Package: ctdR (via r-universe)

November 25, 2026

Type Package

Title Enrichment Analysis of Chemical-Gene Interactions from the Comparative Toxicogenomics Database

Version 1.0.0

Description Identifies chemicals significantly associated with a set of genes using data from the Comparative Toxicogenomics Database (CTD, <https://ctdbase.org>). Supports four enrichment methods through a unified interface: Over-Representation Analysis (ORA) via a hypergeometric test on stats::phyper(), Gene Set Enrichment Analysis (GSEA) via 'fgsea', the competitive gene-set test CAMERA via 'limma', and per-sample Gene Set Variation Analysis (GSVA) via 'GSVA'. Users download the CTD chemical-gene interactions file once, import it with import_CTD(), and then run enrichment_CTD() on either a gene list (ORA, GSEA) or an expression matrix (CAMERA, GSVA). This package does not bundle or redistribute CTD data: its example interaction file is synthetic, and the tables of the replication vignette hold counts and p-values only. Users must download the required data files directly from <https://ctdbase.org> and comply with the CTD data licensing terms (see <https://ctdbase.org/about/legal.jsp>).

License Apache License 2.0 | file LICENSE

URL https://luigicorsaro.com/ctdR/, https://github.com/drake69/ctdR

BugReports https://github.com/drake69/ctdR/issues

Encoding UTF-8

Depends R (>= 4.5.0)

Imports fgsea, ggplot2, readr, org.Hs.eg.db, AnnotationDbi, limma, GSVA, stats, BiocIO, BiocFileCache, S4Vectors, SummarizedExperiment, methods

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

biocViews GeneSetEnrichment, Pathways, Annotation, GeneExpression, DataImport

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Config/pak/sysreqs
libmagick++-dev gsfonts libicu-dev libpng-dev libxml2-dev libssl-dev libx11-dev zlib1g-dev

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:57:01 UTC

RemoteUrl https://github.com/bioc/ctdR

RemoteRef RELEASE_3_24

RemoteSha f8c6e99bd5412c9bfd34d56f8d79ec160370cac0

Contents

ctdR-package 2
as_genesets_CTD 4
ctd_cache 5
ctd_provenance 6
CTDFile 8
enrichment_CTD 9
import_CTD 13
plot_CTD 15

Index **17**

`ctdR-package` *ctdR: Enrichment Analysis of Chemical-Gene Interactions from CTD*

Description

ctdR identifies chemicals significantly associated with a set of genes using data from the Comparative Toxicogenomics Database (CTD, `https://ctdbase.org`).

Four enrichment methods are supported through a unified interface:

ORA Over-Representation Analysis via `phyper`.

GSEA Gene Set Enrichment Analysis via `fgsea`.

CAMERA Competitive gene-set test accounting for inter-gene correlation, via `camera`.

GSVA Per-sample Gene Set Variation Analysis via `gsva`.

ORA and GSEA take a gene-level data frame. CAMERA and GSVA take expression data, either as a numeric matrix (genes x samples) or as a `SummarizedExperiment`; GSVA returns its per-sample scores in whichever of the two it was given, so sample annotation stays attached to the results.

Quick Start

1. Download **CTD_chem_gene_ixns.csv.gz** from the CTD downloads page, https://ctdbase.org/downloads/, in a browser. There is no need to decompress it.
2. Import the data once: `import_CTD("path/to/CTD_chem_gene_ixns.csv.gz")`
3. Run enrichment analysis: `enrichment_CTD(my_genes, method = "ORA")`

Data Licensing Disclaimer

This package does **not** bundle, redistribute, or embed any data from the Comparative Toxicogenomics Database. CTD data are created and maintained by NC State University and are subject to specific licensing terms and conditions.

Users are solely responsible for:

- Downloading the required data files directly from https://ctdbase.org.
- Reading and accepting the CTD Terms of Service at https://ctdbase.org/about/legal.jsp before using the data.
- Complying with all applicable CTD data licensing requirements, including proper citation of CTD in any publications or derived works.

By using ctdR with CTD data, you acknowledge that you have read, understood, and agreed to the CTD Terms of Service.

Author(s)

Maintainer: Luigi Corsaro <lcorsaro69@gmail.com> (ORCID)

Authors:

- Luigi Corsaro <lcorsaro69@gmail.com> (ORCID)

References

Davis AP, Wiegers TC, Johnson RJ, Sciaky D, Wiegers J, Mattingly CJ (2023). "Comparative Toxicogenomics Database (CTD): update 2023." *Nucleic Acids Research*, 51(D1), D1257-D1262. doi:10.1093/nar/gkac833. https://ctdbase.org

See Also

`import_CTD` to import CTD data, `enrichment_CTD` to run enrichment analysis.

`as_genesets_CTD` *Export CTD Gene Sets for Existing Enrichment Engines*

Description

Returns the cached CTD chemical gene sets as a named list, in the shape expected by third-party Bioconductor enrichment engines — for example the `gs` argument of `EnrichmentBrowser::sbea()`, or the input to `GSEABase::GeneSetCollection()`. This lets you run the CTD gene sets through existing infrastructure instead of, or alongside, `enrichment_CTD`.

Usage

```
as_genesets_CTD(
  id_type = c("entrez", "symbol"),
  interaction_types = NULL,
  cache_dir = .ctd_cache_dir()
)
```

Arguments

`id_type`	Either `"entrez"` (default) or `"symbol"`: the identifier space of the returned genes, to match your expression data or downstream engine.
`interaction_types`	Optional character vector of CTD interaction actions (e.g. `"increases^expression"`) to retain. `NULL` (default) keeps all interactions.
`cache_dir`	Directory holding the cached CTD files. Defaults to the **ctdR** user cache populated by `import_CTD`.

Details

ctdR builds *chemical-centric* gene sets: each CTD chemical becomes a gene set of its interacting genes. `enrichment_CTD` consumes these internally; this helper exposes the same gene sets so they integrate with the wider ecosystem. Call `import_CTD` once first to populate the cache.

The returned representation is deliberately a plain named list (not a bespoke class): it is accepted directly by `EnrichmentBrowser::sbea()` and is a one-line conversion away from a `GeneSetCollection`. **ctdR** does not take a dependency on those engines — you supply them.

Value

A named list: names are CTD `ChemicalIDs`, elements are character vectors of gene identifiers (Entrez IDs or HGNC symbols).

See Also

`import_CTD` to populate the cache; `enrichment_CTD` for the built-in enrichment interface.

Examples

```
# Examples write to a temporary cache, so running them cannot
# disturb CTD data you have already imported. Set the same option
# yourself to keep an analysis isolated from your main cache.
options(ctdR.cache = tempfile())

sample_file <- system.file(
    "extdata", "CTD_chem_gene_ixns_sample.csv",
    package = "ctdR"
)
import_CTD(sample_file)
gene_sets <- as_genesets_CTD("entrez")
length(gene_sets)
gene_sets[[1]]

## Not run:
# Hand the CTD gene sets to an existing engine (EnrichmentBrowser):
library(EnrichmentBrowser)
res <- sbea(method = "camera", se = my_se, gs = as_genesets_CTD("entrez"))
gsRanking(res)

## End(Not run)
```

---

ctd_cache *Retrieve Cached CTD Data*

Description

Returns a processed CTD table from the **ctdR** cache populated by `import_CTD`, without reaching into the `.rda` cache files by hand. Use this instead of `load(file.path(cache_dir, "...rda"))` when you need the chemical metadata or the raw interaction table in your own code.

For the chemical *gene sets* (the input to enrichment engines) use `as_genesets_CTD` instead.

Usage

```
ctd_cache(what = c("chemicals", "interactions"), cache_dir = .ctd_cache_dir())
```

Arguments

`what`	Which cached table to return: `"chemicals"` a data frame with one row per chemical (`ChemicalID`, `ChemicalName`). `"interactions"` the long-format interaction table (`ChemicalID`, `EntrezID`, `InteractionActions`).
`cache_dir`	Directory holding the cached CTD files. Defaults to the **ctdR** user cache populated by `import_CTD`.

Value

A data frame with the requested cached table.

See Also

import_CTD to populate the cache; as_genesets_CTD for the chemical gene sets.

Examples

```
# Examples write to a temporary cache, so running them cannot
# disturb CTD data you have already imported. Set the same option
# yourself to keep an analysis isolated from your main cache.
options(ctdR.cache = tempfile())

sample_file <- system.file(
    "extdata", "CTD_chem_gene_ixns_sample.csv",
    package = "ctdR"
)
import_CTD(sample_file)
chem <- ctd_cache("chemicals")
head(chem)
```

---

ctd_provenance *Provenance of the CTD data an analysis ran on*

Description

Returns the record of which CTD release produced a result: the `Report created` date CTD stamps into its own file header, where the file came from, when it was imported, and how much of it was kept.

An analysis is only reproducible if the version of the data behind it can be named. CTD is re-released continuously and its downloads are not versioned in the filename, so the release date inside the header is the only thing that identifies which snapshot a result came from. This carries it from the file all the way to the object you report on.

Usage

```
ctd_provenance(x)

## S3 method for class 'ctd_provenance'
print(x, ...)
```

Arguments

x	An object returned by enrichment_CTD, or the DataFrame returned by importing a CTDFile. Omit it to read the record of the data currently cached, which answers "which release am I about to analyse" before any analysis has been run.
...	Ignored, present for compatibility with the generic.

Details

Where the record is stored depends on what the object is, because the methods do not all return the same container. Objects that provide a `metadata()` slot keep it there, which covers the `SummarizedExperiment` returned by GSVA and the `DataFrame` returned by importing a `CTDFile`. The data frames returned by ORA, GSEA and CAMERA, and a plain score matrix, have no such slot, so there it rides on an attribute. Use this accessor rather than reaching for either directly, so that code keeps working whichever method produced the object.

The record survives subsetting, ordering, `head()` and the common **dplyr** verbs. It does not survive `merge()` or `subset()`, which drop attributes; retrieve it before those if you need it afterwards.

Value

An object of class `ctd_provenance`: a list with `report_created` (the CTD release string, `NA` if the file carried none), `source`, `accessed`, `n_chemicals`, `n_interactions`, `organism` (the NCBI Taxonomy IDs kept, or `"all"`) and `ctdR_version`. Returns `NULL`, with a warning, when `x` carries no record.

See Also

`import_CTD`, which reads the record, and `enrichment_CTD`, which attaches it to its results.

Examples

```
# Examples write to a temporary cache, so running them cannot
# disturb CTD data you have already imported. Set the same option
# yourself to keep an analysis isolated from your main cache.
options(ctdR.cache = tempfile())

sample_file <- system.file(
    "extdata", "CTD_chem_gene_ixns_sample.csv",
    package = "ctdR"
)
import_CTD(sample_file)

genes <- data.frame(
    EntrezID = c("7124", "3569", "7157", "672", "1956"),
    pvalue = c(0.001, 0.003, 0.01, 0.02, 0.05)
)
# Which release is cached, before running anything:
ctd_provenance()

res <- enrichment_CTD(genes, method = "ORA")
ctd_provenance(res)
```

CTDFile *CTD Chemical-Gene Interactions File*

Description

CTDFile is a BiocFile subclass representing a Comparative Toxicogenomics Database (CTD) chemical-gene interactions file (CTD_chem_gene_ixns.csv or its gzip-compressed .csv.gz form). Pair it with import to read the file into R following the Bioconductor **BiocIO** import/export convention, as an alternative to the side-effecting import_CTD caching workflow.

In **ctdR**, **CTD** always denotes the *Comparative Toxicogenomics Database* (https://ctdbase.org). The class is named CTDFile (not CTD) to avoid confusion with unrelated software that shares the acronym (the CRAN package **CTD** is a graph algorithm; the ctd object in **EWCE** is a cell-type dataset).

Usage

```
CTDFile(resource)

## S4 method for signature 'CTDFile,ANY,ANY'
import(con, format, text, ...)
```

Arguments

resource	Character scalar: a local path or a URL to the CTD chemical-gene interactions file.
con	A CTDFile object to import.
format	Ignored; present for compatibility with the **BiocIO** import generic.
text	Ignored; present for compatibility with the **BiocIO** import generic.
...	Additional arguments (currently unused).

Value

CTDFile() returns a CTDFile object. import() on a CTDFile returns a DataFrame of validated human (OrganismID 9606) CTD chemical-gene interactions.

Data Licensing Disclaimer

This package does **not** bundle or redistribute any CTD data. The Comparative Toxicogenomics Database is maintained by NC State University and its data are subject to specific licensing terms. Users are responsible for downloading the data directly from https://ctdbase.org and for complying with the CTD Terms of Service (https://ctdbase.org/about/legal.jsp).

See Also

import_CTD for the caching workflow required before enrichment_CTD; BiocFile for the parent class.

Examples

```
sample_file <- system.file(
    "extdata", "CTD_chem_gene_ixns_sample.csv",
    package = "ctdR"
)
ctd_file <- CTDFile(sample_file)
ctd_file
interactions <- BiocIO::import(ctd_file)
head(interactions)
```

`enrichment_CTD` *Chemical Enrichment Analysis Using CTD*

Description

Identifies chemicals whose known gene targets are significantly enriched in a user-supplied gene list or expression matrix, using data from the Comparative Toxicogenomics Database (CTD).

Four methods are available:

ORA Over-Representation Analysis (default). Tests whether the overlap between your gene list and each chemical's target genes is larger than expected by chance. Uses a hypergeometric test computed directly on `phyper`. Input: data frame with column `EntrezID` (character or numeric Entrez gene IDs) and an optional numeric value column.

GSEA Gene Set Enrichment Analysis. Uses a ranked gene list (ranked by the numeric column in the input, e.g. p-values or fold changes) to detect chemicals whose targets cluster toward the top or bottom of the ranking. Uses `fgsea`.

CAMERA Competitive gene-set test accounting for inter-gene correlation. Uses `camera`. Input: a numeric expression matrix (genes x samples) or a `SummarizedExperiment`, plus a design matrix and a contrast.

GSVA Gene Set Variation Analysis (sample-level scoring). Returns per-sample enrichment scores for each chemical, suitable for downstream clustering or association testing. Uses `gsva`. Input: a numeric expression matrix (genes x samples) or a `SummarizedExperiment`, which is also the container the scores are returned in.

Usage

```
enrichment_CTD(
  x,
  method = c("ORA", "GSEA", "CAMERA", "GSVA"),
  design = NULL,
  contrast = NULL,
  id_type = NULL,
  pAdjustMethod = "BH",
  interaction_types = NULL,
  gene_id_type = c("symbol", "entrez"),
```

```
  universe = NULL,
  alpha = NULL,
  alpha_column = NULL,
  assay = NULL,
  ...
)
```

Arguments

x	The input. Its expected type depends on `method`: • For `"ORA"` and `"GSEA"`: a data frame with at least two columns, `EntrezID` (character or numeric Entrez gene IDs) and a numeric value column (e.g. p-value). For ORA this is either the genes you have already selected, in which case say what the background was with `universe`, or the whole result table with `alpha` to select from it. For GSEA, an optional column named `stat` can be added with a signed ranking statistic (e.g. the moderated t-statistic from `limma::eBayes()`); when present it is used directly for ranking, preserving directionality and avoiding ties. When absent, the second column must hold p-values: it is transformed via `-log10()` with a warning, scored one-sided (`scoreType = "pos"`), and values outside [0, 1] stop the run. The second column is ignored by ORA. • For `"CAMERA"` and `"GSVA"`: either a numeric expression matrix with genes in rows and samples in columns, or a `SummarizedExperiment` whose assay holds that matrix (see `assay` to choose which one). `rownames(x)` must be either Entrez IDs or HGNC SYMBOLs.
method	Character. Enrichment method: `"ORA"` (default), `"GSEA"`, `"CAMERA"`, or `"GSVA"`.
design	Design matrix (required when `method = "CAMERA"`).
contrast	Contrast specification for `camera` (column number, column name, or numeric vector). Required when `method = "CAMERA"`.
id_type	Either `"entrez"`, `"symbol"`, or `NULL` (default) for auto-detection from `rownames(x)`. Only used when `method` is `"CAMERA"` or `"GSVA"`.
pAdjustMethod	Character. Multiple-testing correction applied to the raw p-values, passed to `p.adjust`. Any value in `stats::p.adjust.methods` is accepted: `"holm"`, `"hochberg"`, `"hommel"`, `"bonferroni"`, `"BH"` (Benjamini-Hochberg, the default), `"BY"`, `"fdr"` (alias for `"BH"`), or `"none"`. See `p.adjust` for the meaning of each method. Not used for `method = "GSVA"` (which returns per-sample scores, not p-values).
interaction_types	Character vector of CTD `InteractionActions` values to retain when building gene sets, or `NULL` (default) to use all cached interactions. Values follow the `verb\^{}noun` convention used by CTD, e.g. `"increases\^{}expression"`, `"decreases\^{}expression"`, `"affects\^{}binding"`. A gene is included in a chemical's set if *any* of its recorded interaction actions matches one of the specified types. Requires that `import_CTD()` has been run (the filter is applied to the cached `ctd_interactions.rda` file). Restricting to expression interactions is recommended for RNA-seq analyses to improve biological specificity.

`gene_id_type` Character. Identifier type used in the `EnrichedGenes` output column: `"symbol"` (default) returns HGNC gene symbols with Entrez ID as fallback for unmapped genes; `"entrez"` skips the symbol lookup and returns Entrez IDs directly. Only applies to `"ORA"` and `"GSEA"`; ignored by `"CAMERA"` and `"GSVA"`.

`universe` Background gene identifiers for `"ORA"`: every gene that entered your differential test, not only the significant ones and not every gene sequenced. `NULL` (default) falls back to every gene in the CTD gene sets, and says so, because the package cannot know what your platform measured.

This is the input that decides whether an ORA result means anything. A background wider than what the experiment could detect fills the urn with genes that could never have been drawn, so the overlap looks more selective than it was and p-values come out too small. The error is anti-conservative. On the RNA-seq example bundled with this package, the fallback background returns 32 chemicals at FDR < 0.05 and the correct one returns 19.

ORA is the only method that needs this argument, and the reason is the shape of its input. A bare gene list carries no record of what was measurable, so the background has to be supplied separately. GSEA ranks the whole list you give it, which is already the background; `"CAMERA"` and `"GSVA"` intersect the gene sets with `rownames(x)`, so theirs is the set of measured genes by construction. Passing `universe` to any of those three raises a warning rather than being quietly dropped.

`alpha` Significance threshold for `"ORA"`, or `NULL` (default). Supplying it says that `x` is the *whole* result table rather than a list already filtered: the rows whose second column falls below `alpha` become the genes to test, and every row becomes the background.

Every row of the table becomes the background, the selected genes included: a hypergeometric test draws n genes from an urn of N, and the drawn ones were in the urn. The background is not the non-significant remainder.

This is the safer way to run ORA, because the background is then derived from the same object as the gene list and cannot disagree with it. Filtering first and passing `universe` separately asks the caller to reconnect two things that were together a moment earlier, and that reconnection is what goes wrong.

Name the column to threshold with `alpha_column`: which p-value to judge on is the researcher's decision, not the package's. The function reports the column it used, how many genes passed and how many form the background.

Mutually exclusive with `universe`, and ignored by the other three methods, which already hold their own background.

`alpha_column` Which column `alpha` applies to: a name, a positive index, or `NULL` (default) for the second column. Naming it matters on a real result table, where the second column is usually a fold change: `limma::topTable()` puts `logFC` there. Pass the whole table and say which p-value to judge on, `alpha_column = "padj"` or `"adj.P.Val"`, rather than cutting the table down to two columns first.

`assay` Which assay to use when `x` is a `SummarizedExperiment`: an assay name, a positive index, or `NULL` (default) for the first assay. Ignored when `x` is a matrix or a data frame.

`...` Additional arguments forwarded to the underlying engine: `ora` for ORA (`universe`, `minGSSize`, `maxGSSize`; both thresholds are chosen for CTD rather than inher-

ited, minGSSize defaulting to 2 and maxGSSize to Inf, see ora for the measurements behind them), fgseaMultilevel for GSEA (e.g. minSize, maxSize, nproc), camera for CAMERA, gsva for GSVA (e.g. minSize, maxSize).

Details

Before calling this function you must import the CTD data once with import_CTD. If the cached data is not found, the function stops with an informative error message.

Value

- For "ORA", "GSEA", and "CAMERA": a data frame of enrichment results sorted by PValueAdjusted ascending. All three methods share the leading columns ChemicalID, ChemicalName, Method, PValue, PValueAdjusted; method-specific extras follow (see the package vignette for the full per-method schema).
- For "GSVA": enrichment scores with chemicals (CTD chemical IDs) in rows and samples in columns. The container follows the input: a matrix in returns a numeric matrix, while a SummarizedExperiment in returns a SummarizedExperiment whose assay holds the scores and whose colData is carried over from the input, so sample annotation stays attached to the results.

Data Licensing Disclaimer

This package does **not** bundle or redistribute any CTD data. The Comparative Toxicogenomics Database is maintained by NC State University and its data are subject to specific licensing terms. Users must download the data directly from https://ctdbase.org and comply with the CTD Terms of Service (https://ctdbase.org/about/legal.jsp).

See Also

import_CTD to import and cache the CTD data; plot_CTD to visualize results.

Examples

```
# Import the bundled sample data first:
# Examples write to a temporary cache, so running them cannot
# disturb CTD data you have already imported. Set the same option
# yourself to keep an analysis isolated from your main cache.
options(ctdR.cache = tempfile())

sample_file <- system.file(
    "extdata", "CTD_chem_gene_ixns_sample.csv",
    package = "ctdR"
)
import_CTD(sample_file)

# ORA / GSEA: prepare a gene list with Entrez IDs and a numeric value
genes <- data.frame(
    EntrezID = c("7124", "3569", "7157", "672", "1956"),
    pvalue = c(0.001, 0.003, 0.01, 0.02, 0.05)
)
```

```
ora_results <- enrichment_CTD(genes, method = "ORA")
gsea_results <- enrichment_CTD(genes, method = "GSEA")

# CAMERA / GSVA: expression data plus, for CAMERA, design + contrast.
# Uses the bundled GSE311566 subset, a SummarizedExperiment
# (Dex vs DMSO, female PBMCs; see inst/extdata/README.md).
se <- readRDS(system.file(
    "extdata", "GSE311566_subset.rds", package = "ctdR"
))
d <- model.matrix(~ se$group)
camera_results <- enrichment_CTD(se, method = "CAMERA",
    design = d, contrast = 2)

# GSVA: SummarizedExperiment in, SummarizedExperiment out,
# so the sample annotation stays attached to the scores.
gsva_scores <- enrichment_CTD(se, method = "GSVA")
SummarizedExperiment::colData(gsva_scores)$group
```

`import_CTD` *Import CTD Chemical-Gene Interaction Data*

Description

Parses, filters, and caches the CTD chemical-gene interactions file so that it can be used by `enrichment_CTD`. This function must be called **once** before running any enrichment analysis.

The raw data file must be downloaded manually from the CTD website. The required file is **CTD_chem_gene_ixns.csv.gz**, available at `https://ctdbase.org/reports/CTD_chem_gene_ixns.csv.gz`.

Usage

```
import_CTD(file_path, organism = 9606)
```

Arguments

`file_path`	Character. A local path **or** a URL to the CTD chemical-gene interactions file (`CTD_chem_gene_ixns.csv` or `CTD_chem_gene_ixns.csv.gz`). Remote URLs are downloaded and cached with **BiocFileCache**; the package assumes no default URL, so you supply the source and a one-time CTD data-licensing reminder is shown on first remote fetch.
`organism`	NCBI Taxonomy IDs of the species to keep, as numbers or character strings. Defaults to `9606` (*Homo sapiens*). `"all"` keeps every species and forces symbol-keyed gene sets to pool species by gene symbol (see the Species section). The choice is recorded in the provenance of the import (see `ctd_provenance`).

Details

Processing steps performed by `import_CTD`:

1. Reads the file (plain or gzip-compressed), taking the column names from its commented header.
2. Keeps the interactions of the species named in `organism`, **Homo sapiens** only (OrganismID 9606) by default.
3. For each chemical, collects the associated Entrez gene IDs.
4. Maps Entrez IDs to HGNC gene symbols via **org.Hs.eg.db**.
5. Saves four cached objects via **BiocFileCache**: `chemicals`, `ChemicalName_GeneEntrezIds`, `ChemicalName_GeneSymbols`, and `ctd_interactions` (a long-format table of chemical–gene–action triples used by `enrichment_CTD(interaction_types = ...)`).

The cache is managed by **BiocFileCache** under `tools::R_user_dir("ctdR", "cache")`. To re-import (e.g. after downloading a newer CTD release), simply call `import_CTD()` again — existing cache resources are overwritten.

Filtering by interaction type (e.g., to retain only `"increases^expression"` interactions) is done at enrichment time via the `interaction_types` argument of `enrichment_CTD`, not here — so a single import supports any combination of filters without re-running this step.

Value

Invisible `NULL`. Called for its side effect of caching the processed data.

Data Licensing Disclaimer

This package does **not** bundle or redistribute any CTD data. The Comparative Toxicogenomics Database is maintained by NC State University and its data are subject to specific licensing terms. Users are responsible for downloading the data directly from `https://ctdbase.org` and for complying with the CTD Terms of Service (`https://ctdbase.org/about/legal.jsp`). By using this function you acknowledge that you have read and accepted those terms.

Species

By default only human interactions are kept, as in every earlier version. `organism` accepts other NCBI Taxonomy IDs (for example `10090` for mouse, `10116` for rat), several at once, or `"all"` for every species in the file, including the interactions CTD records without a species.

When species other than human are imported, and always with `organism = "all"`, the symbol-keyed gene sets are **forced** to pool genes across species **by gene symbol** (orthology by homonymy); there is no option to turn it off: a mouse *Esr1* interaction counts as *ESR1*. This is how the CTD website aggregates genes, and what published analyses based on it did; it is a match by name, not an orthology mapping. Human genes take their symbol from **org.Hs.eg.db** as before; genes of other species take the symbol CTD gives them, in upper case. The Entrez-keyed gene sets are not pooled: each interaction keeps the Entrez ID of the gene CTD curated it on, so analyses on Entrez IDs (`gene_id_type = "entrez"`, and GSEA) match a non-human gene only to an input in the same species.

See Also

enrichment_CTD for running enrichment analysis after import, including the interaction_types filter.

Examples

```
# Examples write to a temporary cache, so running them cannot
# disturb CTD data you have already imported. Set the same option
# yourself to keep an analysis isolated from your main cache.
options(ctdR.cache = tempfile())

sample_file <- system.file(
    "extdata", "CTD_chem_gene_ixns_sample.csv",
    package = "ctdR"
)
import_CTD(sample_file)
```

---

plot_CTD *Plot Chemical Enrichment Results*

Description

Visualizes the output of enrichment_CTD. The function auto-detects the enrichment method from the input class and column structure:

- **ORA / GSEA** (data frame with Count or NES): bar or dot plot of fold enrichment.
- **CAMERA** (data frame with Direction and NGenes): bar or dot plot of $-\log_{10}(\text{padj})$, colored by direction of enrichment.
- **GSVA** (numeric matrix chemicals x samples, or a SummarizedExperiment wrapping one): heatmap of per-sample enrichment scores for the top-N chemicals selected by score variance across samples.

Usage

```
plot_CTD(results, type = "bar", n = 20, title = NULL)
```

Arguments

results	A data frame, a numeric matrix, or a SummarizedExperiment returned by enrichment_CTD.
type	Character. Plot type for tabular results: "bar" (default) or "dot". Ignored for GSVA score input.
n	Integer. Number of top chemicals to display (default 20). Selection criterion: ascending padj for tabular results, descending score variance across samples for GSVA matrices.
title	Character. Plot title. If NULL (default), a title is generated automatically based on the detected method.

Value

A ggplot2 object.

Examples

```
# Examples write to a temporary cache, so running them cannot
# disturb CTD data you have already imported. Set the same option
# yourself to keep an analysis isolated from your main cache.
options(ctdR.cache = tempfile())

sample_file <- system.file(
    "extdata", "CTD_chem_gene_ixns_sample.csv",
    package = "ctdR"
)
import_CTD(sample_file)

# ORA / GSEA
genes <- data.frame(
    EntrezID = c("7124", "3569", "7157", "672", "1956"),
    pvalue = c(0.001, 0.003, 0.01, 0.02, 0.05)
)
ora_results <- enrichment_CTD(genes, method = "ORA")
plot_CTD(ora_results, type = "bar")
plot_CTD(ora_results, type = "dot")

# GSVA scores: the heatmap accepts either the score matrix or the
# SummarizedExperiment that enrichment_CTD() returns for an SE input.
se <- readRDS(system.file(
    "extdata", "GSE311566_subset.rds", package = "ctdR"
))
gsva_scores <- enrichment_CTD(se, method = "GSVA")
plot_CTD(gsva_scores)
```

Index

.CTDFile *(CTDFile)*, 8

as_genesets_CTD, 4, *5, 6*

BiocFile, *8*

camera, *2*, *9, 10*, *12*
ctd_cache, 5
ctd_provenance, 6, *13*
CTDFile, *6, 7*, 8
CTDFile-class *(CTDFile)*, 8
ctdR *(ctdR-package)*, 2
ctdR-package, 2

DataFrame, *8*

enrichment_CTD, *3, 4*, *6–8*, 9, *13–15*

fgsea, *2*, *9*
fgseaMultilevel, *12*

gsva, *2*, *9*, *12*

import, *8*
import,CTDFile,ANY,ANY-method
(CTDFile), 8
import_CTD, *3–8*, *12*, 13

ora, *11, 12*

p.adjust, *10*
phyper, *2*, *9*
plot_CTD, *12*, 15
print.ctd_provenance *(ctd_provenance)*, 6

SummarizedExperiment, *2*, *7*, *9–12*, *15*