Package 'punKEGGer'

October 29, 2026

Title Parse, Expand, Annotate and Visualize KEGG Pathway Networks as Tidy Graphs

Version 0.99.65

Description Provides tools to parse KEGG pathway KGML files into tidy graphs, expand metanodes (e.g., multi-gene entries), annotate them with custom dictionaries, and produce consistent visualizations using ggraph. Enables integration of KEGG pathways with tidygraph-based workflows.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.6.0)

Imports dplyr, ggplot2, ggraph, glue, igraph, purrr, rlang, stringr, tibble, tidygraph, tidyselect, tidyr, KEGGREST, xml2

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

VignetteBuilder knitr

biocViews Pathways, GraphAndNetwork, Software, Network, Visualization, KEGG

URL https://github.com/guillermodeandajauregui/punKEGGer

BugReports https://github.com/guillermodeandajauregui/punKEGGer/issues

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

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

git_branch devel

git_last_commit 68a8e2d

git_last_commit_date 2026-07-10

Repository Bioconductor 3.24

Date/Publication 2026-10-28

Author Guillermo de Anda Jáuregui [aut, cre]

Maintainer Guillermo de Anda Jáuregui <gdeanda@inmegen.edu.mx>

Contents

add_kegg_layout . . . 2
annotate_kegg_graph . . . 3
combine_kegg_network . . . 4
create_meta_dict . . . 4
expand_metagraph . . . 5
extract_kegg_nodes . . . 6
fetch_kegg_kgml . . . 6
layout_kegg . . . 7
parse_kegg_groups . . . 7
parse_kegg_relations_clean . . . 8
plot_kegg_graph . . . 8

Index **10**

`add_kegg_layout` *Add KEGG layout coordinates to a tidygraph*

Description

Joins a KEGG layout table (as returned by `layout_kegg()`) to a tidygraph object, using the `meta_id` field of each node as the key.

Usage

```
add_kegg_layout(graph, layout_tbl)
```

Arguments

`graph`	A `tidygraph` object, where each node has a `meta_id`.
`layout_tbl`	A tibble with layout coordinates: `id`, `x`, `y`.

Value

The same `tidygraph`, with `x` and `y` coordinates added to nodes.

Examples

```
kgml_file <- system.file("extdata", "hsa04210.xml", package = "punKEGGer")
doc <- xml2::read_xml(kgml_file)

g <- combine_kegg_network(doc)
layout_tbl <- layout_kegg(doc)
g_layout <- add_kegg_layout(g, layout_tbl)

# View the layout coordinates
head(tidygraph::as_tibble(g_layout))
```

annotate_kegg_graph *Annotate a KEGG tidygraph with custom identifiers*

Description

Adds external identifiers (e.g., HGNC symbols, Ensembl IDs) to the nodes of a tidygraph-style KEGG network, using a custom dictionary.

Usage

```
annotate_kegg_graph(graph, kegg_dict, identifiers = c("hgnc_symbol"))
```

Arguments

graph	A tidygraph object (nodes should be named with KEGG IDs).
kegg_dict	A tibble with one column kegg_id, plus any other IDs (e.g. hgnc_symbol, ensembl_id, etc.).
identifiers	Character vector of column names in kegg_dict to attach (default: "hgnc_symbol").

Details

Note: You must provide your own annotation dictionary (kegg_dict) as a tibble with at least one column named kegg_id, containing KEGG gene IDs in the form "hsa:1234". This dictionary can be generated from BioMart or other curated sources.

Value

The same tidygraph, with new columns added to the node attributes.

Examples

```
kgml_file <- system.file("extdata", "hsa04210.xml", package = "punKEGGer")
dict_file <- system.file("extdata", "example_dict.hsa04210.csv", package = "punKEGGer")

doc <- xml2::read_xml(kgml_file)
dict <- read.csv(dict_file)
nodes <- extract_kegg_nodes(doc)
meta_dict <- create_meta_dict(nodes, dict)
g <- combine_kegg_network(doc)
g_annot <- annotate_kegg_graph(g, meta_dict)
head(tidygraph::as_tibble(g_annot))
```

combine_kegg_network *Combine KEGG group and relation edges into a tidygraph*

Description

Parses both standard KEGG relations and intra-group edges, and builds a directed tidygraph. Also assigns meta_id to each node: either its group_id (if in a group) or its own ID.

Usage

```
combine_kegg_network(kgml)
```

Arguments

kgml	A parsed XML document from `xml2::read_xml()`.

Value

A `tidygraph` object with all edges from the KGML and nodes carrying `meta_id`.

Examples

```
kgml_file <- system.file("extdata", "hsa04210.xml", package = "punKEGGer")
doc <- xml2::read_xml(kgml_file)
g <- combine_kegg_network(doc)
print(g)
igraph::gorder(g)
```

create_meta_dict *Create a meta_id-to-annotation dictionary*

Description

Takes the output of `extract_kegg_nodes()` and a KEGG annotation dictionary (e.g., from BioMart or custom source), and merges them based on `kegg_id`. This enables annotating each metanode (`meta_id`) with HGNC, Ensembl, Entrez, etc.

Usage

```
create_meta_dict(node_info, kegg_dict)
```

Arguments

node_info	A tibble from `extract_kegg_nodes()` with `meta_id` and `kegg_id` columns.
kegg_dict	A tibble mapping KEGG gene IDs to external identifiers. Must contain a column `kegg_id` with KEGG-formatted IDs (e.g., `"hsa:1234"`).

Value

A tibble linking each `meta_id` to any identifiers in `kegg_dict`.

Examples

```
kgml_file <- system.file("extdata", "hsa04210.xml", package = "punKEGGer")
dict_file <- system.file("extdata", "example_dict.hsa04210.csv", package = "punKEGGer")

doc <- xml2::read_xml(kgml_file)
nodes <- extract_kegg_nodes(doc)
dict <- read.csv(dict_file)
meta_dict <- create_meta_dict(nodes, dict)
head(meta_dict)
```

---

expand_metagraph *Expand a KEGG tidygraph by unrolling multi-gene metanodes*

Description

Takes a tidygraph where node names are KGML entry IDs and expands the network by replacing group/multi-gene nodes with individual gene-level edges.

Usage

```
expand_metagraph(g, node_info, node_types = c("gene"))
```

Arguments

g	A tidygraph object, typically created by combine_kegg_network().
node_info	A tibble as returned by extract_kegg_nodes().
node_types	Character vector indicating which node types to expand (default is "gene").

Value

A new tidygraph object with expanded node IDs (e.g., hsa:00001, hsa:00002).

Examples

```
# Parse KGML and extract nodes and relations
kgml_file <- system.file("extdata", "hsa04210.xml", package = "punKEGGer")
doc <- xml2::read_xml(kgml_file)
node_info <- extract_kegg_nodes(doc)
rels <- parse_kegg_relations_clean(doc)
g <- combine_kegg_network(doc)

# Expand the graph from metanodes to gene-level nodes
g_exp <- expand_metagraph(g, node_info)

# Compare number of nodes before and after expansion
igraph::gorder(g)      # number of metanodes
igraph::gorder(g_exp)  # number of gene-level nodes
```

extract_kegg_nodes *Extract KEGG node information from a KGML file*

Description

Parses all `entry` elements from a KEGG KGML XML document and extracts ID, name, type, and individual KEGG IDs. Supports multi-gene entries (e.g., `hsa:1234+hsa:5678`) by expanding them into separate rows with a shared `meta_id`.

Usage

```
extract_kegg_nodes(kgml)
```

Arguments

`kgml`	A parsed XML document from `xml2::read_xml()`.

Value

A tibble with columns: `id`, `meta_id`, `type`, `kegg_id`

Examples

```
kgml_file <- system.file("extdata", "hsa04210.xml", package = "punKEGGer")
doc <- xml2::read_xml(kgml_file)
nodes <- extract_kegg_nodes(doc)
head(nodes)
```

fetch_kegg_kgml *Fetch a KEGG pathway KGML via KEGGREST*

Description

Downloads a KEGG pathway in KGML format using `KEGGREST::keggGet()` and returns it as an `xml2` document ready for downstream parsing with `punKEGGer`.

Usage

```
fetch_kegg_kgml(pathway_id)
```

Arguments

`pathway_id`	A KEGG pathway identifier, e.g. `"hsa04210"`.

Value

An `xml2` document.

Examples

```
kgml <- fetch_kegg_kgml("hsa04210")
g <- combine_kegg_network(kgml)
```

layout_kegg *Extract node layout coordinates from a KEGG KGML file*

Description

Parses the graphics element of each node in the KGML to extract manual layout positions (x, y), which are typically defined for KEGG maps.

Usage

```
layout_kegg(kgml)
```

Arguments

kgml	A parsed XML document from xml2::read_xml().

Value

A tibble with columns: id, x, y (corresponding to KEGG entry IDs).

Examples

```
kgml_file <- system.file("extdata", "hsa04210.xml", package = "punKEGGer")
doc <- xml2::read_xml(kgml_file)
layout_tbl <- layout_kegg(doc)
head(layout_tbl)
```

parse_kegg_groups *Parse KEGG group nodes from a KGML file*

Description

Extracts group-type entries from a KEGG KGML file and generates both: (1) a mapping of group ID to component node IDs (with meta_id), (2) a set of intra-group edges linking the components as undirected pairs.

Usage

```
parse_kegg_groups(kgml)
```

Arguments

kgml	A parsed XML document from xml2::read_xml(), representing a KEGG KGML file.

Value

A list with two elements:

mapping A tibble mapping each member ID to its meta_id (group_id or itself).

edges A tibble with pairwise intra-group edges (from, to, type = "group").

Examples

```
kgml_file <- system.file("extdata", "hsa04210.xml", package = "punKEGGer")
doc <- xml2::read_xml(kgml_file)
group_data <- parse_kegg_groups(doc)
head(group_data$mapping)
head(group_data$edges)
```

`parse_kegg_relations_clean` *Parse clean KEGG relations from a KGML file*

Description

Extracts pairwise relations between KEGG entries and their subtypes if available.

Usage

```
parse_kegg_relations_clean(kgml)
```

Arguments

`kgml`	A parsed XML document from `xml2::read_xml()`.

Value

A tibble with columns: `from`, `to`, `type`, `subtype`.

Examples

```
kgml_file <- system.file("extdata", "hsa04210.xml", package = "punKEGGer")
doc <- xml2::read_xml(kgml_file)
rels <- parse_kegg_relations_clean(doc)
head(rels)
```

`plot_kegg_graph` *Visualize a KEGG pathway tidygraph using manual layout*

Description

Generates a `ggraph` plot of a KEGG network with manual layout (x, y), colored by node type and edge subtype. The user can specify which identifier to display on each node label (e.g., `hgnc_symbol`).

Usage

```
plot_kegg_graph(g, id = hgnc_symbol)
```

Arguments

g	A tidygraph object, where nodes contain x, y, type, and a label column (e.g. hgnc_symbol).
id	Unquoted column name to use for node labels (e.g., hgnc_symbol, kegg_id). Default: hgnc_symbol.

Value

A ggplot object with the KEGG network visualization.

Examples

```
kgml_file <- system.file("extdata", "hsa04210.xml", package = "punKEGGer")
dict_file <- system.file("extdata", "example_dict.hsa04210.csv", package = "punKEGGer")

kgml <- xml2::read_xml(kgml_file)
g <- combine_kegg_network(kgml)

node_info <- extract_kegg_nodes(kgml)
dict <- read.csv(dict_file)
meta_dict <- create_meta_dict(node_info, dict)

g_exp <- expand_metagraph(g, node_info, node_types = "gene")
g_annot <- annotate_kegg_graph(g_exp, meta_dict, identifiers = "hgnc_symbol")

layout_tbl <- layout_kegg(kgml)
g_layout <- add_kegg_layout(g_annot, layout_tbl)

plot_kegg_graph(g_layout, id = hgnc_symbol)
```

Index

add_kegg_layout, 2
annotate_kegg_graph, 3

combine_kegg_network, 4
create_meta_dict, 4

expand_metagraph, 5
extract_kegg_nodes, 6

fetch_kegg_kgml, 6

layout_kegg, 7

parse_kegg_groups, 7
parse_kegg_relations_clean, 8
plot_kegg_graph, 8