Annotation

R Markdown.

This document presents a comprehensive statistical and correlation analysis of immune scores in relation to delta age and cancer subtypes. The analysis includes:

T-tests to evaluate correlations between immune scores and Delta_Hannum (a measure of delta age), Wilcoxon rank-sum tests to assess differences in immune scores across cancer subtypes, A comparative analysis of immune features using both raw data and SHAP (SHapley Additive exPlanations) features, highlighting feature importance and model interpretability. This analysis aims to uncover significant immune-related patterns and associations that may have biological or clinical relevance

This analysis also includes GO enrichment, reactome pathway and Kegg pathway analysis of top features to further identify top pathways common among features selected and top_genes.

Immune celö

```
Immunoscores = read.csv("/mnt/home/fayyaz/MPIB-SRT/1040-pelotas/private/data/autobids/CIBERSORTx_Job2_
immune_cells = read.csv("/mnt/home/fayyaz/MPIB-SRT/1040-pelotas/private/data/autobids/immune_cells.csv
```

t-test and Wilcox-test

```
#Initialize result data frame
cor_results <- data.frame(
  Cell_Type = character(),
  Corr_with_Delta = numeric(),
  Pval_with_Delta = numeric(),
  Subtype_Wilcox_P = numeric(),
  Subtype_Median_Diff = numeric()
)

 #Loop over immune cell columns
  for (cell in colnames(immune_cells)[!colnames(immune_cells) %in% c("delta_Hannum", "Subtype_binary")]
  x <- as.numeric(immune_cells[[cell]])
  y_delta <- as.numeric(immune_cells$delta_Hannum)
  y_subtype <- as.factor(immune_cells$Subtype_binary)

  if (all(is.na(x)) || length(na.omit(x)) < 3) next

 # Correlation with delta_Hannum
  test_delta <- cor.test(x, y_delta, method = "pearson")

 #  Wilcoxon test between subtypes
   test_subtype <- wilcox.test(x ~ y_subtype)

  # Median difference
   median_diff <- median(x[y_subtype == 1], na.rm = TRUE) -
```

```
              median(x[y_subtype == 0], na.rm = TRUE)

  cor_results <- rbind(cor_results, data.frame(
  Cell_Type = cell,
  Corr_with_Delta = test_delta$estimate,
  Pval_with_Delta = test_delta$p.value,
  Subtype_Wilcox_P = test_subtype$p.value,
  Subtype_Median_Diff = median_diff
 ))
}
```

```
## Warning: NAs introduced by coercion
## Warning: NAs introduced by coercion
```

```
# Sort by p-value with delta_Hannum (or change to Pval_with_Subtype)
# Sort by p-value (choose which one: Delta or Subtype)
cor_results <- cor_results[order(cor_results$Pval_with_Delta), ]

# View result
print(cor_results)
```

```
##                         Cell_Type Corr_with_Delta Pval_with_Delta
## cor19           Mast.cells.resting    1.557444e-01    0.0002624081
## cor15               Macrophages.M1   -1.449875e-01    0.0006864094
## cor21                  Eosinophils    8.767199e-02    0.0407598192
## cor7  T.cells.CD4.memory.activated   -7.820881e-02    0.0680906743
## cor8     T.cells.follicular.helper   -7.766910e-02    0.0700205190
## cor16               Macrophages.M2    6.894019e-02    0.1079134160
## cor3                  Plasma.cells    6.726850e-02    0.1167480924
## cor14               Macrophages.M0   -5.612639e-02    0.1907692330
## cor17      Dendritic.cells.resting   -4.550772e-02    0.2889178054
## cor5             T.cells.CD4.naive    4.103752e-02    0.3389541101
## cor6    T.cells.CD4.memory.resting   -3.961871e-02    0.3559305742
## cor2                B.cells.memory   -3.400772e-02    0.4281714908
## cor9    T.cells.regulatory..Tregs.   -2.677492e-02    0.5327960863
## cor20         Mast.cells.activated   -2.408619e-02    0.5747374586
## cor4                    T.cells.CD8   -1.470922e-02    0.7318840605
## cor                               X    1.130505e-02    0.7923027486
## cor22                  Neutrophils   -1.060274e-02    0.8049375753
## cor11             NK.cells.resting   -9.413765e-03    0.8264421959
## cor18    Dendritic.cells.activated   -9.215246e-03    0.8300458981
## cor10          T.cells.gamma.delta    3.637529e-03    0.9324806033
## cor1                 B.cells.naive   -3.233906e-03    0.9399575918
## cor13                   Monocytes   -6.229609e-04    0.9884232784
## cor12           NK.cells.activated    7.317116e-05    0.9986401852
##       Subtype_Wilcox_P Subtype_Median_Diff
## cor19     1.321126e-22        -0.046325179
## cor15     4.777185e-07         0.032259375
## cor21     5.884952e-01         0.000000000
## cor7      1.992189e-07         0.000000000
## cor8      6.242527e-16         0.035555216
## cor16     4.043979e-13        -0.096952417
## cor3      1.315302e-01        -0.009826864
## cor14     2.549554e-09         0.090761037
```

```
## cor17     4.419359e-01         0.000000000
## cor5      8.679405e-01         0.000000000
## cor6      5.294672e-03        -0.041735451
## cor2      7.449030e-08         0.000000000
## cor9      8.622964e-01        -0.000305942
## cor20     2.134916e-02         0.000000000
## cor4      9.162786e-01        -0.005725090
## cor       3.648510e-01         5.500000000
## cor22     1.898111e-01         0.000000000
## cor11     4.634209e-02         0.010366707
## cor18     1.158042e-06         0.000000000
## cor10     1.343306e-01         0.000000000
## cor1      1.938400e-05        -0.030818625
## cor13     1.807910e-01        -0.004258575
## cor12     1.083445e-01         0.000000000
```

```r
# Adjust p-values for multiple testing (Benjamini-Hochberg method)
cor_results$FDR_with_Delta <- p.adjust(cor_results$Pval_with_Delta, method = "BH")
cor_results$FDR_with_Subtype <- p.adjust(cor_results$Subtype_Wilcox_P, method = "BH")

# Filter significant results
significant_fdr <- subset(cor_results, FDR_with_Delta < 0.05 | FDR_with_Subtype < 0.05)

print(significant_fdr)
```

```
##                          Cell_Type Corr_with_Delta Pval_with_Delta
## cor19           Mast.cells.resting     0.155744363    0.0002624081
## cor15               Macrophages.M1    -0.144987497    0.0006864094
## cor7  T.cells.CD4.memory.activated    -0.078208812    0.0680906743
## cor8     T.cells.follicular.helper    -0.077669100    0.0700205190
## cor16               Macrophages.M2     0.068940192    0.1079134160
## cor14               Macrophages.M0    -0.056126394    0.1907692330
## cor6    T.cells.CD4.memory.resting    -0.039618713    0.3559305742
## cor2                B.cells.memory    -0.034007717    0.4281714908
## cor20         Mast.cells.activated    -0.024086190    0.5747374586
## cor18    Dendritic.cells.activated    -0.009215246    0.8300458981
## cor1                  B.cells.naive    -0.003233906    0.9399575918
##       Subtype_Wilcox_P Subtype_Median_Diff FDR_with_Delta FDR_with_Subtype
## cor19     1.321126e-22         -0.04632518    0.006035386     3.038591e-21
## cor15     4.777185e-07          0.03225937    0.007893708     1.569647e-06
## cor7      1.992189e-07          0.00000000    0.322094387     7.636723e-07
## cor8      6.242527e-16          0.03555522    0.322094387     7.178906e-15
## cor16     4.043979e-13         -0.09695242    0.383600875     3.100384e-12
## cor14     2.549554e-09          0.09076104    0.548461545     1.465994e-08
## cor6      5.294672e-03         -0.04173545    0.744218473     1.217774e-02
## cor2      7.449030e-08          0.00000000    0.820662024     3.426554e-07
## cor20     2.134916e-02          0.00000000    0.944211539     4.463916e-02
## cor18     1.158042e-06          0.00000000    0.998640185     3.329371e-06
## cor1      1.938400e-05         -0.03081862    0.998640185     4.953688e-05
```

Immune cel correlation with Delta age

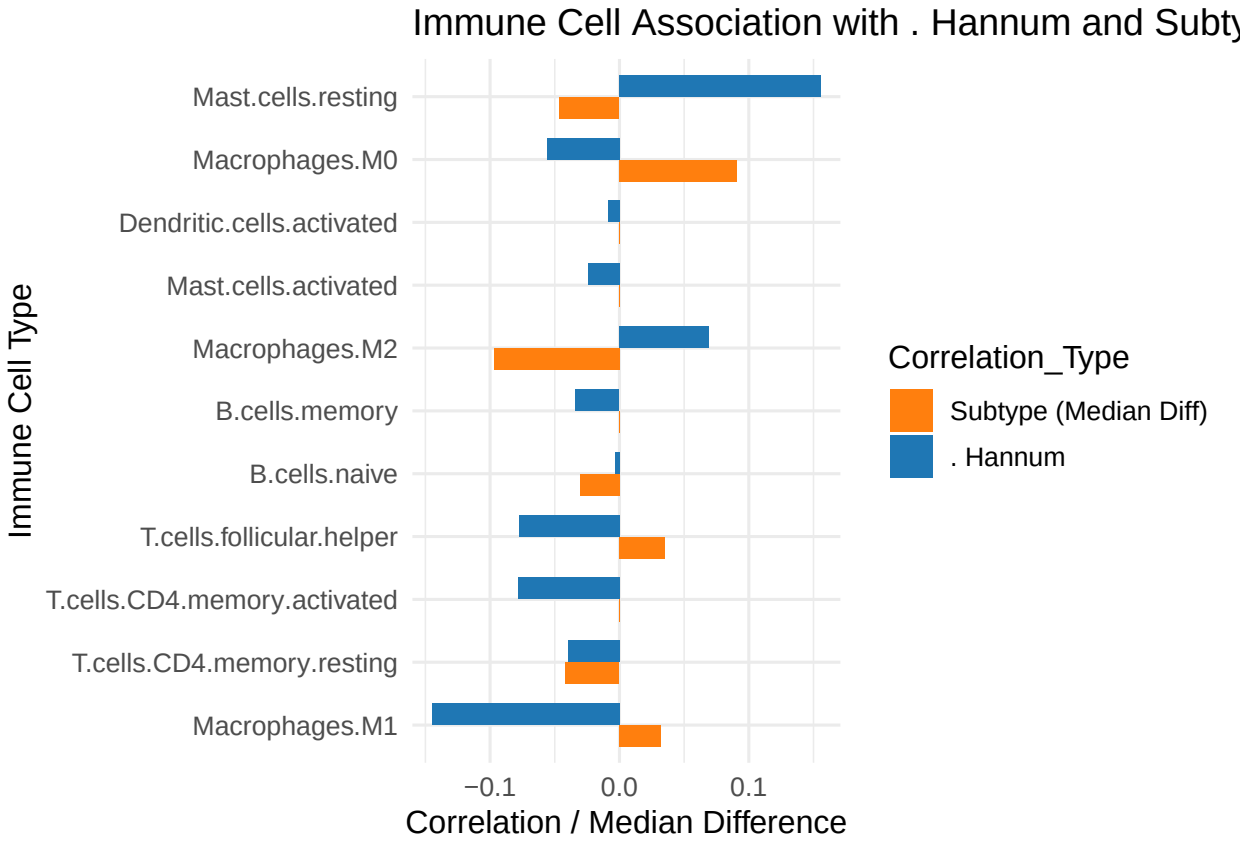
```r
cor_long = read.csv("/mnt/home/fayyaz/MPIB-SRT/1040-pelotas/private/data/autobids/cor_long.csv")
```

```r
# Load necessary libraries
library(reshape2)
library(ggplot2)
library(forcats)

cor_long$Correlation_Type <- fct_recode(
  cor_long$Correlation_Type,
  "Δ Hannum" = "Corr_with_Delta",
  "Subtype (Median Diff)" = "Subtype_Median_Diff"
)
```

```
## Warning: Unknown levels in `f`: Corr_with_Delta, Subtype_Median_Diff
```

```r
# Plot
ggplot(cor_long, aes(x = reorder(Cell_Type, Correlation), y = Correlation, fill = Correlation_Type)) +
  geom_bar(stat = "identity", position = "dodge", width = 0.7) +
  coord_flip() +
  labs(title = "Immune Cell Association with Δ Hannum and Subtype",
       x = "Immune Cell Type",
       y = "Correlation / Median Difference") +
  scale_fill_manual(values = c("Δ Hannum" = "#1f77b4", "Subtype (Median Diff)" = "#ff7f0e")) +
  theme_minimal(base_size = 12)
```

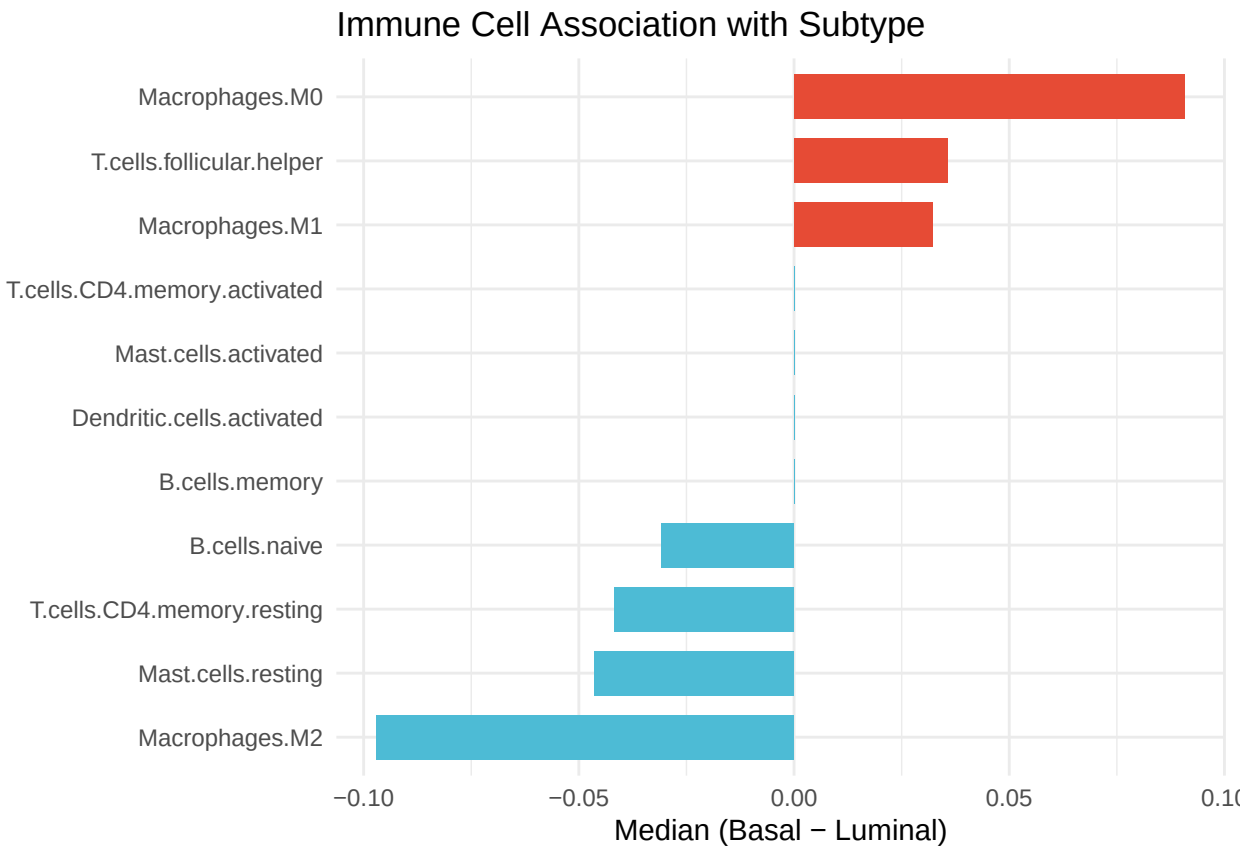


#Interpretations 1.Link ageing to innate immunity: The two strongest Δ-age correlates are mast-cell resting and macrophage M0 infiltration → age-accelerated tumours show a "wound-healing / innate" profile. 2. Highlight discordance: M1 macrophages lower Δ-age yet are slightly Luminal-skewed – evidence that not all Luminal features drive ageing in the same direction. 3. Prioritise features for the SVM / downstream

pathway analysis: immune cell types with consistent Δ-age and subtype signals (Mast cells, M0) are prime candidates.

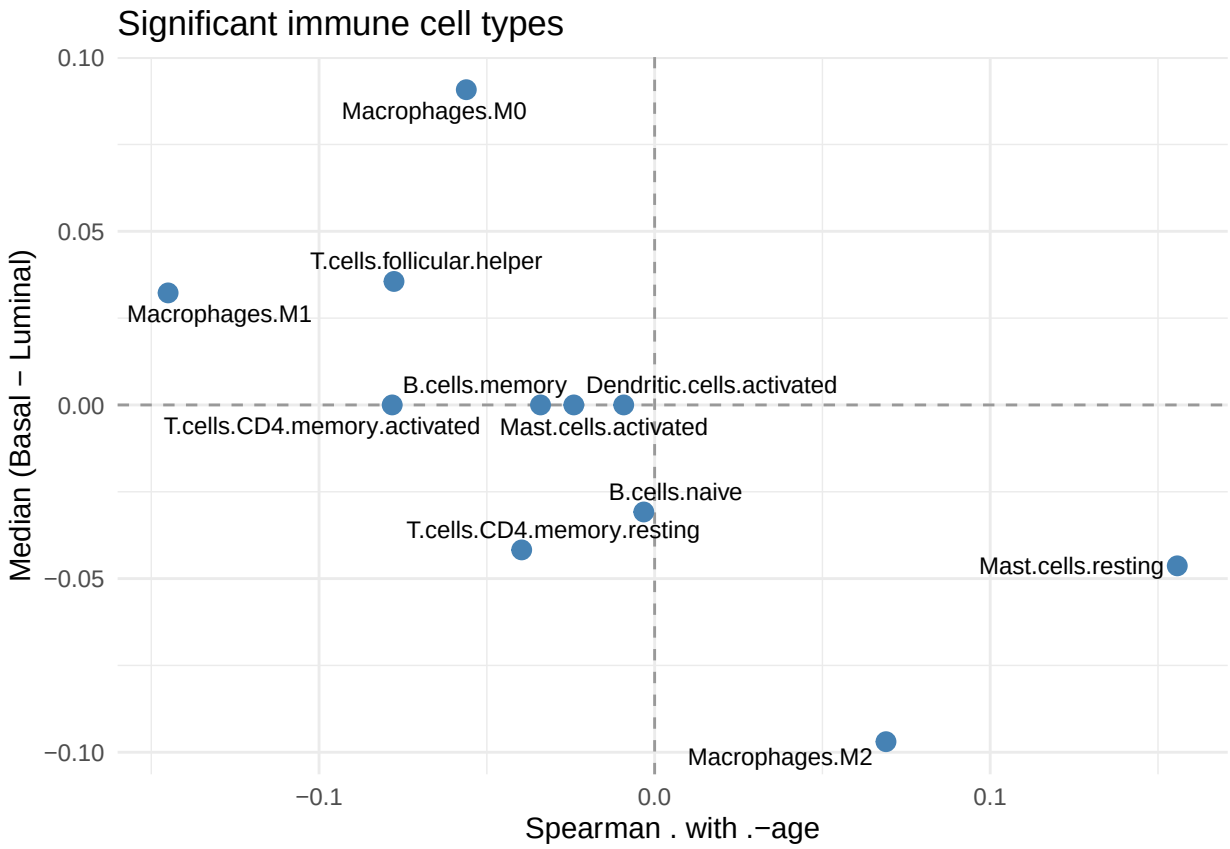
```
library(ggplot2)

ggplot(significant_fdr,
       aes(x = reorder(Cell_Type, Subtype_Median_Diff),
           y = Subtype_Median_Diff,
           fill = Subtype_Median_Diff > 0)) +
  geom_col(width = .7) +
  coord_flip() +
  scale_fill_manual(values = c("TRUE" = "#E64B35", "FALSE" = "#4DBBD5"),
                    guide  = "none") +
  labs(title   = "Immune Cell Association with Subtype",
       x       = NULL,
       y       = "Median (Basal - Luminal)") +
  theme_minimal()
```




```
ggplot(significant_fdr,
       aes(x = Corr_with_Delta,
           y = Subtype_Median_Diff,
           label = Cell_Type)) +
  geom_hline(yintercept = 0, linetype = "dashed", colour = "grey60") +
  geom_vline(xintercept = 0, linetype = "dashed", colour = "grey60") +
  geom_point(size = 3, colour = "steelblue") +
  ggrepel::geom_text_repel(size = 3) +
  labs(title = "Significant immune cell types",
       x = "Spearman  with Δ-age",
```

```
      y = "Median (Basal - Luminal)") +
  theme_minimal()
```




```
subtype_sig  = read.csv("/mnt/home/fayyaz/MPIB-SRT/1040-pelotas/private/data/autobids/subtype_sig")
```

#Immune cell correlation with Subtypes

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

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

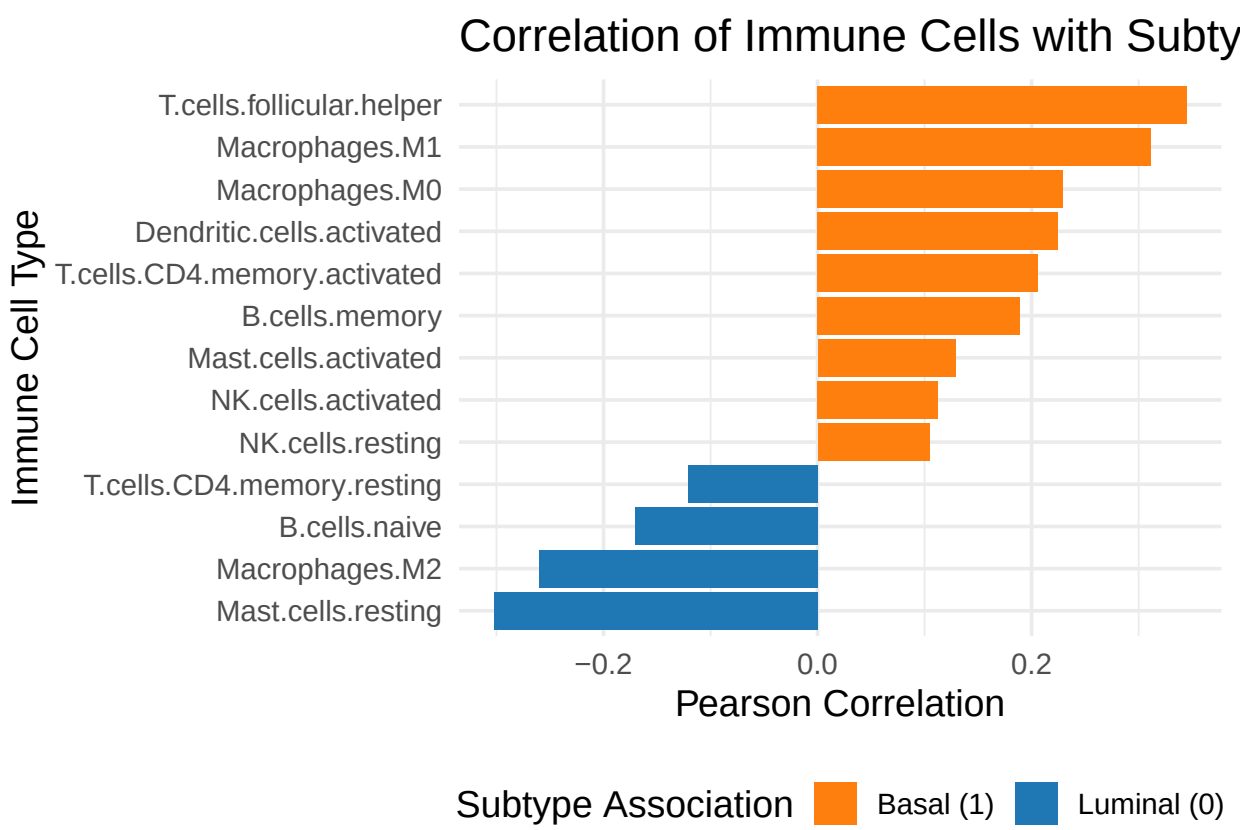
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
# Order Cell_Type by correlation for a nicer bar plot
subtype_sig$Cell_Type <- factor(subtype_sig$Cell_Type,
                                levels = subtype_sig$Cell_Type[order(subtype_sig$Correlation)])

# Plot bar graph
ggplot(subtype_sig, aes(x = Cell_Type, y = Correlation, fill = Subtype_Label)) +
  geom_bar(stat = "identity", position = "dodge") +
  coord_flip() +  # flip for horizontal bars
```

```
  scale_fill_manual(values = c("Basal (1)" = "#ff7f0e", "Luminal (0)" = "#1f77b4")) +
  labs(
    title = "Correlation of Immune Cells with Subtypes)",
    x = "Immune Cell Type",
    y = "Pearson Correlation",
    fill = "Subtype Association"
  ) +
  theme_minimal(base_size = 14) +
  theme(legend.position = "bottom")
```



##Interpretation

The result indicates that certain Immune cells such as Tcells follicular and Macrophages show high correlation with luminal subtypes. while Mast cell resting and Macrophages show high correlation with basal subtypes.

Immune features Comparison among raw data comparison and SHAP features

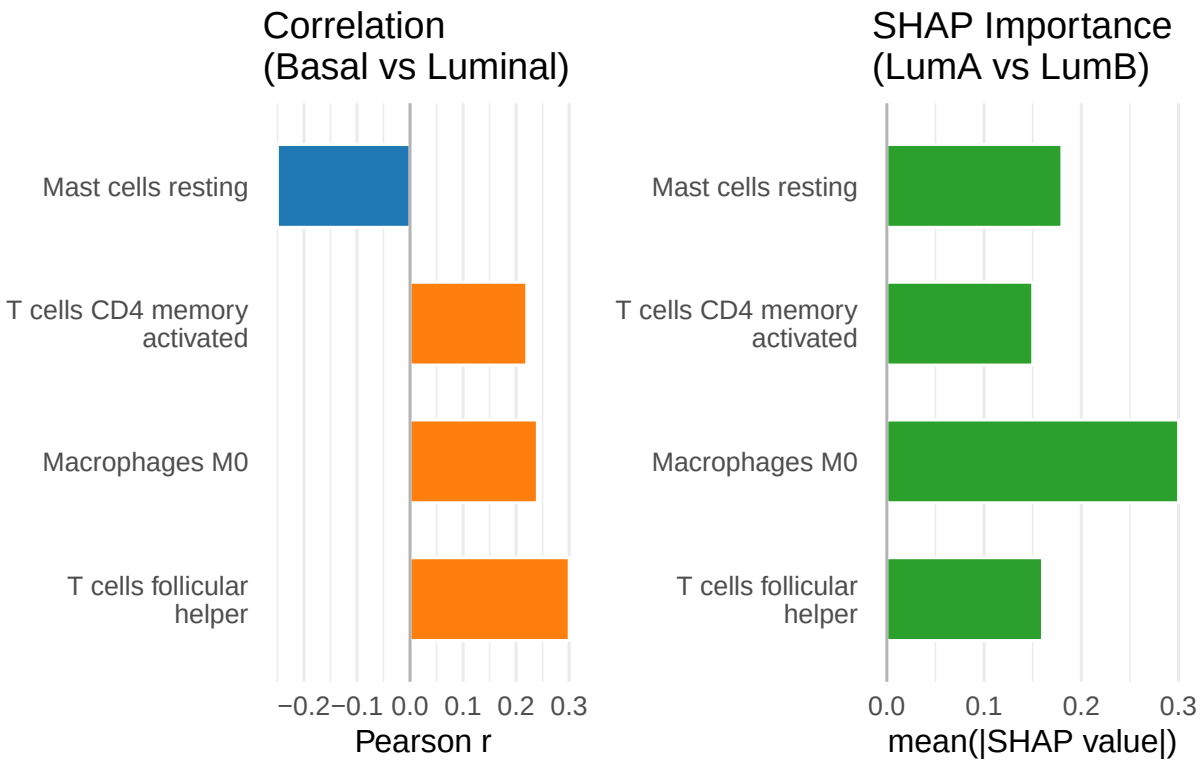
```
# Required libraries
library(ggplot2)
library(dplyr)
library(stringr)
library(patchwork)

# 1) Construct the data frame
```

```r
df <- tibble::tibble(
  cell         = c(
    "T cells follicular helper",
    "Macrophages M0",
    "Mast cells resting",
    "T cells CD4 memory activated"
  ),
  correlation  = c( 0.30,  0.24, -0.25,  0.22),
  shap_mean    = c( 0.16,  0.30,  0.18,  0.15)
) %>%
  # wrap long labels at ~20 chars
  mutate(label = str_wrap(cell, 20)) %>%
  # order by descending correlation
  arrange(desc(correlation)) %>%
  mutate(label = factor(label, levels = label),
         subtype = if_else(correlation >= 0, "Basal", "Luminal"))
```

```r
# Left panel: correlation bar plot
p_corr <- ggplot(df, aes(x = correlation, y = label, fill = subtype)) +
  geom_col(color = "white", width = 0.6) +
  scale_fill_manual(values = c(Basal = "#ff7f0e", Luminal = "#1f77b4")) +
  geom_vline(xintercept = 0, color = "gray70") +
  labs(
    title = "Correlation\n(Basal vs Luminal)",
    x     = "Pearson r",
    y     = NULL,
    fill  = "Subtype"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    legend.position = "none",
    panel.grid.major.y = element_blank()
  )
# Right panel: SHAP importance bar plot
p_shap <- ggplot(df, aes(x = shap_mean, y = label)) +
  geom_col(fill = "#2ca02c", color = "white", width = 0.6) +
  geom_vline(xintercept = 0, color = "gray70") +
  labs(
    title = "SHAP Importance\n(LumA vs LumB)",
    x     = "mean(|SHAP value|)",
    y     = NULL
  ) +
  theme_minimal(base_size = 12) +
  theme(
    panel.grid.major.y = element_blank()
  )

#  Combine panels with a shared title
(p_corr | p_shap) +
  plot_annotation(
    title = "Immune Features Driving Subtypes Separation",
    theme = theme(plot.title = element_text(size = 16, face = "bold", hjust = 0.5))
  )
```



#Interpretations

Left: how each immune cell correlates with Basal vs Luminal (orange positive, blue negative).

Right: how strongly each same cell drives the LumA vs LumB decision (green SHAP)

Featurea Annotation

```
# Install BiocManager if needed
if (!requireNamespace("BiocManager", quietly = TRUE)) {
  install.packages("BiocManager", quiet = TRUE)
}

# List all packages you need
pkgs <- c(
  "clusterProfiler",
  "org.Hs.eg.db",
  "ReactomePA",
  "AnnotationDbi",
  "dplyr",
  "tibble",
  "ggplot2"
)

# Install (no asking, no updating everything)
suppressWarnings(
  BiocManager::install(pkgs, ask = FALSE, update = FALSE, quiet = TRUE)
)
```

```
## Bioconductor version 3.20 (BiocManager 1.30.26), R 4.4.2 (2024-10-31)
```

```
suppressPackageStartupMessages({
  library(clusterProfiler)
  library(org.Hs.eg.db)
  library(ReactomePA)
  library(AnnotationDbi)
  library(dplyr)
  library(tibble)
  library(ggplot2)
})

sessionInfo()
```

```
## R version 4.4.2 (2024-10-31)
## Platform: x86_64-pc-linux-gnu
## Running under: Ubuntu 20.04.6 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3;  LAPACK version 3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Etc/UTC
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] tibble_3.3.0           ReactomePA_1.50.0      org.Hs.eg.db_3.20.0
##  [4] AnnotationDbi_1.68.0   IRanges_2.40.1         S4Vectors_0.44.0
##  [7] Biobase_2.66.0         BiocGenerics_0.52.0    clusterProfiler_4.14.6
## [10] patchwork_1.3.1        stringr_1.5.1          dplyr_1.1.4
## [13] forcats_1.0.0          ggplot2_3.5.2          reshape2_1.4.4
##
## loaded via a namespace (and not attached):
##   [1] DBI_1.2.3               gson_0.1.0              gridExtra_2.3
##   [4] rlang_1.1.6             magrittr_2.0.3          DOSE_4.0.1
##   [7] compiler_4.4.2          RSQLite_2.4.1           reactome.db_1.89.0
##  [10] png_0.1-8               vctrs_0.6.5             pkgconfig_2.0.3
##  [13] crayon_1.5.3            fastmap_1.2.0           XVector_0.46.0
##  [16] ggraph_2.2.1            labeling_0.4.3          rmarkdown_2.29
##  [19] enrichplot_1.26.6       graph_1.84.1            UCSC.utils_1.2.0
##  [22] tinytex_0.57            purrr_1.1.0             bit_4.6.0
##  [25] xfun_0.52               zlibbioc_1.52.0         cachem_1.1.0
##  [28] graphite_1.52.0         aplot_0.2.8             GenomeInfoDb_1.42.3
```

```
##  [31] jsonlite_2.0.0          blob_1.2.4              tweenr_2.0.3
##  [34] BiocParallel_1.40.2     parallel_4.4.2          R6_2.6.1
##  [37] stringi_1.8.7           RColorBrewer_1.1-3      GOSemSim_2.32.0
##  [40] Rcpp_1.1.0              knitr_1.50              ggtangle_0.0.7
##  [43] R.utils_2.13.0          Matrix_1.7-3            splines_4.4.2
##  [46] igraph_2.1.4            tidyselect_1.2.1        viridis_0.6.5
##  [49] qvalue_2.38.0           rstudioapi_0.17.1       yaml_2.3.10
##  [52] codetools_0.2-20        lattice_0.22-7          plyr_1.8.9
##  [55] treeio_1.30.0           withr_3.0.2             KEGGREST_1.46.0
##  [58] evaluate_1.0.4          gridGraphics_0.5-1      polyclip_1.10-7
##  [61] Biostrings_2.74.1       pillar_1.11.0           BiocManager_1.30.26
##  [64] ggtree_3.14.0           ggfun_0.1.9             generics_0.1.4
##  [67] scales_1.4.0            tidytree_0.4.6          glue_1.8.0
##  [70] lazyeval_0.2.2          tools_4.4.2             data.table_1.17.8
##  [73] fgsea_1.32.4            graphlayouts_1.2.2      fs_1.6.6
##  [76] tidygraph_1.3.1         fastmatch_1.1-6         cowplot_1.2.0
##  [79] grid_4.4.2              tidyr_1.3.1             ape_5.8-1
##  [82] nlme_3.1-168            GenomeInfoDbData_1.2.13 ggforce_0.5.0
##  [85] cli_3.6.5               rappdirs_0.3.3          viridisLite_0.4.2
##  [88] gtable_0.3.6            R.methodsS3_1.8.2       yulab.utils_0.2.0
##  [91] digest_0.6.37           ggrepel_0.9.6           ggplotify_0.1.2
##  [94] farver_2.1.2            memoise_2.0.1           htmltools_0.5.8.1
##  [97] R.oo_1.27.1             lifecycle_1.0.4         httr_1.4.7
## [100] GO.db_3.20.0            MASS_7.3-65             bit64_4.6.0-1
```

Features extracted from SHAP

```
# 1) Ensembl ID vector
features_all <- c(
  "ENSG00000054598","ENSG00000186832","ENSG0000005513",
  "ENSG00000198729","ENSG00000186868","ENSG00000154548",
  "ENSG00000259793","ENSG00000172425","ENSG00000254615",
  "ENSG00000082175","ENSG00000204385","ENSG00000148513",
  "ENSG00000236313","ENSG00000135912","ENSG00000234918"
)


features_sig <- features_all
```

```
is_ensembl <- function(x) grepl("^ENSG\\d{11}$", x)

bad_all <- features_all[!is_ensembl(features_all)]
if (length(bad_all)) {
  message("Dropping malformed Ensembl IDs (ALL): ",
          paste(bad_all, collapse = ", "))
}
```

```
## Dropping malformed Ensembl IDs (ALL): ENSG0000005513
```

```
features_all <- features_all[is_ensembl(features_all)]

if (length(features_sig)) {
  bad_sig <- features_sig[!is_ensembl(features_sig)]
  if (length(bad_sig)) {
```

```
    message("Dropping malformed Ensembl IDs (SIG): ",
            paste(bad_sig, collapse = ", "))
  }
  features_sig <- features_sig[is_ensembl(features_sig)]
}
```

```
## Dropping malformed Ensembl IDs (SIG): ENSG0000005513
```

```
length(features_all)
```

```
## [1] 14
```

```
length(features_sig)
```

```
## [1] 14
```

```
map_all <- bitr(features_all,
                fromType = "ENSEMBL",
                toType   = "ENTREZID",
                OrgDb    = org.Hs.eg.db)
```

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

```
## Warning in bitr(features_all, fromType = "ENSEMBL", toType = "ENTREZID", :
## 21.43% of input gene IDs are fail to map...
```

```
entrez_all <- unique(map_all$ENTREZID)
length(entrez_all)
```

```
## [1] 11
```

```
if (length(features_sig)) {
  map_sig <- bitr(features_sig,
                  fromType = "ENSEMBL",
                  toType   = "ENTREZID",
                  OrgDb    = org.Hs.eg.db)
  entrez_sig <- unique(map_sig$ENTREZID)
  length(entrez_sig)
} else {
  entrez_sig <- character(0)
}
```

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

```
## Warning in bitr(features_sig, fromType = "ENSEMBL", toType = "ENTREZID", :
## 21.43% of input gene IDs are fail to map...
```

```
## [1] 11
```

```
ego_all <- enrichGO(
  gene          = entrez_all,
  OrgDb         = org.Hs.eg.db,
  keyType       = "ENTREZID",
  ont           = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff  = 1,
  qvalueCutoff  = 1,
  readable      = TRUE
)
```

```r
ekegg_all <- enrichKEGG(
  gene          = entrez_all,
  organism      = "hsa",
  keyType       = "ncbi-geneid",
  pvalueCutoff  = 1,
  qvalueCutoff  = 1
)
```

```
## Reading KEGG annotation online: "https://rest.kegg.jp/link/hsa/pathway"...
```

```
## Reading KEGG annotation online: "https://rest.kegg.jp/list/pathway/hsa"...
```

```
## Reading KEGG annotation online: "https://rest.kegg.jp/conv/ncbi-geneid/hsa"...
```

```r
# Make gene IDs readable (map Entrez back to gene symbols)
ekegg_all <- setReadable(ekegg_all, OrgDb = org.Hs.eg.db, keyType = "ENTREZID")
```

```r
ereact_all <- enrichPathway(
  gene          = entrez_all,
  organism      = "human",
  pvalueCutoff  = 1,
  qvalueCutoff  = 1,
  readable      = TRUE
)
```

```r
# Load the stringr library for text wrapping
library(stringr)

# --- Enhanced Bar Plot Function ---
safe_barplot <- function(enrich_result, title, showCategory = 20) {
  # Check if there's anything to plot
  if (!is.null(enrich_result) && nrow(as.data.frame(enrich_result)) > 0) {
    p <- barplot(enrich_result, showCategory = showCategory) +
      labs(
        title = title,
        x = "Gene Ratio",
        y = "Enriched Term"
      ) +
      # Wrap long labels on the y-axis to 40 characters
      scale_y_discrete(labels = function(x) str_wrap(x, width = 40)) +
      theme_minimal(base_size = 14) +
      theme(plot.title = element_text(hjust = 0.5)) # Center the title

    print(p)
  } else {
    message("No terms to plot for: ", title)
  }
}
```

```r
# Filter each result for adjusted p-value < 0.05
ego_all_df <- as.data.frame(ego_all)
ekegg_all_df <- as.data.frame(ekegg_all)
ereact_all_df <- as.data.frame(ereact_all)

ego_sig <- ego_all_df %>% filter(p.adjust < 0.05)
ekegg_sig <- ekegg_all_df %>% filter(p.adjust < 0.05)
```

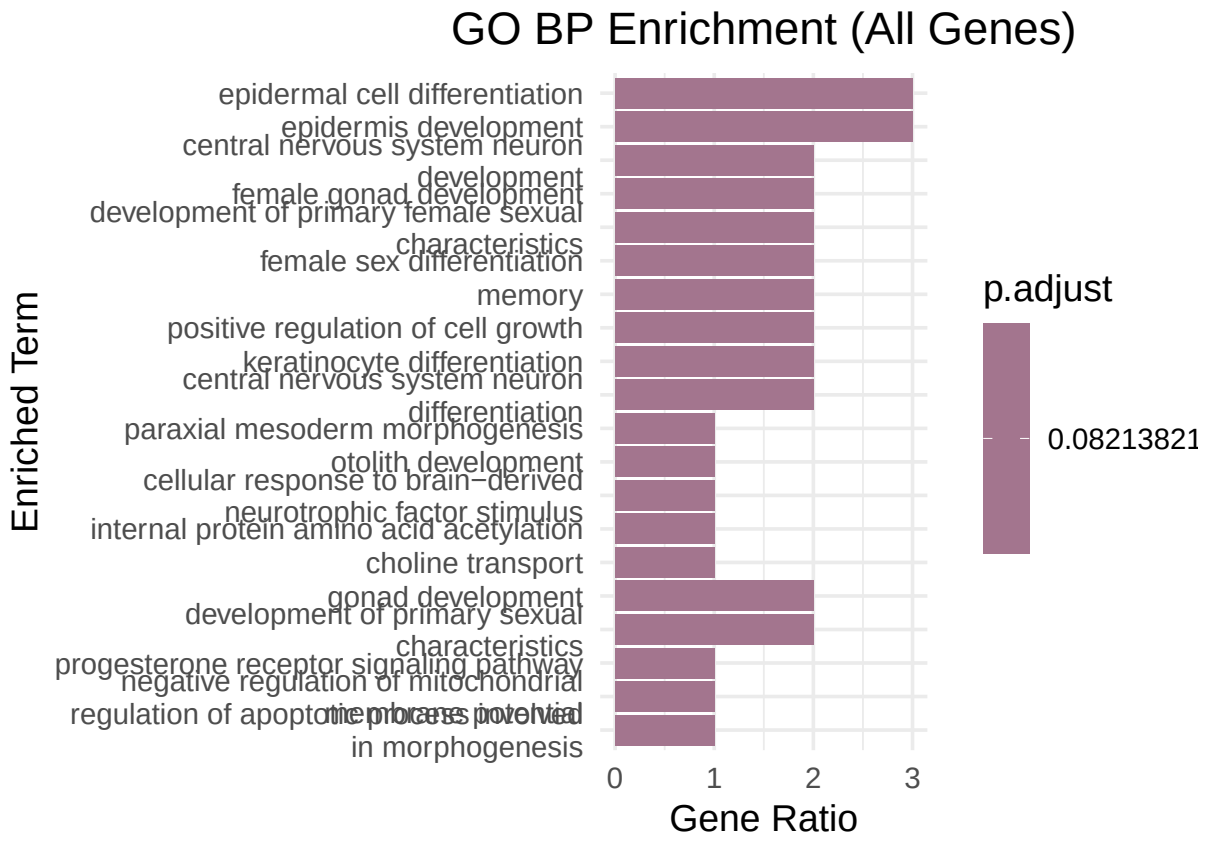
```
ereact_sig <- ereact_all_df %>% filter(p.adjust < 0.05)

# See how many significant GO terms you have
print(paste("Found", nrow(ego_sig), "significant GO terms."))
```

```
## [1] "Found 0 significant GO terms."
```

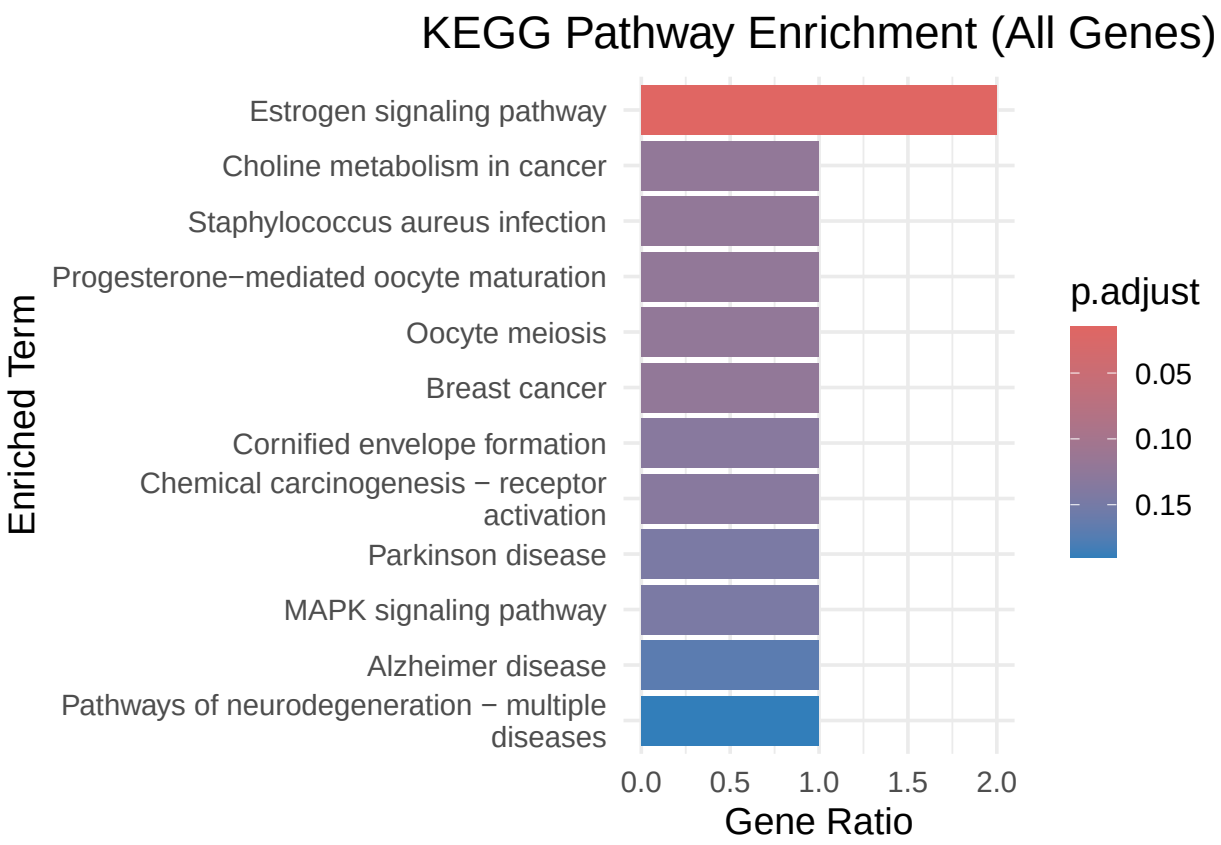
```
safe_barplot(ego_all, "GO BP Enrichment (All Genes)")
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```



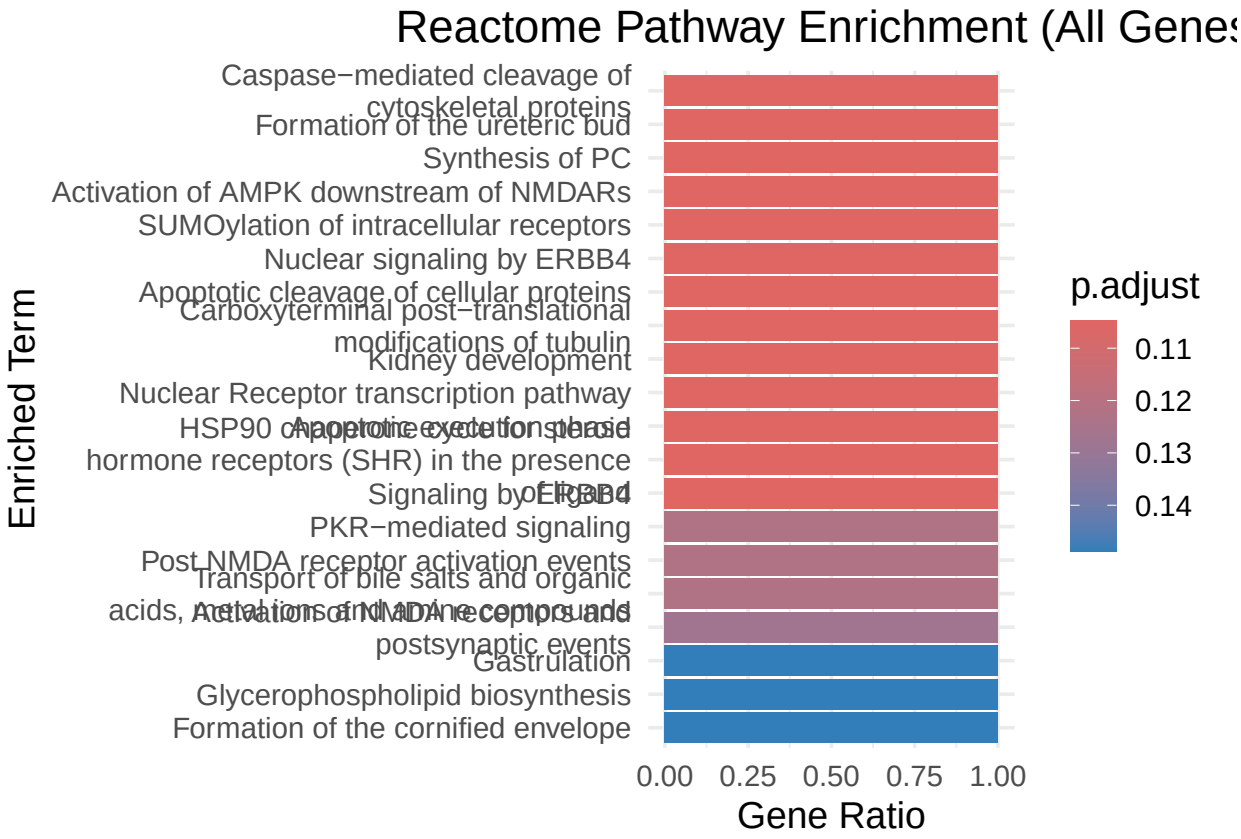

```
safe_barplot(ekegg_all, "KEGG Pathway Enrichment (All Genes)")
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```




```
safe_barplot(ereact_all, "Reactome Pathway Enrichment (All Genes)")
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```


Reactome Pathway Enrichment (All Genes
Enriched Term
Caspase−mediated cleavage of cytoskeletal proteins
Formation of the ureteric bud
Synthesis of PC
Activation of AMPK downstream of NMDARs
SUMOylation of intracellular receptors
Nuclear signaling by ERBB4
Apoptotic cleavage of cellular proteins
Carboxyterminal post−translational modifications of tubulin
Kidney development
Nuclear Receptor transcription pathway
HSP90 chaperone cycle for steroid hormone receptors (SHR) in the presence
Signaling by ERBB4
PKR−mediated signaling
Post NMDA receptor activation events
Transport of bile salts and organic acids, metal ions and amine compounds
Activation of NMDA receptors and postsynaptic events
Gastrulation
Glycerophospholipid biosynthesis
Formation of the cornified envelope
0.00
0.25
0.50
0.75
1.00
Gene Ratio
p.adjust
0.11
0.12
0.13
0.14