Package: gson (via r-universe)

October 9, 2026

Title Base Class and Methods for 'gson' Format

Version 0.2.2

Description Provides a lightweight container and exchange format for gene set collections. A 'GSON' object stores which genes belong to which gene set, together with gene set and gene names, the identifier types in use, species, versions and source metadata. A collection can be built from data frames, read from and written to the 'gson' JavaScript Object Notation (JSON) format and the 'GMT' format, subset by gene set, merged across sources, validated, and resolved to the web addresses of the databases it comes from, so that a collection gathered by one package can be analysed by another.

Imports jsonlite, methods, stats, utils, yulab.utils (>= 0.0.7)

Suggests testthat (>= 3.0.0)

ByteCompile true

License Artistic-2.0

URL https://yulab-smu.top/biomedical-knowledge-mining-book/

BugReports https://github.com/YuLab-SMU/gson/issues

Encoding UTF-8

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake make libuv1-dev

Repository https://yulab-smu.r-universe.dev

Date/Publication 2026-10-09 12:11:13 UTC

RemoteUrl https://github.com/yulab-smu/gson

RemoteRef HEAD

RemoteSha e7863f2f4be94bb10f32f3cd6ef7efce405e7ad6

Contents

[.GSON . . . 2
as.data.frame.GSON . . . 3
gson . . . 3
GSON-class . . . 4
gson_union . . . 5
gson_url . . . 6
gsonList . . . 7
read.gmt . . . 7
read.gson . . . 8
show . . . 8
validate_gson . . . 9

Index **10**

`[.GSON` *Subset a GSON object*

Description

`x[i]` keeps the gene sets selected by `i` and returns a `GSON`, with the three mapping tables pruned together: `gene2name` is filtered by the genes that are still a member of a kept gene set, so the subset cannot grow a reference to a gene that is no longer there. The metadata slots carry over unchanged. `x[[i]]` returns the genes of one gene set.

Usage

```
## S3 method for class 'GSON'
x[i, j, ..., drop = FALSE]

## S3 method for class 'GSON'
x[[i, ...]]
```

Arguments

`x`	A `GSON` object.
`i`	Gene set IDs, or a logical or numeric index into `names(x)`.
`j`	Unused. Supplying a second index is an error; a `GSON` is only subset by gene sets.
`...`	Unused. `x[]` returns `x`.
`drop`	Ignored; a `GSON` has no dimension to drop.

Value

`[.GSON` returns a `GSON`, `[[.GSON` a character vector of genes.

as.data.frame.GSON *Coerce GSON to a data frame*

Description

Coerce GSON to a data frame

Usage

```
## S3 method for class 'GSON'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	A GSON object.
row.names	Unused.
optional	Unused.
...	Unused.

Value

A data frame with gene set-gene memberships and optional names.

gson *construct a 'GSON' object*

Description

construct a 'GSON' object

Usage

```
gson(
  gsid2gene,
  gsid2name = NULL,
  gene2name = NULL,
  schema_version = "1.0",
  species = NULL,
  gsname = NULL,
  version = NULL,
  accessed_date = NULL,
  keytype = NULL,
  gsidtype = NULL,
  urlpattern = NULL,
  info = NULL
)
```

Arguments

gsid2gene	A data frame with first column of gene set IDs and second column of genes
gsid2name	A data frame with first column of gene set IDs and second column of gene set names
gene2name	A data frame with first column of genes and second column of gene symbols
schema_version	GSON file schema version
species	Which species of the genes belongs to
gsname	Name of the gene set (e.g., GO, KEGG, etc.)
version	version of the gene set
accessed_date	date to obtain the gene set data
keytype	keytype of genes
gsidtype	identifier domain of the gene set IDs, e.g., GO, KEGG, REAC
urlpattern	URL pattern
info	extra information

Value

A 'GSON' instance

Examples

```
wpfile <- system.file('extdata', "wikipathways-20220310-gmt-Homo_sapiens.gmt", package='gson')
x <- read.gmt.wp(wpfile)
gsid2gene <- data.frame(gsid=x$wpid, gene=x$gene)
gsid2name <- unique(data.frame(gsid=x$wpid, name=x$name))
species <- unique(x$species)
version <- unique(x$version)
gson(gsid2gene=gsid2gene, gsid2name=gsid2name, species=species, version=version)
```

GSON-class *Class "GSON" This class represents gene set information.*

Description

Class "GSON" This class represents gene set information.

Slots

gsid2gene data.frame with two columns of 'gsid' and 'gene'

gsid2name data.frame with two columns of 'gsid' and 'name'

gene2name data.frame with two columns of 'gene' and 'name'

schema_version version of the GSON file schema

species species of the annotation

`gsname` gene set name, e.g., GO, KEGG

`version` version of the gene set

`accessed_date` time to obtain the gene set data

`keytype` keytype of genes

`gsidtype` identifier domain of the gene set IDs, e.g., GO, KEGG, REAC

`urlpattern` URL pattern to browse gene set online

`info` extra information

Author(s)

Guangchuang Yu https://yulab-smu.top

`gson_union` *Combine gene set collections*

Description

`gson_union()` merges `GSON` objects into one collection. Two objects are the same source when they declare the same `gsname`, and – once every object declares one – the same `gsidtype`; the gene sets of one source are joined by ID, so a gene set that appears in two files of one collection becomes one set holding every gene either file gives it. A gene set ID that two objects from different sources both use is a conflict, because the merged membership table would say that one ID names two sets of genes, and `.conflict` decides what happens: `"error"` refuses, `"prefix"` qualifies every gene set ID with the source it comes from (`KEGG:hsa00010`), and `"rename"` qualifies only the IDs that actually collide, leaving the rest of the collection alone. Where two objects name the same gene set or the same gene differently, the first object that says something wins.

Usage

```
gson_union(x = NULL, y = NULL, ..., .conflict = c("error", "prefix", "rename"))
```

Arguments

`x, y`	A `GSON` or `GSONList` object. Either may be omitted.
`...`	More `GSON` or `GSONList` objects.
`.conflict`	What to do about a gene set ID used by two different sources: `"error"`, `"prefix"` or `"rename"`.

Details

The metadata is merged by what the merged object can still honestly say: `species`, `keytype` and `schema_version` have to agree or the call is an error, because a membership table holds genes of one organism identified by one identifier type, while `gsname`, `gsidtype`, `version`, `accessed_date` and `info` are joined with `" + "` when the sources name themselves differently. `urlpattern` is kept only when every object declares the same one and no gene set ID was renamed – a pattern that resolves to a wrong page for part of the collection is worse than no pattern, and `gson_url()` says so.

Value

A GSON object holding every gene set of every input.

Examples

```
wpfile <- system.file('extdata', "wikipathways-20220310-gmt-Homo_sapiens.gmt", package='gson')
wp <- read.gmt.wp(wpfile, output = "GSON")
gson_union(wp[1:2], wp[3:4])
```

---

gson_url *URLs of the gene sets of a GSON*

Description

gson_url() substitutes each gene set ID into the {gsid} token of x@urlpattern, and browseGS() opens the resulting URL in a browser. The URL only comes from the object itself: gson ships no table of other databases' websites, so a producer that wants its gene sets to be browsable has to declare urlpattern.

Usage

```
gson_url(x, gsid = NULL)

browseGS(x, gsid, ...)
```

Arguments

x	A GSON object
gsid	Gene set IDs to build URLs for. Defaults to names(x), i.e. every gene set in the object.
...	Further arguments passed to utils::browseURL().

Value

gson_url() a named character vector of URLs, every entry NA_character_ and a warning when the object carries no urlpattern. browseGS() returns NULL invisibly.

Examples

```
wpfile <- system.file('extdata', "wikipathways-20220310-gmt-Homo_sapiens.gmt", package='gson')
wp <- read.gmt.wp(wpfile, output = "GSON")
gson_url(wp)[1:2]
gson_url(wp, c("WP100", "WP106"))
# browseGS(wp, "WP100") opens the same URL in a browser
```

gsonList — *construct a 'GSONList' object*

Description

construct a 'GSONList' object

Usage

```
gsonList(...)
```

Arguments

`...`	input GSON objects

Value

A 'GSONList' instance

read.gmt — *read.gmt*

Description

parse gmt file to a data.frame

write a GSON object to GMT format

Usage

```
read.gmt(gmtfile)

read.gmt.wp(gmtfile, output = "data.frame")

write.gmt(x, file = "")
```

Arguments

`gmtfile`	gmt file
`output`	one of 'data.frame' or 'GSON'
`x`	A `GSON` object
`file`	output GMT file

Value

data.frame

Author(s)

Guangchuang Yu

---

read.gson *read and write gson file*

Description

read and write gson file

Usage

```
read.gson(file)

write.gson(x, file = "")
```

Arguments

file	A gson file
x	A GSON instance

Value

A GSON instance

Examples

```
wpfile <- system.file('extdata', "wikipathways-20220310-gmt-Homo_sapiens.gmt", package='gson')
x <- read.gmt.wp(wpfile, output = "GSON")
f = tempfile(fileext = '.gson')
write.gson(x, f)
read.gson(f)
```

---

show *show method*

Description

show method for GSON instance

Usage

```
show(object)
```

Arguments

object	A GSON object

Value

message

Author(s)

Guangchuang Yu https://yulab-smu.top

---

validate_gson *Validate a GSON object*

Description

validate_gson() checks the core data contract of a GSON gene set collection. It is useful after constructing an object manually or reading one from an external source.

Usage

```
validate_gson(x, error = TRUE)
```

Arguments

x	A GSON object.
error	Logical. If TRUE, throw an error when validation fails. If FALSE, return a character vector of validation messages.

Details

Nothing is rewritten. A value the object declares is reported as it stands – a schema_version this package does not know how to read, a keytype or gsidtype that names no identifier type, a gsidtype that the gene set IDs of the object itself contradict – because a reader that knows what a file says is not the one that decides what the producer meant.

Value

TRUE if the object is valid. If error = FALSE, returns a character vector of validation messages when invalid.

Examples

```
x <- gson(data.frame(gsid = "GS1", gene = "gene1"))
validate_gson(x)
```

Index

∗ **classes**
    GSON-class, 4
[.GSON, 2
[[.GSON *([.GSON)*, 2

as.data.frame.GSON, 3

browseGS *(gson_url)*, 6

gson, 3
GSON-class, 4
gson_union, 5
gson_url, 6
gsonList, 7

read.gmt, 7
read.gson, 8

show, 8
show,GSON-method *(GSON-class)*, 4

validate_gson, 9

write.gmt *(read.gmt)*, 7
write.gson *(read.gson)*, 8