

Package ‘mechgraph’

September 21, 2026

Title Mechanism Evidence Graph Data Model

Version 0.0.1

Author Guangchuang Yu [aut, cre]

Maintainer Guangchuang Yu <guangchuangyu@gmail.com>

Description Provides a lightweight graph data model for representing, combining, querying, and summarizing evidence-backed biological mechanism graphs. A mechgraph is an S3 list holding a node table, an edge table, and provenance metadata. The package implements builders that convert STRING and BioGRID interaction tables into mechgraph objects, combiners (mg_bind(), mg_combine()) that merge graphs while preserving duplicate evidence records, accessors (mg_nodes(), mg_edges(), mg_metadata()) and mutators (mg_add_*, mg_drop_*) for node and edge tables, filters by type, source, identifier, and score, induced-subgraph construction, structural validation (mg_validate()), and quality-control summaries (mg_qc()).
Szkarczyk et al. (2023) <doi:10.1093/nar/gkac1000>
Oughtred et al. (2021) <doi:10.1002/pro.3938>.

License MIT + file LICENSE

Encoding UTF-8

Imports igraph

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2026-09-21 18:00:08 UTC

Contents

as.igraph.mechgraph	2
mg_add_nodes	3

mg_bind	3
mg_filter_nodes	4
mg_from_biogrid	5
mg_from_edges	6
mg_from_string	7
mg_graph	9
mg_nodes	10
mg_qc	11
mg_validate	11
Index	13

as.igraph.mechgraph	<i>Convert a mechgraph object to an igraph object</i>
---------------------	---

Description

Convert a mechgraph object (an S3 list with nodes, edges and metadata) to an igraph object, preserving node and edge columns as vertex/edge attributes.

Usage

```
## S3 method for class 'mechgraph'
as.igraph(x, ...)
```

Arguments

- x a mechgraph object.
- ... additional arguments (currently ignored).

Value

An igraph object. Node columns become vertex attributes (id, type, taxon_id, ...), the display label is set as the vertex name, and size is set to the degree. Edge columns (type, source, evidence, n_evidence, ...) become edge attributes. Parallel (multi) edges are preserved; call `igraph::simplify()` to collapse them.

mg_add_nodes	<i>Modify mechgraph nodes and edges</i>
--------------	---

Description

Add or remove nodes and edges while preserving the mechgraph table schema.

Usage

```
mg_add_nodes(x, nodes)
```

```
mg_add_edges(x, edges, add_missing_nodes = FALSE)
```

```
mg_drop_nodes(x, nodes)
```

```
mg_drop_edges(x, edges)
```

Arguments

x	A mechgraph object.
nodes	A node table, node identifiers, or rows to remove depending on the function.
edges	An edge table, numeric row indices, or logical row mask.
add_missing_nodes	Whether to create unknown nodes for edge endpoints missing from x.

Value

A new mechgraph object with the modified node and/or edge tables; the input graph is left unchanged. `mg_add_nodes()` and `mg_add_edges()` row-bind the new records onto the existing tables (with `add_missing_nodes = TRUE`, `mg_add_edges()` creates type = "unknown" nodes for endpoints absent from the graph). `mg_drop_nodes()` removes the given nodes and every edge incident to them; `mg_drop_edges()` removes the given edges (a data frame of from/to/type rows, numeric row indices, or a logical mask) while keeping all nodes.

mg_bind	<i>Combine mechgraph objects</i>
---------	----------------------------------

Description

Combine graphs while preserving duplicate evidence records by default.

Usage

```
mg_bind(...)
```

```
mg_combine(..., merge_nodes = TRUE, merge_edges = FALSE)
```

Arguments

...	Mechgraph objects, or a list of mechgraph objects.
merge_nodes	Reserved for API compatibility. Nodes are currently merged by identifier.
merge_edges	Whether to drop duplicate edges with the same from, to, and type.

Value

A new mechgraph object. `mg_bind()` row-binds the edge tables of the input graphs (duplicate evidence records are preserved), merges the node tables by identifier (duplicated node ids are dropped), and records the input metadata in a list under `metadata$inputs`. `mg_combine()` additionally removes duplicate edges sharing the same from, to, and type when `merge_edges = TRUE`.

<code>mg_filter_nodes</code>	<i>Filter mechgraph objects</i>
------------------------------	---------------------------------

Description

Filter node and edge tables and return a new mechgraph object.

Usage

```
mg_filter_nodes(x, type = NULL, ids = NULL, source = NULL)

mg_filter_edges(x, type = NULL, source = NULL, score = NULL, weight = NULL)

mg_induced_subgraph(x, nodes)
```

Arguments

x	A mechgraph object.
type	Optional node or edge type filter.
ids	Optional node identifier filter.
source	Optional source filter.
score	Optional score threshold or range.
weight	Optional weight threshold or range.
nodes	Node identifiers or a node table defining the induced subgraph.

Value

A new mechgraph object with the filtered node and/or edge tables and the metadata of the input graph. `mg_filter_nodes()` and `mg_induced_subgraph()` keep only the selected nodes and the edges whose both endpoints are among them; `mg_filter_edges()` keeps the selected edges and leaves the node table unchanged.

mg_from_biogrid	<i>Build a mechgraph from BioGRID</i>
-----------------	---------------------------------------

Description

Download, parse, and convert BioGRID physical interaction tables into a mechgraph object.

Usage

```
mg_biogrid_versions()

mg_biogrid_file_url(version = "latest", dataset = c("mv_physical", "all"),
  format = c("tab3", "mitab"))

mg_biogrid_download(version = "latest", dataset = c("mv_physical", "all"),
  format = c("tab3", "mitab"), cache = TRUE, destdir = NULL, quiet = TRUE)

mg_biogrid_parse(file, format = c("tab3", "mitab"), taxon_id = NULL,
  collapse = TRUE)

mg_from_biogrid(file = NULL, version = "latest", dataset = "mv_physical",
  format = c("tab3", "mitab"), taxon_id = NULL,
  collapse = TRUE, cache = TRUE)
```

Arguments

version	BioGRID version string, or "latest".
dataset	BioGRID dataset. The MVP supports "mv_physical" and "all".
format	Download or parse format. The MVP supports "tab3" and "mitab".
cache	Whether to reuse an existing cached download.
destdir	Optional cache directory. Defaults to a session-scoped directory under tempdir(); pass an explicit path for persistent caching.
quiet	Reserved for API compatibility. Downloads are handled by yulab.utils::mydownload().
file	A local BioGRID table or zip file. If missing, mg_from_biogrid() downloads BioGRID first.
taxon_id	Optional NCBI taxonomy identifier used to filter both interactors.
collapse	Whether to collapse duplicate (undirected) edge records between the same pair of interactors. TRUE (default) keeps one row per unique pair, aggregates the evidence-type columns, and adds a count column n_evidence; FALSE keeps one row per BioGRID evidence record.

Value

- `mg_biogrid_versions()` returns a character vector of supported BioGRID version labels ("latest").
- `mg_biogrid_file_url()` returns the download URL as a character string.
- `mg_biogrid_download()` returns the path to the downloaded (or cached) file as a character string.
- `mg_biogrid_parse()` returns a list with components `nodes` and `edges`, the two `data.frames` to be passed to `mg_from_edges()`.
- `mg_from_biogrid()` returns a `mechgraph` object with `type = "ppi"` edges carrying the BioGRID evidence annotation (`evidence`, `evidence_type`, `pmid`) and provenance metadata (`source`, `version`, `dataset`, `format`, `taxon_id`, `URL`, `license`).

mg_from_edges

*Build a mechgraph from an edge list***Description**

Create a `mechgraph` from a plain edge list. If `nodes` is missing, nodes are inferred from edge endpoints.

Usage

```
mg_from_edges(edges, nodes = NULL, metadata = list())
```

```
as_mechgraph(x, ...)
```

Arguments

<code>edges</code>	A data frame with required columns <code>from</code> , <code>to</code> , and <code>type</code> .
<code>nodes</code>	Optional node table with required columns <code>id</code> , <code>type</code> , and <code>label</code> .
<code>metadata</code>	A named list of graph metadata.
<code>x</code>	An object to coerce.
<code>...</code>	Additional arguments passed to methods.

Value

`mg_from_edges()` returns a `mechgraph` object (an S3 list with components `nodes`, `edges`, and `metadata`). When `nodes` is missing, a node table with `type = "unknown"` is inferred from the unique edge endpoints. `as_mechgraph()` returns the coerced object: for a `data.frame` it returns `mg_from_edges(x, ...)` and for a `mechgraph` it returns the object unchanged.

mg_from_string

*Build a mechgraph from STRING***Description**

Download, parse, and convert STRING functional-association networks into a mechgraph object with type = "ppi" edges carrying the STRING combined score and provenance metadata (source, version, taxon_id, network, score threshold, keytype, URL, license). mg_string_species() returns the STRING species list for a release.

Usage

```
mg_string_versions()

mg_string_species(version = "12.0", cache = TRUE)

mg_string_file_url(taxon_id = 9606, version = "12.0",
  network = c("functional", "physical", "detailed", "aliases", "info"))

mg_string_download(taxon_id = 9606, version = "12.0",
  network = c("functional", "physical", "detailed", "aliases", "info"),
  cache = TRUE, destdir = NULL)

mg_string_parse(links, aliases = NULL, info = NULL, taxon_id = 9606,
  version = "12.0", score_threshold = 400,
  keytype = c("entrez", "symbol", "uniprot", "ensembl_gene",
    "ensembl_protein", "string_id"), labels = TRUE)

mg_from_string(file = NULL, taxon_id = 9606, version = "12.0",
  network = c("functional", "physical"), score_threshold = 400,
  keytype = c("entrez", "symbol", "uniprot", "ensembl_gene",
    "ensembl_protein", "string_id"), labels = TRUE, cache = TRUE)
```

Arguments

taxon_id	NCBI taxonomy id (e.g. 9606 for human).
version	STRING release version (e.g. "12.0").
network	Which STRING table: "functional" (protein.links), "physical" (protein.physical.links), "detailed" (protein.links.detailed), "aliases" (protein.aliases) or "info" (protein.info). mg_from_string() accepts "functional" (default) and "physical".
cache	Whether to reuse an existing cached download.
destdir	Optional cache directory. Defaults to a session-scoped directory under tempdir(); pass an explicit path for persistent caching.
links	A local STRING links table path or data.frame (columns protein1, protein2, combined_score).

aliases	A local protein.aliases table path or data.frame (columns #string_protein_id, alias, source). Required unless keytype is "ensembl_protein" or "string_id".
info	A local protein.info table path or data.frame. Used for preferred-name node labels when labels = TRUE.
score_threshold	Minimum combined_score (0-1000) to keep an edge; NULL keeps all edges.
keytype	Node identifier type: "entrez" (default), "symbol", "uniprot", "ensembl_gene", "ensembl_protein" (Ensembl protein id without the species prefix) or "string_id" (full STRING id).
labels	Logical, use STRING preferred names as node labels when info is available; otherwise labels equal node ids.
file	A local STRING links table. If missing, mg_from_string() downloads STRING first.

Details

STRING protein ids ("9606.ENSEP...") are mapped to the requested identifier type through the STRING protein.aliases table. For keytype = "entrez" the Ensembl_HGNC_entrez_id aliases cover roughly 98% of human STRING proteins. Edges below score_threshold are dropped; self-loops (which can appear after identifier collapse, e.g. isoforms mapping to one gene) and duplicate undirected edges (highest score kept) are removed.

The full human protein.links table is large (~1.5 GB uncompressed); the download is cached, but parsing it requires roughly that much free memory.

The identifier mapping currently assumes the human (9606) protein.aliases table: the Ensembl_HGNC_* alias sources used by "entrez", "symbol", "uniprot" and "ensembl_gene" exist only for human. For other species use keytype = "ensembl_protein" or "string_id" (identity mappings) until per-species alias sources are added.

Value

- mg_string_versions() returns a character vector of supported STRING release versions ("12.0").
- mg_string_species() returns a data.frame of STRING species for the requested release (columns taxon_id, STRING_type, STRING_name_compact, official_name_NCBI, domain).
- mg_string_file_url() returns the download URL as a character string.
- mg_string_download() returns the path to the downloaded (or cached) file as a character string.
- mg_string_parse() returns a list with components nodes and edges, the two data.frames to be passed to mg_from_edges().
- mg_from_string() returns a mechgraph object with type = "ppi" edges carrying the STRING combined_score and provenance metadata (source, version, taxon_id, network, score threshold, keytype, URL, license).

Examples

```
## Build a mechgraph from a small STRING links table (no download)
links <- data.frame(
  protein1 = c("9606.ENSP00000000233", "9606.ENSP00000000412"),
  protein2 = c("9606.ENSP00000000412", "9606.ENSP00000001008"),
  combined_score = c(800, 700),
  stringsAsFactors = FALSE
)
mg <- mg_from_string(file = links, keytype = "string_id", labels = FALSE)
mg
mg_edges(mg)

## mg_from_string(), mg_string_download() and mg_string_species() download
## STRING tables and therefore need network access. Only the species table
## is small enough to fetch here: building the human network would pull
## protein.links, which is ~80 MB compressed and ~1.5 GB uncompressed.
species <- mg_string_species()
head(species)
species[species$taxon_id == 9606, ]

## For a real download-to-graph run, pick a species with a small network
## and use an identity keytype so no aliases/info tables are fetched, e.g.
##   mg_from_string(taxon_id = 416591, keytype = "string_id",
##                 labels = FALSE)
```

mg_graph

*Create and inspect mechgraph objects***Description**

Create the lightweight S3 list object used by mechgraph.

Usage

```
mg_graph(nodes, edges, metadata = list(), validate = TRUE)
```

```
mg_empty(metadata = list())
```

```
is_mechgraph(x)
```

Arguments

nodes	A base data frame with required columns id, type, and label.
edges	A base data frame with required columns from, to, and type.
metadata	A named list of graph metadata.
validate	Whether to validate the object before returning it.
x	An object to test.

Value

mg_graph() returns a mechgraph object: an S3 list with components nodes (a data.frame with columns id, type, label), edges (a data.frame with columns from, to, type), and metadata (a named list). When validate = TRUE the object is checked with mg_validate() before being returned. mg_empty() returns such an object with empty node and edge tables. is_mechgraph() returns TRUE if x inherits from class "mechgraph" and FALSE otherwise.

mg_nodes	<i>Access mechgraph tables and metadata</i>
----------	---

Description

Return the node table, edge table, metadata list, or unique source labels from a mechgraph object.

Usage

```
mg_nodes(x)
```

```
mg_edges(x)
```

```
mg_metadata(x)
```

```
mg_sources(x)
```

Arguments

x A mechgraph object.

Value

- mg_nodes() returns the node table, a data.frame with the required columns id, type, label plus any additional node columns.
- mg_edges() returns the edge table, a data.frame with the required columns from, to, type plus any additional edge columns.
- mg_metadata() returns the metadata list stored in the graph.
- mg_sources() returns a sorted character vector of the unique non-missing source labels found in the node and edge tables.

`mg_qc`*Summarize mechgraph quality-control metrics*

Description

Return lightweight graph summary metrics including node and edge counts, isolated node count, and type/source distributions.

Usage`mg_qc(x)`**Arguments**

`x` A mechgraph object.

Value

A named list with components:

- `n_nodes`: total number of nodes.
- `n_edges`: total number of edges.
- `n_isolated_nodes`: number of nodes that are not an endpoint of any edge.
- `node_type_distribution`: a table of node types.
- `edge_type_distribution`: a table of edge types.
- `edge_source_distribution`: a table of edge source labels (empty table when the edge table has no source column).

`mg_validate`*Validate a mechgraph object*

Description

Checks the required node and edge schema and verifies that edge endpoints are present in `nodes$id`.

Usage`mg_validate(x)`**Arguments**

`x` A mechgraph object.

Value

Invisible TRUE when x is a valid mechgraph object. An error is raised (with a message describing the problem) when x is not a mechgraph, the node or edge tables are missing required columns, node ids are missing, empty, or duplicated, or edge endpoints reference nodes that are not in nodes\$id.

Index

`as.igraph.mechgraph`, [2](#)
`as_mechgraph (mg_from_edges)`, [6](#)

`is_mechgraph (mg_graph)`, [9](#)

`mg_add_edges (mg_add_nodes)`, [3](#)
`mg_add_nodes`, [3](#)
`mg_bind`, [3](#)
`mg_biogrid_download (mg_from_biogrid)`, [5](#)
`mg_biogrid_file_url (mg_from_biogrid)`, [5](#)
`mg_biogrid_parse (mg_from_biogrid)`, [5](#)
`mg_biogrid_versions (mg_from_biogrid)`, [5](#)
`mg_combine (mg_bind)`, [3](#)
`mg_drop_edges (mg_add_nodes)`, [3](#)
`mg_drop_nodes (mg_add_nodes)`, [3](#)
`mg_edges (mg_nodes)`, [10](#)
`mg_empty (mg_graph)`, [9](#)
`mg_filter_edges (mg_filter_nodes)`, [4](#)
`mg_filter_nodes`, [4](#)
`mg_from_biogrid`, [5](#)
`mg_from_edges`, [6](#)
`mg_from_string`, [7](#)
`mg_graph`, [9](#)
`mg_induced_subgraph (mg_filter_nodes)`, [4](#)
`mg_metadata (mg_nodes)`, [10](#)
`mg_nodes`, [10](#)
`mg_qc`, [11](#)
`mg_sources (mg_nodes)`, [10](#)
`mg_string_download (mg_from_string)`, [7](#)
`mg_string_file_url (mg_from_string)`, [7](#)
`mg_string_parse (mg_from_string)`, [7](#)
`mg_string_species (mg_from_string)`, [7](#)
`mg_string_versions (mg_from_string)`, [7](#)
`mg_validate`, [11](#)