Entrez IDs: 20
```

```
cat("Sample Entrez IDs:", head(entrez_ids), "\n")
```

```
## Sample Entrez IDs: 79841 1234 29763 6688 169792 19
```

```
go <- enrichGO(gene = entrez_ids, OrgDb = org.Hs.eg.db,
               keyType = "ENTREZID", ont = "BP",
               pAdjustMethod = "BH", pvalueCutoff = 0.1)

if (is.null(go) || nrow(as.data.frame(go)) == 0) {
  cat("No significant GO terms found.\n")
} else {
  print(summary(go))
  dotplot(go, showCategory = 10) +
    labs(title = "GO Biological Process Enrichment of Top 20 Novel Genes")
  ggsave("go_dotplot.png", width = 8, height = 6, dpi = 300)
  write.csv(as.data.frame(go), "go_enrichment_results.csv")
}
```

```
##                               ID
```

G0:0070723 G0:0070723
G0:0036314 G0:0036314
G0:0010887 G0:0010887
G0:0071222 G0:0071222
G0:0010745 G0:0010745
G0:0071219 G0:0071219
G0:0071216 G0:0071216
G0:0010885 G0:0010885
G0:0071397 G0:0071397
G0:0010888 G0:0010888
G0:0010878 G0:0010878
G0:0036315 G0:0036315
G0:0015918 G0:0015918
G0:0010875 G0:0010875
G0:0032496 G0:0032496
G0:0010743 G0:0010743
G0:0002237 G0:0002237
G0:0032373 G0:0032373
G0:0032376 G0:0032376
G0:0006367 G0:0006367
G0:0010742 G0:0010742
G0:0090077 G0:0090077
G0:1903573 G0:1903573
G0:0051051 G0:0051051
G0:0006352 G0:0006352
G0:0006968 G0:0006968
G0:0045815 G0:0045815
G0:0010874 G0:0010874
G0:0010883 G0:0010883
G0:0141137 G0:0141137
G0:1902895 G0:1902895
G0:2000630 G0:2000630
G0:1905953 G0:1905953
G0:0051926 G0:0051926
G0:0033344 G0:0033344
G0:0097305 G0:0097305
G0:0045806 G0:0045806
G0:1902893 G0:1902893
G0:0034976 G0:0034976
G0:0032371 G0:0032371
G0:0032374 G0:0032374
G0:0061614 G0:0061614
G0:0015850 G0:0015850
G0:0032370 G0:0032370
G0:1905897 G0:1905897
G0:0009791 G0:0009791
G0:2000628 G0:2000628
G0:0070098 G0:0070098
G0:0030217 G0:0030217
G0:0019915 G0:0019915
G0:0008202 G0:0008202
G0:1990868 G0:1990868
G0:1990869 G0:1990869
G0:0051235 G0:0051235

G0:0042632 G0:0042632
G0:0055092 G0:0055092
G0:0010586 G0:0010586
G0:0097306 G0:0097306
G0:1905954 G0:1905954
G0:0035967 G0:0035967
G0:0002768 G0:0002768
G0:0030301 G0:0030301
G0:0043271 G0:0043271
G0:0006639 G0:0006639
G0:0006638 G0:0006638
G0:1901875 G0:1901875
G0:0045621 G0:0045621
G0:0014706 G0:0014706
G0:0030098 G0:0030098
G0:0010212 G0:0010212
G0:0046834 G0:0046834
G0:0072203 G0:0072203
G0:1905634 G0:1905634
G0:0006816 G0:0006816
G0:0060537 G0:0060537
G0:0010565 G0:0010565
G0:1903531 G0:1903531
G0:0002645 G0:0002645
G0:0002765 G0:0002765
G0:0002887 G0:0002887
G0:0072124 G0:0072124
G0:1904238 G0:1904238
G0:2000109 G0:2000109
G0:0032368 G0:0032368
G0:0048568 G0:0048568
G0:0006869 G0:0006869
G0:0035264 G0:0035264
G0:0035966 G0:0035966
G0:0009650 G0:0009650
G0:0030157 G0:0030157
G0:0043312 G0:0043312
G0:0048548 G0:0048548
G0:0060707 G0:0060707
G0:0072110 G0:0072110
G0:0140105 G0:0140105
G0:0007584 G0:0007584
G0:0002357 G0:0002357
G0:0002664 G0:0002664
G0:0006596 G0:0006596
G0:0034145 G0:0034145
G0:0034380 G0:0034380
G0:0036462 G0:0036462
G0:0043301 G0:0043301
G0:0044029 G0:0044029
G0:0070278 G0:0070278
G0:0070914 G0:0070914
G0:1903977 G0:1903977
G0:2001198 G0:2001198

G0:0007204 G0:0007204
G0:0051048 G0:0051048
G0:0000722 G0:0000722
G0:0003084 G0:0003084
G0:0010867 G0:0010867
G0:0033700 G0:0033700
G0:0051004 G0:0051004
G0:0071888 G0:0071888
G0:1904353 G0:1904353
G0:0007519 G0:0007519
G0:0055088 G0:0055088
G0:0001325 G0:0001325
G0:0002517 G0:0002517
G0:0032352 G0:0032352
G0:0036151 G0:0036151
G0:0060330 G0:0060330
G0:0060334 G0:0060334
G0:0060457 G0:0060457
G0:0072148 G0:0072148
G0:0090656 G0:0090656
G0:0090737 G0:0090737
G0:1900452 G0:1900452
G0:1901722 G0:1901722
G0:1904936 G0:1904936
G0:0001659 G0:0001659
G0:0071478 G0:0071478
G0:1905952 G0:1905952
G0:0034616 G0:0034616
G0:0044849 G0:0044849
G0:1904177 G0:1904177
G0:0006694 G0:0006694
G0:0060538 G0:0060538
G0:1902107 G0:1902107
G0:1903708 G0:1903708
G0:0032488 G0:0032488
G0:0042448 G0:0042448
G0:0043031 G0:0043031
G0:0045579 G0:0045579
G0:0051044 G0:0051044
G0:0055091 G0:0055091
G0:0072109 G0:0072109
G0:0072224 G0:0072224
G0:0090399 G0:0090399
G0:0002695 G0:0002695
G0:0046890 G0:0046890
G0:0071466 G0:0071466
G0:0006595 G0:0006595
G0:0045475 G0:0045475
G0:0071218 G0:0071218
G0:0002520 G0:0002520
G0:0050728 G0:0050728
G0:0007623 G0:0007623
G0:0022408 G0:0022408
G0:0050777 G0:0050777

##	G0:0002283	G0:0002283
##	G0:0002643	G0:0002643
##	G0:0042402	G0:0042402
##	G0:0097320	G0:0097320
##	G0:1904505	G0:1904505
##		Description
##	G0:0070723	response to cholesterol
##	G0:0036314	response to sterol
##	G0:0010887	negative regulation of cholesterol storage
##	G0:0071222	cellular response to lipopolysaccharide
##	G0:0010745	negative regulation of macrophage derived foam cell differentiation
##	G0:0071219	cellular response to molecule of bacterial origin
##	G0:0071216	cellular response to biotic stimulus
##	G0:0010885	regulation of cholesterol storage
##	G0:0071397	cellular response to cholesterol
##	G0:0010888	negative regulation of lipid storage
##	G0:0010878	cholesterol storage
##	G0:0036315	cellular response to sterol
##	G0:0015918	sterol transport
##	G0:0010875	positive regulation of cholesterol efflux
##	G0:0032496	response to lipopolysaccharide
##	G0:0010743	regulation of macrophage derived foam cell differentiation
##	G0:0002237	response to molecule of bacterial origin
##	G0:0032373	positive regulation of sterol transport
##	G0:0032376	positive regulation of cholesterol transport
##	G0:0006367	transcription initiation at RNA polymerase II promoter
##	G0:0010742	macrophage derived foam cell differentiation
##	G0:0090077	foam cell differentiation
##	G0:1903573	negative regulation of response to endoplasmic reticulum stress
##	G0:0051051	negative regulation of transport
##	G0:0006352	DNA-templated transcription initiation
##	G0:0006968	cellular defense response
##	G0:0045815	transcription initiation-coupled chromatin remodeling
##	G0:0010874	regulation of cholesterol efflux
##	G0:0010883	regulation of lipid storage
##	G0:0141137	positive regulation of gene expression, epigenetic
##	G0:1902895	positive regulation of miRNA transcription
##	G0:2000630	positive regulation of miRNA metabolic process
##	G0:1905953	negative regulation of lipid localization
##	G0:0051926	negative regulation of calcium ion transport
##	G0:0033344	cholesterol efflux
##	G0:0097305	response to alcohol
##	G0:0045806	negative regulation of endocytosis
##	G0:1902893	regulation of miRNA transcription
##	G0:0034976	response to endoplasmic reticulum stress
##	G0:0032371	regulation of sterol transport
##	G0:0032374	regulation of cholesterol transport
##	G0:0061614	miRNA transcription
##	G0:0015850	organic hydroxy compound transport
##	G0:0032370	positive regulation of lipid transport
##	G0:1905897	regulation of response to endoplasmic reticulum stress
##	G0:0009791	post-embryonic development
##	G0:2000628	regulation of miRNA metabolic process
##	G0:0070098	chemokine-mediated signaling pathway

GO:0030217 T cell differentiation
 ## GO:0019915 lipid storage
 ## GO:0008202 steroid metabolic process
 ## GO:1990868 response to chemokine
 ## GO:1990869 cellular response to chemokine
 ## GO:0051235 maintenance of location
 ## GO:0042632 cholesterol homeostasis
 ## GO:0055092 sterol homeostasis
 ## GO:0010586 miRNA metabolic process
 ## GO:0097306 cellular response to alcohol
 ## GO:1905954 positive regulation of lipid localization
 ## GO:0035967 cellular response to topologically incorrect protein
 ## GO:0002768 immune response-regulating cell surface receptor signaling pathway
 ## GO:0030301 cholesterol transport
 ## GO:0043271 negative regulation of monoatomic ion transport
 ## GO:0006639 acylglycerol metabolic process
 ## GO:0006638 neutral lipid metabolic process
 ## GO:1901875 positive regulation of post-translational protein modification
 ## GO:0045621 positive regulation of lymphocyte differentiation
 ## GO:0014706 striated muscle tissue development
 ## GO:0030098 lymphocyte differentiation
 ## GO:0010212 response to ionizing radiation
 ## GO:0046834 lipid phosphorylation
 ## GO:0072203 cell proliferation involved in metanephros development
 ## GO:1905634 regulation of protein localization to chromatin
 ## GO:0006816 calcium ion transport
 ## GO:0060537 muscle tissue development
 ## GO:0010565 regulation of cellular ketone metabolic process
 ## GO:1903531 negative regulation of secretion by cell
 ## GO:0002645 positive regulation of tolerance induction
 ## GO:0002765 immune response-inhibiting signal transduction
 ## GO:0002887 negative regulation of myeloid leukocyte mediated immunity
 ## GO:0072124 regulation of glomerular mesangial cell proliferation
 ## GO:1904238 pericyte cell differentiation
 ## GO:2000109 regulation of macrophage apoptotic process
 ## GO:0032368 regulation of lipid transport
 ## GO:0048568 embryonic organ development
 ## GO:0006869 lipid transport
 ## GO:0035264 multicellular organism growth
 ## GO:0035966 response to topologically incorrect protein
 ## GO:0009650 UV protection
 ## GO:0030157 pancreatic juice secretion
 ## GO:0043312 neutrophil degranulation
 ## GO:0048548 regulation of pinocytosis
 ## GO:0060707 trophoblast giant cell differentiation
 ## GO:0072110 glomerular mesangial cell proliferation
 ## GO:0140105 interleukin-10-mediated signaling pathway
 ## GO:0007584 response to nutrient
 ## GO:0002357 defense response to tumor cell
 ## GO:0002664 regulation of T cell tolerance induction
 ## GO:0006596 polyamine biosynthetic process
 ## GO:0034145 positive regulation of toll-like receptor 4 signaling pathway
 ## GO:0034380 high-density lipoprotein particle assembly
 ## GO:0036462 TRAIL-activated apoptotic signaling pathway

G0:0043301 negative regulation of leukocyte degranulation
 ## G0:0044029 positive regulation of gene expression via chromosomal CpG island demethylation
 ## G0:0070278 extracellular matrix constituent secretion
 ## G0:0070914 UV-damage excision repair
 ## G0:1903977 positive regulation of glial cell migration
 ## G0:2001198 regulation of dendritic cell differentiation
 ## G0:0007204 positive regulation of cytosolic calcium ion concentration
 ## G0:0051048 negative regulation of secretion
 ## G0:0000722 telomere maintenance via recombination
 ## G0:0003084 positive regulation of systemic arterial blood pressure
 ## G0:0010867 positive regulation of triglyceride biosynthetic process
 ## G0:0033700 phospholipid efflux
 ## G0:0051004 regulation of lipoprotein lipase activity
 ## G0:0071888 macrophage apoptotic process
 ## G0:1904353 regulation of telomere capping
 ## G0:0007519 skeletal muscle tissue development
 ## G0:0055088 lipid homeostasis
 ## G0:0001325 formation of extrachromosomal circular DNA
 ## G0:0002517 T cell tolerance induction
 ## G0:0032352 positive regulation of hormone metabolic process
 ## G0:0036151 phosphatidylcholine acyl-chain remodeling
 ## G0:0060330 regulation of response to type II interferon
 ## G0:0060334 regulation of type II interferon-mediated signaling pathway
 ## G0:0060457 negative regulation of digestive system process
 ## G0:0072148 epithelial cell fate commitment
 ## G0:0090656 t-circle formation
 ## G0:0090737 telomere maintenance via telomere trimming
 ## G0:1900452 regulation of long-term synaptic depression
 ## G0:1901722 regulation of cell proliferation involved in kidney development
 ## G0:1904936 interneuron migration
 ## G0:0001659 temperature homeostasis
 ## G0:0071478 cellular response to radiation
 ## G0:1905952 regulation of lipid localization
 ## G0:0034616 response to laminar fluid shear stress
 ## G0:0044849 estrous cycle
 ## G0:1904177 regulation of adipose tissue development
 ## G0:0006694 steroid biosynthetic process
 ## G0:0060538 skeletal muscle organ development
 ## G0:1902107 positive regulation of leukocyte differentiation
 ## G0:1903708 positive regulation of hemopoiesis
 ## G0:0032488 Cdc42 protein signal transduction
 ## G0:0042448 progesterone metabolic process
 ## G0:0043031 negative regulation of macrophage activation
 ## G0:0045579 positive regulation of B cell differentiation
 ## G0:0051044 positive regulation of membrane protein ectodomain proteolysis
 ## G0:0055091 phospholipid homeostasis
 ## G0:0072109 glomerular mesangium development
 ## G0:0072224 metanephric glomerulus development
 ## G0:0090399 replicative senescence
 ## G0:0002695 negative regulation of leukocyte activation
 ## G0:0046890 regulation of lipid biosynthetic process
 ## G0:0071466 cellular response to xenobiotic stimulus
 ## G0:0006595 polyamine metabolic process
 ## G0:0045475 locomotor rhythm

## G0:0071218					cellular response to misfolded protein
## G0:0002520					immune system development
## G0:0050728					negative regulation of inflammatory response
## G0:0007623					circadian rhythm
## G0:0022408					negative regulation of cell-cell adhesion
## G0:0050777					negative regulation of immune response
## G0:0002283					neutrophil activation involved in immune response
## G0:0002643					regulation of tolerance induction
## G0:0042402					cellular biogenic amine catabolic process
## G0:0097320					plasma membrane tubulation
## G0:1904505					regulation of telomere maintenance in response to DNA damage
##	GeneRatio	BgRatio	RichFactor	FoldEnrichment	zScore
## G0:0070723	3/20	31/18986	0.096774194	91.867742	16.442245
## G0:0036314	3/20	36/18986	0.083333333	79.108333	15.232684
## G0:0010887	2/20	11/18986	0.181818182	172.600000	18.486520
## G0:0071222	4/20	228/18986	0.017543860	16.654386	7.722228
## G0:0010745	2/20	14/18986	0.142857143	135.614286	16.361798
## G0:0071219	4/20	242/18986	0.016528926	15.690909	7.468917
## G0:0071216	4/20	269/18986	0.014869888	14.115985	7.035442
## G0:0010885	2/20	19/18986	0.105263158	99.926316	14.009471
## G0:0071397	2/20	21/18986	0.095238095	90.409524	13.312188
## G0:0010888	2/20	23/18986	0.086956522	82.547826	12.707356
## G0:0010878	2/20	25/18986	0.080000000	75.944000	12.176113
## G0:0036315	2/20	26/18986	0.076923077	73.023077	11.933603
## G0:0015918	3/20	132/18986	0.022727273	21.575000	7.702957
## G0:0010875	2/20	27/18986	0.074074074	70.318519	11.704581
## G0:0032496	4/20	354/18986	0.011299435	10.726554	5.998785
## G0:0010743	2/20	33/18986	0.060606061	57.533333	10.554921
## G0:0002237	4/20	375/18986	0.010666667	10.125867	5.796119
## G0:0032373	2/20	36/18986	0.055555556	52.738889	10.090116
## G0:0032376	2/20	36/18986	0.055555556	52.738889	10.090116
## G0:0006367	3/20	167/18986	0.017964072	17.053293	6.766382
## G0:0010742	2/20	38/18986	0.052631579	49.963158	9.810970
## G0:0090077	2/20	39/18986	0.051282051	48.682051	9.679422
## G0:1903573	2/20	46/18986	0.043478261	41.273913	8.880659
## G0:0051051	4/20	467/18986	0.008565310	8.131049	5.066821
## G0:0006352	3/20	209/18986	0.014354067	13.626316	5.960314
## G0:0006968	2/20	52/18986	0.038461538	36.511538	8.326882
## G0:0045815	2/20	52/18986	0.038461538	36.511538	8.326882
## G0:0010874	2/20	54/18986	0.037037037	35.159259	8.162807
## G0:0010883	2/20	55/18986	0.036363636	34.520000	8.084087
## G0:0141137	2/20	56/18986	0.035714286	33.903571	8.007449
## G0:1902895	2/20	56/18986	0.035714286	33.903571	8.007449
## G0:2000630	2/20	67/18986	0.029850746	28.337313	7.279081
## G0:1905953	2/20	68/18986	0.029411765	27.920588	7.221606
## G0:0051926	2/20	71/18986	0.028169014	26.740845	7.056368
## G0:0033344	2/20	74/18986	0.027027027	25.656757	6.901055
## G0:0097305	3/20	275/18986	0.010909091	10.356000	5.075052
## G0:0045806	2/20	76/18986	0.026315789	24.981579	6.802542
## G0:1902893	2/20	76/18986	0.026315789	24.981579	6.802542
## G0:0034976	3/20	277/18986	0.010830325	10.281227	5.053037
## G0:0032371	2/20	77/18986	0.025974026	24.657143	6.754696
## G0:0032374	2/20	77/18986	0.025974026	24.657143	6.754696
## G0:0061614	2/20	77/18986	0.025974026	24.657143	6.754696

## G0:0015850	3/20	292/18986	0.010273973	9.753082	4.894786
## G0:0032370	2/20	88/18986	0.022727273	21.575000	6.282112
## G0:1905897	2/20	88/18986	0.022727273	21.575000	6.282112
## G0:0009791	2/20	89/18986	0.022471910	21.332584	6.243435
## G0:2000628	2/20	92/18986	0.021739130	20.636957	6.131102
## G0:0070098	2/20	93/18986	0.021505376	20.415054	6.094836
## G0:0030217	3/20	320/18986	0.009375000	8.899688	4.627982
## G0:0019915	2/20	97/18986	0.020618557	19.573196	5.955256
## G0:0008202	3/20	334/18986	0.008982036	8.526647	4.506553
## G0:1990868	2/20	102/18986	0.019607843	18.613725	5.792109
## G0:1990869	2/20	102/18986	0.019607843	18.613725	5.792109
## G0:0051235	3/20	345/18986	0.008695652	8.254783	4.416028
## G0:0042632	2/20	106/18986	0.018867925	17.911321	5.669724
## G0:0055092	2/20	107/18986	0.018691589	17.743925	5.640169
## G0:0010586	2/20	110/18986	0.018181818	17.260000	5.553852
## G0:0097306	2/20	110/18986	0.018181818	17.260000	5.553852
## G0:1905954	2/20	113/18986	0.017699115	16.801770	5.470876
## G0:0035967	2/20	114/18986	0.017543860	16.654386	5.443922
## G0:0002768	3/20	369/18986	0.008130081	7.717886	4.231783
## G0:0030301	2/20	123/18986	0.016260163	15.435772	5.215784
## G0:0043271	2/20	134/18986	0.014925373	14.168657	4.967611
## G0:0006639	2/20	135/18986	0.014814815	14.063704	4.946505
## G0:0006638	2/20	136/18986	0.014705882	13.960294	4.925622
## G0:1901875	2/20	136/18986	0.014705882	13.960294	4.925622
## G0:0045621	2/20	137/18986	0.014598540	13.858394	4.904958
## G0:0014706	3/20	424/18986	0.007075472	6.716745	3.865911
## G0:0030098	3/20	426/18986	0.007042254	6.685211	3.853850
## G0:0010212	2/20	143/18986	0.013986014	13.276923	4.785364
## G0:0046834	1/20	10/18986	0.100000000	94.930000	9.647936
## G0:0072203	1/20	10/18986	0.100000000	94.930000	9.647936
## G0:1905634	1/20	10/18986	0.100000000	94.930000	9.647936
## G0:0006816	3/20	445/18986	0.006741573	6.399775	3.743015
## G0:0060537	3/20	447/18986	0.006711409	6.371141	3.731725
## G0:0010565	2/20	153/18986	0.013071895	12.409150	4.601206
## G0:1903531	2/20	154/18986	0.012987013	12.328571	4.583737
## G0:0002645	1/20	11/18986	0.090909091	86.300000	9.189395
## G0:0002765	1/20	11/18986	0.090909091	86.300000	9.189395
## G0:0002887	1/20	11/18986	0.090909091	86.300000	9.189395
## G0:0072124	1/20	11/18986	0.090909091	86.300000	9.189395
## G0:1904238	1/20	11/18986	0.090909091	86.300000	9.189395
## G0:2000109	1/20	11/18986	0.090909091	86.300000	9.189395
## G0:0032368	2/20	156/18986	0.012820513	12.170513	4.549280
## G0:0048568	3/20	458/18986	0.006550218	6.218122	3.670837
## G0:0006869	3/20	459/18986	0.006535948	6.204575	3.665401
## G0:0035264	2/20	158/18986	0.012658228	12.016456	4.515447
## G0:0035966	2/20	160/18986	0.012500000	11.866250	4.482219
## G0:0009650	1/20	12/18986	0.083333333	79.108333	8.789030
## G0:0030157	1/20	12/18986	0.083333333	79.108333	8.789030
## G0:0043312	1/20	12/18986	0.083333333	79.108333	8.789030
## G0:0048548	1/20	12/18986	0.083333333	79.108333	8.789030
## G0:0060707	1/20	12/18986	0.083333333	79.108333	8.789030
## G0:0072110	1/20	12/18986	0.083333333	79.108333	8.789030
## G0:0140105	1/20	12/18986	0.083333333	79.108333	8.789030
## G0:0007584	2/20	168/18986	0.011904762	11.301190	4.354996

## G0:0002357	1/20	13/18986	0.076923077	73.023077	8.435441
## G0:0002664	1/20	13/18986	0.076923077	73.023077	8.435441
## G0:0006596	1/20	13/18986	0.076923077	73.023077	8.435441
## G0:0034145	1/20	13/18986	0.076923077	73.023077	8.435441
## G0:0034380	1/20	13/18986	0.076923077	73.023077	8.435441
## G0:0036462	1/20	13/18986	0.076923077	73.023077	8.435441
## G0:0043301	1/20	13/18986	0.076923077	73.023077	8.435441
## G0:0044029	1/20	13/18986	0.076923077	73.023077	8.435441
## G0:0070278	1/20	13/18986	0.076923077	73.023077	8.435441
## G0:0070914	1/20	13/18986	0.076923077	73.023077	8.435441
## G0:1903977	1/20	13/18986	0.076923077	73.023077	8.435441
## G0:2001198	1/20	13/18986	0.076923077	73.023077	8.435441
## G0:0007204	2/20	170/18986	0.011764706	11.168235	4.324529
## G0:0051048	2/20	175/18986	0.011428571	10.849143	4.250538
## G0:0000722	1/20	14/18986	0.071428571	67.807143	8.120126
## G0:0003084	1/20	14/18986	0.071428571	67.807143	8.120126
## G0:0010867	1/20	14/18986	0.071428571	67.807143	8.120126
## G0:0033700	1/20	14/18986	0.071428571	67.807143	8.120126
## G0:0051004	1/20	14/18986	0.071428571	67.807143	8.120126
## G0:0071888	1/20	14/18986	0.071428571	67.807143	8.120126
## G0:1904353	1/20	14/18986	0.071428571	67.807143	8.120126
## G0:0007519	2/20	178/18986	0.011235955	10.666292	4.207567
## G0:0055088	2/20	180/18986	0.011111111	10.547778	4.179485
## G0:0001325	1/20	15/18986	0.066666667	63.286667	7.836606
## G0:0002517	1/20	15/18986	0.066666667	63.286667	7.836606
## G0:0032352	1/20	15/18986	0.066666667	63.286667	7.836606
## G0:0036151	1/20	15/18986	0.066666667	63.286667	7.836606
## G0:0060330	1/20	15/18986	0.066666667	63.286667	7.836606
## G0:0060334	1/20	15/18986	0.066666667	63.286667	7.836606
## G0:0060457	1/20	15/18986	0.066666667	63.286667	7.836606
## G0:0072148	1/20	15/18986	0.066666667	63.286667	7.836606
## G0:0090656	1/20	15/18986	0.066666667	63.286667	7.836606
## G0:0090737	1/20	15/18986	0.066666667	63.286667	7.836606
## G0:1900452	1/20	15/18986	0.066666667	63.286667	7.836606
## G0:1901722	1/20	15/18986	0.066666667	63.286667	7.836606
## G0:1904936	1/20	15/18986	0.066666667	63.286667	7.836606
## G0:0001659	2/20	185/18986	0.010810811	10.262703	4.111171
## G0:0071478	2/20	185/18986	0.010810811	10.262703	4.111171
## G0:1905952	2/20	188/18986	0.010638298	10.098936	4.071422
## G0:0034616	1/20	16/18986	0.062500000	59.331250	7.579840
## G0:0044849	1/20	16/18986	0.062500000	59.331250	7.579840
## G0:1904177	1/20	16/18986	0.062500000	59.331250	7.579840
## G0:0006694	2/20	191/18986	0.010471204	9.940314	4.032559
## G0:0060538	2/20	192/18986	0.010416667	9.888542	4.019795
## G0:1902107	2/20	195/18986	0.010256410	9.736410	3.982060
## G0:1903708	2/20	195/18986	0.010256410	9.736410	3.982060
## G0:0032488	1/20	17/18986	0.058823529	55.841176	7.345839
## G0:0042448	1/20	17/18986	0.058823529	55.841176	7.345839
## G0:0043031	1/20	17/18986	0.058823529	55.841176	7.345839
## G0:0045579	1/20	17/18986	0.058823529	55.841176	7.345839
## G0:0051044	1/20	17/18986	0.058823529	55.841176	7.345839
## G0:0055091	1/20	17/18986	0.058823529	55.841176	7.345839
## G0:0072109	1/20	17/18986	0.058823529	55.841176	7.345839
## G0:0072224	1/20	17/18986	0.058823529	55.841176	7.345839

##	G0:0090399	1/20	17/18986	0.058823529	55.841176	7.345839	
##	G0:0002695	2/20	196/18986	0.010204082	9.686735	3.969663	
##	G0:0046890	2/20	196/18986	0.010204082	9.686735	3.969663	
##	G0:0071466	2/20	198/18986	0.010101010	9.588889	3.945134	
##	G0:0006595	1/20	18/18986	0.055555556	52.738889	7.131403	
##	G0:0045475	1/20	18/18986	0.055555556	52.738889	7.131403	
##	G0:0071218	1/20	18/18986	0.055555556	52.738889	7.131403	
##	G0:0002520	2/20	202/18986	0.009900990	9.399010	3.897106	
##	G0:0050728	2/20	203/18986	0.009852217	9.352709	3.885307	
##	G0:0007623	2/20	206/18986	0.009708738	9.216505	3.850396	
##	G0:0022408	2/20	207/18986	0.009661836	9.171981	3.838917	
##	G0:0050777	2/20	207/18986	0.009661836	9.171981	3.838917	
##	G0:0002283	1/20	19/18986	0.052631579	49.963158	6.933928	
##	G0:0002643	1/20	19/18986	0.052631579	49.963158	6.933928	
##	G0:0042402	1/20	19/18986	0.052631579	49.963158	6.933928	
##	G0:0097320	1/20	19/18986	0.052631579	49.963158	6.933928	
##	G0:1904505	1/20	19/18986	0.052631579	49.963158	6.933928	
##		pvalue	p.adjust	qvalue	geneID	Count	
##	G0:0070723	4.409447e-06	0.002914407	0.001829656	1234/19/114881	3	
##	G0:0036314	6.980616e-06	0.002914407	0.001829656	1234/19/114881	3	
##	G0:0010887	5.765422e-05	0.014785120	0.009282054	19/10062	2	
##	G0:0071222	8.437529e-05	0.014785120	0.009282054	1234/19/10062/10288	4	
##	G0:0010745	9.521082e-05	0.014785120	0.009282054	19/10062	2	
##	G0:0071219	1.062404e-04	0.014785120	0.009282054	1234/19/10062/10288	4	
##	G0:0071216	1.596572e-04	0.018615094	0.011686501	1234/19/10062/10288	4	
##	G0:0010885	1.783482e-04	0.018615094	0.011686501	19/10062	2	
##	G0:0071397	2.187475e-04	0.020294907	0.012741083	19/114881	2	
##	G0:0010888	2.632058e-04	0.021724166	0.013638367	19/10062	2	
##	G0:0010878	3.117077e-04	0.021724166	0.013638367	19/10062	2	
##	G0:0036315	3.374701e-04	0.021724166	0.013638367	19/114881	2	
##	G0:0015918	3.434219e-04	0.021724166	0.013638367	19/10062/114881	3	
##	G0:0010875	3.642375e-04	0.021724166	0.013638367	19/10062	2	
##	G0:0032496	4.544882e-04	0.025299843	0.015883166	1234/19/10062/10288	4	
##	G0:0010743	5.458403e-04	0.027739765	0.017414942	19/10062	2	
##	G0:0002237	5.647617e-04	0.027739765	0.017414942	1234/19/10062/10288	4	
##	G0:0032373	6.500541e-04	0.028496858	0.017890243	19/10062	2	
##	G0:0032376	6.500541e-04	0.028496858	0.017890243	19/10062	2	
##	G0:0006367	6.825595e-04	0.028496858	0.017890243	6688/1958/2067	3	
##	G0:0010742	7.244623e-04	0.028806002	0.018084323	19/10062	2	
##	G0:0090077	7.631404e-04	0.028964649	0.018183921	19/10062	2	
##	G0:1903573	1.061225e-03	0.038527077	0.024187185	10062/90993	2	
##	G0:0051051	1.280181e-03	0.041887913	0.026297108	29763/6688/10062/10288	4	
##	G0:0006352	1.305465e-03	0.041887913	0.026297108	6688/1958/2067	3	
##	G0:0006968	1.354459e-03	0.041887913	0.026297108	1234/10288	2	
##	G0:0045815	1.354459e-03	0.041887913	0.026297108	6688/1958	2	
##	G0:0010874	1.459869e-03	0.042264119	0.026533289	19/10062	2	
##	G0:0010883	1.514003e-03	0.042264119	0.026533289	19/10062	2	
##	G0:0141137	1.569087e-03	0.042264119	0.026533289	6688/1958	2	
##	G0:1902895	1.569087e-03	0.042264119	0.026533289	6688/1958	2	
##	G0:2000630	2.237183e-03	0.058286121	0.036591854	6688/1958	2	
##	G0:1905953	2.303523e-03	0.058286121	0.036591854	19/10062	2	
##	G0:0051926	2.508093e-03	0.058490576	0.036720210	29763/10288	2	
##	G0:0033344	2.720949e-03	0.058490576	0.036720210	19/10062	2	
##	G0:0097305	2.854908e-03	0.058490576	0.036720210	1234/19/114881	3	

##	G0:0045806	2.867431e-03	0.058490576	0.036720210	29763/10062	2
##	G0:1902893	2.867431e-03	0.058490576	0.036720210	6688/1958	2
##	G0:0034976	2.913969e-03	0.058490576	0.036720210	23753/10062/90993	3
##	G0:0032371	2.942041e-03	0.058490576	0.036720210	19/10062	2
##	G0:0032374	2.942041e-03	0.058490576	0.036720210	19/10062	2
##	G0:0061614	2.942041e-03	0.058490576	0.036720210	6688/1958	2
##	G0:0015850	3.381161e-03	0.065657422	0.041219535	19/10062/114881	3
##	G0:0032370	3.822393e-03	0.070926634	0.044527531	19/10062	2
##	G0:1905897	3.822393e-03	0.070926634	0.044527531	10062/90993	2
##	G0:0009791	3.907801e-03	0.070935092	0.044532841	10848/2067	2
##	G0:2000628	4.169348e-03	0.074072459	0.046502470	6688/1958	2
##	G0:0070098	4.258298e-03	0.074076649	0.046505101	1234/9034	2
##	G0:0030217	4.371202e-03	0.074488858	0.046763884	6688/10288/1958	3
##	G0:0019915	4.622894e-03	0.077202336	0.048467398	19/10062	2
##	G0:0008202	4.925872e-03	0.080322186	0.050426030	19/114881/1958	3
##	G0:1990868	5.098294e-03	0.080322186	0.050426030	1234/9034	2
##	G0:1990869	5.098294e-03	0.080322186	0.050426030	1234/9034	2
##	G0:0051235	5.390406e-03	0.083351642	0.052327914	1234/19/10062	3
##	G0:0042632	5.494202e-03	0.083411977	0.052365792	19/10062	2
##	G0:0055092	5.595329e-03	0.083430358	0.052377332	19/10062	2
##	G0:0010586	5.903845e-03	0.084995006	0.053359613	6688/1958	2
##	G0:0097306	5.903845e-03	0.084995006	0.053359613	19/114881	2
##	G0:1905954	6.220024e-03	0.088029150	0.055264440	19/10062	2
##	G0:0035967	6.327112e-03	0.088052305	0.055278977	23753/90993	2
##	G0:0002768	6.493805e-03	0.088890608	0.055805260	10062/8525/10288	3
##	G0:0030301	7.328707e-03	0.097304227	0.061087306	19/10062	2
##	G0:0043271	8.643980e-03	0.097304227	0.061087306	29763/10288	2
##	G0:0006639	8.768449e-03	0.097304227	0.061087306	10062/8525	2
##	G0:0006638	8.893726e-03	0.097304227	0.061087306	10062/8525	2
##	G0:1901875	8.893726e-03	0.097304227	0.061087306	91947/1958	2
##	G0:0045621	9.019811e-03	0.097304227	0.061087306	6688/10288	2
##	G0:0014706	9.505214e-03	0.097304227	0.061087306	10848/7026/1958	3
##	G0:0030098	9.627762e-03	0.097304227	0.061087306	6688/10288/1958	3
##	G0:0010212	9.793186e-03	0.097304227	0.061087306	1958/2067	2
##	G0:0046834	1.048676e-02	0.097304227	0.061087306	8525	1
##	G0:0072203	1.048676e-02	0.097304227	0.061087306	1958	1
##	G0:1905634	1.048676e-02	0.097304227	0.061087306	6688	1
##	G0:0006816	1.083869e-02	0.097304227	0.061087306	1234/29763/10288	3
##	G0:0060537	1.097110e-02	0.097304227	0.061087306	10848/7026/1958	3
##	G0:0010565	1.114564e-02	0.097304227	0.061087306	10062/1958	2
##	G0:1903531	1.128520e-02	0.097304227	0.061087306	6688/10062	2
##	G0:0002645	1.152967e-02	0.097304227	0.061087306	10288	1
##	G0:0002765	1.152967e-02	0.097304227	0.061087306	10288	1
##	G0:0002887	1.152967e-02	0.097304227	0.061087306	6688	1
##	G0:0072124	1.152967e-02	0.097304227	0.061087306	1958	1
##	G0:1904238	1.152967e-02	0.097304227	0.061087306	6688	1
##	G0:2000109	1.152967e-02	0.097304227	0.061087306	1234	1
##	G0:0032368	1.156665e-02	0.097304227	0.061087306	19/10062	2
##	G0:0048568	1.171636e-02	0.097304227	0.061087306	10848/7026/2067	3
##	G0:0006869	1.178554e-02	0.097304227	0.061087306	19/10062/114881	3
##	G0:0035264	1.185120e-02	0.097304227	0.061087306	10848/2067	2
##	G0:0035966	1.213884e-02	0.097304227	0.061087306	23753/90993	2
##	G0:0009650	1.257153e-02	0.097304227	0.061087306	2067	1
##	G0:0030157	1.257153e-02	0.097304227	0.061087306	10062	1

## G0:0043312	1.257153e-02	0.097304227	0.061087306	6688	1
## G0:0048548	1.257153e-02	0.097304227	0.061087306	10062	1
## G0:0060707	1.257153e-02	0.097304227	0.061087306	7026	1
## G0:0072110	1.257153e-02	0.097304227	0.061087306	1958	1
## G0:0140105	1.257153e-02	0.097304227	0.061087306	10288	1
## G0:0007584	1.331998e-02	0.097304227	0.061087306	19/2067	2
## G0:0002357	1.361236e-02	0.097304227	0.061087306	6688	1
## G0:0002664	1.361236e-02	0.097304227	0.061087306	10288	1
## G0:0006596	1.361236e-02	0.097304227	0.061087306	54498	1
## G0:0034145	1.361236e-02	0.097304227	0.061087306	10062	1
## G0:0034380	1.361236e-02	0.097304227	0.061087306	19	1
## G0:0036462	1.361236e-02	0.097304227	0.061087306	6688	1
## G0:0043301	1.361236e-02	0.097304227	0.061087306	6688	1
## G0:0044029	1.361236e-02	0.097304227	0.061087306	1958	1
## G0:0070278	1.361236e-02	0.097304227	0.061087306	90993	1
## G0:0070914	1.361236e-02	0.097304227	0.061087306	2067	1
## G0:1903977	1.361236e-02	0.097304227	0.061087306	22800	1
## G0:2001198	1.361236e-02	0.097304227	0.061087306	10288	1
## G0:0007204	1.362285e-02	0.097304227	0.061087306	1234/9034	2
## G0:0051048	1.439316e-02	0.097304227	0.061087306	6688/10062	2
## G0:0000722	1.465214e-02	0.097304227	0.061087306	2067	1
## G0:0003084	1.465214e-02	0.097304227	0.061087306	7026	1
## G0:0010867	1.465214e-02	0.097304227	0.061087306	10062	1
## G0:0033700	1.465214e-02	0.097304227	0.061087306	19	1
## G0:0051004	1.465214e-02	0.097304227	0.061087306	10062	1
## G0:0071888	1.465214e-02	0.097304227	0.061087306	1234	1
## G0:1904353	1.465214e-02	0.097304227	0.061087306	2067	1
## G0:0007519	1.486429e-02	0.097304227	0.061087306	7026/1958	2
## G0:0055088	1.518208e-02	0.097304227	0.061087306	19/10062	2
## G0:0001325	1.569088e-02	0.097304227	0.061087306	2067	1
## G0:0002517	1.569088e-02	0.097304227	0.061087306	10288	1
## G0:0032352	1.569088e-02	0.097304227	0.061087306	1958	1
## G0:0036151	1.569088e-02	0.097304227	0.061087306	10062	1
## G0:0060330	1.569088e-02	0.097304227	0.061087306	10062	1
## G0:0060334	1.569088e-02	0.097304227	0.061087306	10062	1
## G0:0060457	1.569088e-02	0.097304227	0.061087306	10062	1
## G0:0072148	1.569088e-02	0.097304227	0.061087306	7026	1
## G0:0090656	1.569088e-02	0.097304227	0.061087306	2067	1
## G0:0090737	1.569088e-02	0.097304227	0.061087306	2067	1
## G0:1900452	1.569088e-02	0.097304227	0.061087306	10288	1
## G0:1901722	1.569088e-02	0.097304227	0.061087306	1958	1
## G0:1904936	1.569088e-02	0.097304227	0.061087306	7026	1
## G0:0001659	1.598942e-02	0.097304227	0.061087306	10062/1958	2
## G0:0071478	1.598942e-02	0.097304227	0.061087306	1958/2067	2
## G0:1905952	1.648257e-02	0.097304227	0.061087306	19/10062	2
## G0:0034616	1.672857e-02	0.097304227	0.061087306	19	1
## G0:0044849	1.672857e-02	0.097304227	0.061087306	1958	1
## G0:1904177	1.672857e-02	0.097304227	0.061087306	6688	1
## G0:0006694	1.698224e-02	0.097304227	0.061087306	114881/1958	2
## G0:0060538	1.715024e-02	0.097304227	0.061087306	7026/1958	2
## G0:1902107	1.765854e-02	0.097304227	0.061087306	6688/10288	2
## G0:1903708	1.765854e-02	0.097304227	0.061087306	6688/10288	2
## G0:0032488	1.776523e-02	0.097304227	0.061087306	19	1
## G0:0042448	1.776523e-02	0.097304227	0.061087306	1958	1

## G0:0043031	1.776523e-02	0.097304227	0.061087306	10062	1
## G0:0045579	1.776523e-02	0.097304227	0.061087306	6688	1
## G0:0051044	1.776523e-02	0.097304227	0.061087306	29763	1
## G0:0055091	1.776523e-02	0.097304227	0.061087306	19	1
## G0:0072109	1.776523e-02	0.097304227	0.061087306	1958	1
## G0:0072224	1.776523e-02	0.097304227	0.061087306	1958	1
## G0:0090399	1.776523e-02	0.097304227	0.061087306	2067	1
## G0:0002695	1.782940e-02	0.097304227	0.061087306	10062/10288	2
## G0:0046890	1.782940e-02	0.097304227	0.061087306	10062/1958	2
## G0:0071466	1.817325e-02	0.098536785	0.061861101	19/54498	2
## G0:0006595	1.880085e-02	0.099177189	0.062263145	54498	1
## G0:0045475	1.880085e-02	0.099177189	0.062263145	1958	1
## G0:0071218	1.880085e-02	0.099177189	0.062263145	23753	1
## G0:0002520	1.886945e-02	0.099177189	0.062263145	10288/2067	2
## G0:0050728	1.904526e-02	0.099177189	0.062263145	10062/10848	2
## G0:0007623	1.957689e-02	0.099177189	0.062263145	10062/1958	2
## G0:0022408	1.975550e-02	0.099177189	0.062263145	6688/10288	2
## G0:0050777	1.975550e-02	0.099177189	0.062263145	6688/10062	2
## G0:0002283	1.983544e-02	0.099177189	0.062263145	6688	1
## G0:0002643	1.983544e-02	0.099177189	0.062263145	10288	1
## G0:0042402	1.983544e-02	0.099177189	0.062263145	54498	1
## G0:0097320	1.983544e-02	0.099177189	0.062263145	29763	1
## G0:1904505	1.983544e-02	0.099177189	0.062263145	2067	1