kegg

charvi khanna

2025-09-08

```
# Load required libraries
library(STRINGdb)
library(org.Hs.eg.db)
```

```
## Loading required package: AnnotationDbi
```

```
## Loading required package: stats4
```

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

```
## Loading required package: generics
```

```
##
## Attaching package: 'generics'
```

```
## The following objects are masked from 'package:base':
##
##     as.difftime, as.factor, as.ordered, intersect, is.element, setdiff,
##     setequal, union
```

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

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

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

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

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

## Loading required package: IRanges

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

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

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

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##     windows

##
```

```
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following object is masked from 'package:IRanges':
##
##     union

## The following object is masked from 'package:S4Vectors':
##
##     union

## The following objects are masked from 'package:BiocGenerics':
##
##     normalize, path, union

## The following objects are masked from 'package:generics':
##
##     components, union
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
library(ggraph)
```

```
## Loading required package: ggplot2
```

```
library(clusterProfiler)
```

```
##

## clusterProfiler v4.16.0 Learn more at https://yulab-smu.top/contribution-knowledge-mining/
##
## Please cite:
##
## S Xu, E Hu, Y Cai, Z Xie, X Luo, L Zhan, W Tang, Q Wang, B Liu, R Wang,
## W Xie, T Wu, L Xie, G Yu. Using clusterProfiler to characterize
## multiomics data. Nature Protocols. 2024, 19(11):3292-3320

##
## Attaching package: 'clusterProfiler'

## The following object is masked from 'package:igraph':
##
##     simplify

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

## The following object is masked from 'package:IRanges':
##
##     slice

## The following object is masked from 'package:S4Vectors':
##
##     rename

## The following object is masked from 'package:stats':
##
##     filter
```

```
library(BSgenome)
```

```
## Loading required package: GenomeInfoDb
```

```
## Loading required package: GenomicRanges

## Loading required package: Biostrings

## Loading required package: XVector

##
## Attaching package: 'Biostrings'

## The following object is masked from 'package:base':
##
##     strsplit

## Loading required package: BiocIO

## Loading required package: rtracklayer

##
## Attaching package: 'rtracklayer'

## The following object is masked from 'package:igraph':
##
##     blocks
```

```r
# Define protein list
my_proteins <- data.frame(protein = c("TP53", "BRCA1", "MDM2"))

# Initialize STRING database (Human = 9606, v12, score cutoff = 400)
string_db <- STRINGdb$new(
  version = "12",
  species = 9606,
  score_threshold = 400
)

# Map protein symbols to STRING IDs
mapped <- string_db$map(
  my_proteins,
  "protein",
  removeUnmappedRows = TRUE
)

# Get PPI interactions
ppi <- string_db$get_interactions(mapped$STRING_id)

# Relabel STRING IDs with gene symbols
lookup <- setNames(mapped$protein, mapped$STRING_id)
ppi$from_name <- lookup[ppi$from]
ppi$to_name   <- lookup[ppi$to]

# Clean interactions (remove NAs)
ppi_clean <- na.omit(ppi)
```

```r
# Build igraph object
g <- graph_from_data_frame(
  ppi_clean[, c("from_name", "to_name")],
  directed = FALSE
)

# Visualize network
ggraph(g, layout = "fr") +
  geom_edge_link(alpha = 0.5) +
  geom_node_point(size = 6, color = "steelblue") +
  geom_node_text(aes(label = name), repel = TRUE, size = 4) +
  theme_void() +
  ggtitle("Protein-Protein Interaction Network")
```