Package ‘KEGGemUP’

June 23, 2026

Title Creating, Rendering, and Mapping Omics Data on KEGG Pathway Graphs

Version 0.99.1

Description KEGGemUP retrieves and renders interactively KEGG pathway graphs. The retrieval makes full use of the caching functionality to avoid unnecessary download processes. The interactivity of the graph is warranted via the visNetwork interface to the vis.js library, fully supporting bindings to be used in Shiny for further operations.

License MIT + file LICENSE

Encoding UTF-8

VignetteBuilder knitr

biocViews Software, Pathways, KEGG, Network, Visualization, DifferentialExpression, GraphAndNetwork, Transcriptomics, Proteomics, Metabolomics, SystemsBiology

Roxygen list(markdown = TRUE)

URL `https://github.com/imbeimainz/KEGGemUP`

BugReports `https://github.com/imbeimainz/KEGGemUP/issues`

Depends R (>= 4.6.0)

Imports BiocFileCache, httr2, ggplot2, rlang, methods, igraph, scales, KEGGREST, visNetwork, RColorBrewer, htmlwidgets, xml2, grDevices, stats, tools, utils

Suggests knitr, rmarkdown, BiocStyle, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

git_url https://git.bioconductor.org/packages/KEGGemUP

git_branch devel

git_last_commit b905e7a

git_last_commit_date 2026-06-10

Repository Bioconductor 3.24

Date/Publication 2026-06-22

Author Edoardo Filippi [aut] (ORCID: <https://orcid.org/0009-0003-9858-0137>),
Myriam Meineck [ctb] (ORCID: <https://orcid.org/0000-0002-3243-2540>),
Paul Classen [ctb] (ORCID: <https://orcid.org/0000-0002-9410-0525>),
Julia Weinmann-Menke [fnd] (ORCID:
<https://orcid.org/0000-0001-7344-8381>),
Federico Marini [aut, cre] (ORCID:
<https://orcid.org/0000-0003-3252-7758>)

Maintainer Federico Marini <marinif@uni-mainz.de>

Contents

`cleanup_title_node` *Cleanup the title node from a KEGG igraph object*

Description

Cleanup the title node from a KEGG igraph object

Usage

```
cleanup_title_node(g)
```

Arguments

g	The igraph object

Details

Strips away the node starting with the `TITLE:` label. In some situations, that node might not be too heavily needed...

Value

An igraph object, without the node originally kept as "TITLE:..."

Examples

```
graph <- create_kegg_graph(pathway_id = "hsa04110")
graph_notitlenode <- cleanup_title_node(graph)
graph_notitlenode

render_kegg_graph(graph_notitlenode)
```

`create_kegg_graph` *Create a graph for a KEGG pathway*

Description

Convert KEGG pathway to igraph object

Usage

```
create_kegg_graph(pathway_id, kgml_file = NULL, verbose = FALSE)
```

Arguments

`pathway_id`	Character, KEGG pathway ID (e.g., 'hsa04110') or NULL if using kgml_file
`kgml_file`	Path to a local KGML file (optional, if pathway_id is provided)
`verbose`	Logical indicating whether to print verbose messages (default: FALSE)

Value

An igraph object representing the KEGG pathway graph

Examples

```
pathway <- "hsa04110" # Example pathway ID
graph <- create_kegg_graph(pathway_id = pathway, verbose = TRUE)
plot(graph) # Plot the graph using igraph's plotting functions
```

`display_cache_KEGGemUP` *Return all cached KEGGemUP information*

Description

Return all cached KEGG and mapping files from BiocFileCache

Usage

```
display_cache_KEGGemUP()
```

Details

This function retrieves information about all cached KEGG pathway files and mapping files stored using BiocFileCache.

Value

A list containing data frames of cached KEGG and mapping files

Examples

```
cache_info <- display_cache_KEGGemUP()
# View cached KEGG pathway files
print(cache_info$kegg)
# View cached mapping files
print(cache_info$mappings)
```

`export_kegg_graph` *Export a KEGG graph*

Description

Exporting a KEGG graph into its components, nodes and edges, as tab-separated text files (having them represented as dataframes for max portability)

Usage

```
export_kegg_graph(g, basename)
```

Arguments

`g`	An igraph graph object, e.g. created with KEGGemUP
`basename`	Character string, specifying the base name for the files to write the two individual data frames, for nodes and edges

Value

NULL, invisibly

Examples

```
g <- create_kegg_graph(pathway_id = "hsa04110")
export_kegg_graph(g, basename = tempfile())
```

`get_kegg_db` *Get a KEGG database file*

Description

Get KEGG database, with caching.

Usage

```
get_kegg_db(db_name = "compound", path = NULL, bfc = NULL, verbose = FALSE)
```

Arguments

`db_name`	Name of the KEGG database to retrieve (default: "compound").
`path`	Optional path to save the KEGG database file if not using cache. Defaults to NULL.
`bfc`	BiocFileCache object for caching KEGG database files. Defaults to NULL
`verbose`	Logical, if TRUE, print additional messages.

Details

The valid KEGG database names are: kegg | pathway | brite | module | ko | genes | | vg | vp | ag | genome | ligand | compound | glycan | reaction | rclass | enzyme | network | variant | disease | drug | dgroup

Value

A data frame with KEGG IDs and names.

Examples

```
# Saving in path
data_dir <- tempdir()
kegg_compounds <- get_kegg_db(
  db_name = "compound",
  path = data_dir, verbose = TRUE
)

# Just returning without saving
kegg_compounds_onthefly <- get_kegg_db(
  db_name = "compound",
  verbose = TRUE
)
head(kegg_compounds_onthefly)

# saving to cache (in a temp dir)
kegg_compounds_cached <- get_kegg_db(
  db_name = "compound",
  bfc = BiocFileCache::BiocFileCache(tempdir()),
  verbose = TRUE
)
```

`highlight_kegg_graph` *Highlight a subset of the KEGG graph*

Description

Highlight a subset of the graph based on KEGG IDs

Usage

```
highlight_kegg_graph(g, ids_to_highlight)
```

Arguments

`g`	An igraph object to visualize. Must have vertex attributes 'x' and 'y' for layout.
`ids_to_highlight`	Character vector of KEGG IDs to include in the highlighted subset.

Details

This function highlights the nodes corresponding to the provided KEGG IDs and fades the rest of the graph. It modifies vertex attributes to achieve this effect.

Value

An igraph object with highlighted nodes and faded non-highlighted nodes and edges.

Examples

```
pathway <- "mmu00230"
g <- create_kegg_graph(pathway)
KEGG_to_include <- c("C00262", "C00385", "C00366", "C00294", "C00387",
                     "C01762", "C05512", "C00301", "C01185", "C00455",
                     "22436", "14544", "18950", "11486", "80285", "59027")
highlighted_subg <- highlight_kegg_graph(g, KEGG_to_include)
```

`KEGGemUP-pkg` *KEGGemUP*

Description

`KEGGemUP` is a Bioconductor package that provides a modern, interactive way to create, visualize, and explore KEGG pathways

Details

`KEGGemUP` spices up the KEGG pathways! Not just to simplify their exploration, but also to map your data and results onto the nodes and edges that constitute a KEGG pathway. `KEGGemUP` provides a cache-compatible approach to retrieve the underlying KGML files and avoid unnecessary download operations. Converting the KGML files into full igraph objects and embedding them into the visNetwork framework really brings your pathways to life, providing tooltips and additional information to be used e.g. in analysis reports or in fully fledged Shiny apps.

Author(s)

Maintainer: Federico Marini <`marinif@uni-mainz.de`> (ORCID)

Authors:

- Federico Marini <`marinif@uni-mainz.de`> (ORCID)
- Edoardo Filippi <`edoardo.filippi@uni-mainz.de`> (ORCID)

Other contributors:

- Myriam Meineck <`Myriam.Meineck@unimedizin-mainz.de`> (ORCID) [contributor]

- Paul Classen <`classenp@uni-mainz.de`> (ORCID) [contributor]
- Julia Weinmann-Menke <`Julia.Weinmann-Menke@unimedizin-mainz.de`> (ORCID) [funder]

See Also

Useful links:

- `https://github.com/imbeimainz/KEGGemUP`
- Report bugs at `https://github.com/imbeimainz/KEGGemUP/issues`

`map_results_to_graph` *Map continuous values to graph nodes*

Description

Map differential expression results to nodes

Usage

```
map_results_to_graph(
  g,
  de_results,
  feature_column = NULL,
  value_column = NULL,
  verbose = FALSE,
  palette = NULL,
  palette_limit = NULL,
  palettes_list = list(NA_character_),
  palettes_limits_list = c(NA_real_)
)
```

Arguments

`g`	An igraph object representing the KEGG pathway graph.
`de_results`	A data frame or a list of data frames containing differential expression results
`feature_column`	Name of the column in de_results that contains KEGG IDs
`value_column`	Name of the column in de_results that contains values to map
`verbose`	Logical indicating whether to print verbose messages (default: FALSE)
`palette`	Optional color palette for mapping values (default: NULL, will use a default palette)
`palette_limit`	Optional numeric limit for the color palette (default: NULL, will be determined from data)
`palettes_list`	Optional list of color palettes if de_results is a list (default: NULL)
`palettes_limits_list`	Optional list of numeric limits for multiple palettes if de_results is a list (default: NULL)

Details

This function can be used to map the differential expression results to the graph, the input of the graph must be the output of the function `create_kegg_graph` in the igraph format. The results to be mapped can be provided either as a list or as a single data.frame. If a single data.frame is provided, the default column names that it will look for are KEGG IDs and values are 'KEGG_ids' and 'log2FoldChange', respectively, but these can be changed using the `feature_column` and `value_column` parameters.

Value

An igraph object with differential expression results mapped to node attributes

Examples

```
pathway <- "hsa04110" # Example pathway ID
graph <- create_kegg_graph(pathway_id = pathway)
# Example differential expression results
de_results <- data.frame(
  KEGG_ids = c("hsa:1234", "hsa:5678", "cpd:C00022"),
  log2FoldChange = c(1.5, -2.0, 0.5)
)
vis_graph <- map_results_to_graph(
  graph,
  de_results,
  feature_column = "KEGG_ids",
  value_column = "log2FoldChange"
)
```

`render_kegg_graph` *Create an interactive visualization of KEGG pathways with visNetwork*

Description

Plot KEGG pathway graph using visNetwork

Usage

```
render_kegg_graph(
  g,
  graph_title = NULL,
  scaling_factor = 1.5,
  relationships = c("all", "reactions", "relations", "none"),
  visualization_type = c("standard", "positions", "node_name", "node_size")
)
```

Arguments

`g`	An igraph object representing the KEGG pathway graph.
`graph_title`	Character string, used as a title for the rendered graph. Defaults to NULL, which would fall back to the name specified in the title attribute of the graph.
`scaling_factor`	Numeric factor to scale node sizes (default: 1.5).

`relationships`	Character specifying which relationships to include in edges ("all", "reactions", "relations", "none"; default: "all").
`visualization_type`	Character specifying the type of visualization for nodes: "standard", "positions", "node_name", or "node_size" (default: "standard").

Details

This function takes an igraph object representing a KEGG pathway graph and creates a visNetwork visualization. It maps node attributes to visual properties.

Value

A visNetwork object representing the KEGG pathway graph with mapped results.

Examples

```
pathway <- "hsa04110" # Example pathway ID
graph <- create_kegg_graph(pathway_id = pathway)
# Example differential expression results
de_results <- data.frame(
  KEGG_ids = c("hsa:1234", "hsa:5678", "cpd:C00022"),
  log2FoldChange = c(1.5, -2.0, 0.5)
)
graph <- map_results_to_graph(
  graph,
  de_results,
  feature_column = "KEGG_ids",
  value_column = "log2FoldChange")

vis_graph <- render_kegg_graph(graph, scaling_factor = 1.5,
relationships = "all", visualization_type = "standard")
```

`reset_cache_KEGGemUP` *Reset all KEGGemUP caches*

Description

Reset the KEGG and mapping caches by deleting all cached files

Usage

```
reset_cache_KEGGemUP()
```

Details

This function deletes all cached KEGG pathway files and mapping files stored using BiocFileCache. It prompts the user for confirmation before proceeding with the deletion.

Value

(invisible) NULL

Examples

```
## Not run:
  reset_cache_KEGGemUP()

## End(Not run)
```

`res_de_macro_IFNg_vs_naive` *A sample* `data.frame` *containing Differential Expression Analysis, generated with* `limma`

Description

A sample `data.frame` containing Differential Expression Analysis, generated with `limma`

Format

A `data.frame` object

Details

This `data.frame` object contains the results of a Differential Expression Analysis performed (with `limma`) on data from the `macrophage` package, contrasting the counts from naive macrophage to those associated with IFNg.

The code to create said object can be found in the folder `/inst/scripts` in the KEGGemUP package, the file is called `create_deresults.R`.

Value

A sample `data.frame` object, extracted from the `topTable` function as a result of running the `limma` DE workflow

References

Alasoo, et al. "Shared genetic effects on chromatin and gene expression indicate a role for enhancer priming in immune response", Nature Genetics, January 2018 doi: 10.1038/s41588-018-0046-7.

`retrieve_all_pathways` *Download all pathways*

Description

Download all KEGG pathways for a given organism

Usage

```
retrieve_all_pathways(org, wait = 0.5, verbose = FALSE)
```

Arguments

`org`	KEGG organism code (e.g., 'hsa' for human).
`wait`	Numeric value, needs to be strictly positive. Indicates the amount in seconds to wait in between requests, being polite and respectful of the limit rates imposed by KEGG. Defaults to 0.5, which is safely a bit above the rate of 3max/sec.
`verbose`	Logical, if TRUE, print additional messages.

Details

This function can take a while to run completely, but is an efficient way to retrieve all the kgml files encoding for the KEGG pathways. Some failures could be triggered by a rate limitation imposed by the KEGG website. Since this caches the files locally, it is safe to rerun to complete the operation without extra unneeded requests to the API

Value

The BiocFileCache object is returned invisibly

Examples

```
# Download all pathways for human
## Not run:
  retrieve_all_pathways("hsa", verbose = TRUE)

## End(Not run)
```

`retrieve_kgml` *Download a KGML file*

Description

Download and cache KEGG KGML files.

Usage

```
retrieve_kgml(pathway_id, bfc = NULL, path = NULL, verbose = FALSE)
```

Arguments

`pathway_id`	Character, KEGG pathway ID (e.g., 'hsa04110').
`bfc`	BiocFileCache object for caching KEGG KGML files. Defaults to NULL
`path`	Optional path to save the KGML file if not using cache. Defaults to NULL
`verbose`	Logical, if TRUE, print additional messages.

Details

This function can save an individual KGML to the cache or to a path. If specified, the BiocFileCache cache is prioritized. If a `path` if specified and it is a directory, it will save the file there. If the path specified a file path, this will be the location to store the file. If it is left as NULL, it will save the KMGL file in the current working directory

Value

Path to the cached KGML file.

Examples

```
data_dir <- tempdir()
kgml_path <- retrieve_kgml("hsa04110", path = data_dir, verbose = TRUE)

cache_dir <- tempdir()
kgml_path_cached <- retrieve_kgml("hsa04110",
  bfc = BiocFileCache::BiocFileCache(cache_dir),
  verbose = TRUE
)
```

`subset_kegg_graph` *Subset a graph object for a KEGG pathway*

Description

Create igraph visualization with improved layout

Usage

```
subset_kegg_graph(g, ids_to_include)
```

Arguments

`g`	An igraph object to visualize. Must have vertex attributes 'x' and 'y' for layout.
`ids_to_include`	Character vector of KEGG IDs to include in the subset graph.

Details

All the edges between the vertices are plotted automatically.

Value

A plot of the igraph object with improved layout.

Examples

```
pathway <- "mmu00230"
g <- create_kegg_graph(pathway)
KEGG_to_include <- c("C00262", "C00385", "C00366", "C00294", "C00387",
                 "C01762", "C05512", "C00301", "C01185", "C00455",
                 "22436", "14544", "18950", "11486", "80285", "59027")
subg <- subset_kegg_graph(g, KEGG_to_include)
# plot(g)
# plot(subg)
```

Index