Package: pathdb (via r-universe)

June 4, 2026

Type Package

Title Comprehensive Database for Pathway Enrichment Analysis

Version 0.1.0

Description Provides access to large-scale genomics data from the South Dakota State University's bioinformatics database, a unified platform for pathway analysis of over 13,000 organisms. It includes various gene mappings, gene characteristics, and pathway mapping data from KEGG, GOBP, GOCC, and many more pathway databases. Also provides various helper functions for processing RNA-Seq data for differential expression analysis and pathway enrichment analysis, occasionally sourced from code from Integrated Differential Expression & Pathway analysis (iDEP), developed by Ge, S.X., Son, E.W. & Yao, R. (2018) <doi:10.1186/s12859-018-2486-6>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Language en-US

Imports DBI, dplyr, edgeR, R.utils, RSQLite, stats, utils, tools

Roxygen list(markdown = TRUE)

Suggests clusterProfiler, curl, knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Depends R (>= 4.1.0)

URL https://github.com/aidanfred24/pathdb, https://aidanfred24.github.io/pathdb/

BugReports https://github.com/aidanfred24/pathdb/issues

Config/Needs/editorial spelling

Config/roxygen2/version 8.0.0

Repository https://aidanfred24.r-universe.dev

Date/Publication 2026-06-03 22:02:37 UTC

RemoteUrl https://github.com/aidanfred24/pathdb

RemoteRef HEAD

RemoteSha 09c376a4af39c9d60fe434e37971a8754cca6cca

Contents

`connect_database` *Query Bioinformatics Database*

Description

Retrieves database (.db file) from the SDSU bioinformatics database, creates a connection via SQLite

Usage

```
connect_database(species_id = NULL)
```

Arguments

`species_id`	ID of species selected (Loads organism info data if NULL)

Value

SQLite connection to the downloaded file

Examples

```
# Connect to organism information database
conn <- connect_database()

# Query information using connection
x <- DBI::dbGetQuery(
  conn = conn,
  statement = "select * from orgInfo;"
  )
head(x)

# Disconnect from database file
DBI::dbDisconnect(conn = conn)

# Connect to species information database (e.g. Indian Cobra)
conn <- connect_database(species_id = 99)

# Query information using connection
x <- DBI::dbGetQuery(
  conn = conn,
  statement = "select * from geneInfo;"
  )
head(x)

# Disconnect from database file
DBI::dbDisconnect(conn = conn)
```

---

convert_id *Convert Gene IDs to Ensembl*

Description

Queries the database to map user-provided gene identifiers to Ensembl/Entrez IDs. To ensure best matching and conversion, please verify that all gene identifiers have no whitespace and are at least 2 characters long. Results are often more conservative of initial genes if data is provided, as duplicate removal is done by variance of each gene (highest variance is kept).

Usage

```
convert_id(genes, data = NULL, species_id, id_type = "ens")
```

Arguments

genes	A vector or character string of gene identifiers to convert.
data	Optional data frame or matrix. If provided, the function attempts to match genes to the row names or a column in data and merges the conversion results with the original data.

`species_id` Numeric. The ID of the species for the database connection.

`id_type` Character. The type of ID to convert to:

- "ens" = Ensembl gene IDs (Default)
- "entrez" = Entrez gene IDs
  - WARNING: Not all species or genes have Entrez gene IDs available
  - May take longer than Ensembl IDs
  - Will likely have duplicate Entrez IDs

Value

A data frame.

- If `data` is `NULL`: Returns a mapping table with original IDs and IDs of selected type.
- If `data` is provided: Returns `data` merged with the IDs of selected type. Returns `NULL` if `species_id` is missing or no matches are found.
- Any whitespace found in original IDs will be removed.

Examples

```
# CAUTION: The human database is very large, running these examples require
# the download of the human database.

# View our experimental gene IDs
head(rownames(hypoxia_reads))

# Convert IDs to Ensembl format for further analysis
ens_conv <- convert_id(genes = rownames(hypoxia_reads),
                       species_id = 96)

# Yields a conversion table for our genes
head(ens_conv)

# Can also convert to Entrez IDs, if needed
entrez_conv <- convert_id(genes = rownames(hypoxia_reads),
                          species_id = 96,
                          id_type = "entrez")

# Yields a conversion table for our genes
head(entrez_conv)

# We want to automatically convert our IDs within our data
ens_hypoxia <- convert_id(genes = rownames(hypoxia_reads),
                          species_id = 96,
                          data = hypoxia_reads)

# Original data
head(hypoxia_reads)

# Converted data
head(ens_hypoxia)
```

get_genes *Get Gene Information*

Description

Retrieves gene information (e.g., Ensembl IDs, positions) for a specific species, optionally filtered by a list of user-provided gene identifiers after converting to Ensembl IDs.

Usage

```
get_genes(species_id, genes = NULL)
```

Arguments

species_id	Numeric. The ID of the desired species.
genes	A vector or list of gene identifiers to filter by. If NULL, returns the full gene table.

Value

A data frame containing gene information (from the geneInfo table). If genes are provided, the result is filtered to match the converted Ensembl IDs.

Examples

```
# CAUTION: The human database is very large, running these examples require
# the download of the human database.

# We have gene IDs that are not commonly recognized
head(rownames(hypoxia_reads))

# Retrieve gene information for genes in our sample
# Converts to Ensembl IDs first
genes <- get_genes(species_id = 96,
                   genes = rownames(hypoxia_reads))

head(genes)

# Retrieve all genes for desired species
all_genes <- get_genes(species_id = 96)
head(all_genes)

# This is the same as running get_table(96, "geneInfo")
all(get_genes(96) == get_table(96, "geneInfo"), na.rm = TRUE)
```

get_pathways *Get Species Pathways*

Description

Retrieves pathway information for a specific species and optionally filters for specific genes.

Usage

```
get_pathways(species_id, genes = NULL, category = "GOBP")
```

Arguments

species_id	Numeric. The ID of the desired species (e.g., from srch_species).
genes	A vector or column of a data frame containing gene IDs of interest. If NULL (default), returns all pathways for the species.
category	Character. A vector or character constant of pathway categories/databases (e.g. KEGG, GOBP, GOCC, etc.). It is not recommended to use all categories, as some species have many, leading to performance issues

Details

The function first retrieves the pathway and pathwayInfo tables for the specified species. If a list of genes is provided, it converts the IDs to Ensembl IDs, matches them against the pathway map, and joins the results with pathway metadata.

Value

A data frame containing pathway information. If genes are provided, the data frame is filtered to include only pathways containing those genes and joined with gene mapping data.

Examples

```
# CAUTION: The human database is very large, running these examples require
# the download of the human database.

# Get GOBP pathways for our genes of interest
path_info <- get_pathways(
  species_id = 96,
  genes = rownames(hypoxia_reads),
  category = "GOBP"
)
head(path_info)
```

get_table *Get Table from Selected Database*

Description

Retrieves a specific table from the database for a selected species.

Usage

```
get_table(species_id = NULL, table = NULL)
```

Arguments

species_id	Numeric. The selected species ID. If NULL, the function defaults to loading general organism info.
table	Character. The name of the table to retrieve (e.g., "geneInfo", "pathway"). If NULL, defaults to "geneInfo" (if species_id is provided) or "orgInfo" (if species_id is NULL).

Value

A data frame containing the data from the selected table.

See Also

list_tables to see available tables for a species.

Examples

```
# Retrieve geneInfo table for Indian Cobra Species
cobra_genes <- get_table(species_id = 99,
                         table = "geneInfo")

# View table
head(cobra_genes)
```

hypoxia_deseq *Hypoxia Data Differential Expression Analysis Results*

Description

Results of performing differential expression analysis (DESeq2) on gene counts gathered in the following experiment: RNAseq transcriptomic profile of glioblastoma stem-like cells derived from U87MG cell line treated with a selective A3 adenosine receptor antagonist (MRS1220) under hypoxia.

Usage

```
hypoxia_deseq
```

Format

hypox_deseq**:**

A data frame with 13,818 rows and 6 columns:

baseMean Mean of normalized counts for all samples

log2FoldChange Log2 fold change between treated and control

lfcSE Standard error estimate for the log2 fold change estimate

stat Wald statistic

pvalue Wald test p-value

padj Benjamini-Hochberg adjusted p-value

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE100146

---

hypoxia_reads *Hypoxia Data Gene Counts*

Description

Gene counts gathered in the following experiment: RNAseq transcriptomic profile of glioblastoma stem-like cells derived from U87MG cell line treated with a selective A3 adenosine receptor antagonist (MRS1220) under hypoxia.

Usage

```
hypoxia_reads
```

Format

hypoxia_reads**:**

A data frame with 35,238 rows and 4 columns:

MRS1220_hypoxia_rep1 Treatment replication 1 counts

MRS1220_hypoxia_rep2 Treatment replication 2 counts

vehicle_hypoxia_rep1 Control replication 1 counts

vehicle_hypoxia_rep2 Control replication 2 counts

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE100146

hypoxia_T2G *Example KEGG Pathway Mapping for Hypoxia*

Description

An example TERM2GENE mapping tailored for the hypox_deseq dataset, demonstrating the T2G_prep function's output.

Usage

```
hypoxia_T2G
```

Format

A data frame with 2 columns:

description Pathway ID or description

gene Ensembl Gene ID

list_tables *List Table Options*

Description

Lists all available tables within the database for a specific species.

Usage

```
list_tables(species_id = NULL)
```

Arguments

species_id	Numeric. The selected species ID. If NULL, loads the general organism info database.

Value

A character vector of table names available in the database connection.

Examples

```
# List all tables available for species 99 (Indian Cobra)
list_tables(species_id = 99)
```

path_categories *Retrieve Pathway Categories*

Description

Retrieves pathway category options (e.g. KEGG, GOBP, etc.) for a given species. May take longer for well-documented species (i.e. Human)

Usage

```
path_categories(species_id = NULL)
```

Arguments

species_id	Numeric. The ID of a desired species from database, found using `srch_species()`

Value

Data frame of pathway categories for given species

Examples

```
# Get pathway categories for species 99 (Indian Cobra)
categories <- path_categories(species_id = 99)
head(categories)
```

path_filter *Filter Pathway Information By Category*

Description

Retrieves pathway mapping information for a specific species, filtered by one or more pathway categories (e.g., "GOBP", "KEGG"). Optionally, the results can be further restricted to a specific list of genes.

Usage

```
path_filter(species_id, genes = NULL, category = "GOBP")
```

Arguments

species_id	Numeric. The ID of the species to search for.
genes	Character vector (optional). A vector of gene IDs to filter the pathways. If `NULL` (default), pathways for all genes in the category are returned.
category	Character or character vector. The pathway category or categories to filter by (e.g., `"GOBP"`, `"KEGG"`, `"GOCC"`). Default is `"GOBP"`.

Value

A data frame containing the pathway mapping information (such as gene, pathway ID, and description) for the specified categories and genes.

Examples

```
# Get all GO Biological Process pathways for Human (ID 96)
gobp_paths <- path_filter(species_id = 96, category = "GOBP")

# Get KEGG pathways for specific genes in a dataset
data(hypoxia_reads)
kegg_paths <- path_filter(
    species_id = 96,
    genes = rownames(hypoxia_reads)[1:100],
    category = "KEGG"
)
```

---

process_data *Process Gene Expression Data*

Description

Performs pre-processing, missing value imputation, filtering, and transformation on gene expression count data.

Usage

```
process_data(
  data,
  missing_value = "geneMedian",
  min_cpm = 0.5,
  n_min_samples = 1,
  rescale = FALSE
)
```

Arguments

data	A numeric matrix or data frame (> 1 columns) of gene expression counts.
missing_value	Character. Method to handle missing values. Options: • "geneMedian": Impute using the median expression of the gene across samples. • "treatAsZero": Replace NAs with 0. • "groupMedian": Impute using the median of the sample group (detected from colnames).
min_cpm	Numeric. Minimum counts per million threshold for filtering genes.

n_min_samples	Numeric. Minimum number of samples that must meet the min_cpm threshold for a gene to be retained.
rescale	Logical. TRUE allows for rescaling if values are exceedingly large.

Value

The processed and transformed data matrix.

Examples

```
# Check example data
summary(pathdb::hypoxia_reads)
nrow(pathdb::hypoxia_reads)

# YOU decide how your data is transformed.
# Here, we want to:
# Replace missing values with median
# Set minimum counts-per-million of 0.4
# Meet CPM threshold in 2 samples
# Keep raw counts

hypox_filtered <- process_data(data = pathdb::hypoxia_reads,
                               missing_value = "geneMedian",
                               min_cpm = 0.4,
                               n_min_samples = 2)

# Check filtered data
summary(hypox_filtered)
nrow(hypox_filtered)
```

---

search_species *Search for Species by Name*

Description

Searches the organism database for species matching a query string.

Usage

```
search_species(query, name_type = "all")
```

Arguments

query	Character. The species name, partial name, or ID to search for.
name_type	Character. The type of name to search against. Options: • "all": Default. Searches both primary and academic names. • "academic": Scientific name. (note: not available for all species) • "primary": Primary name in database (common name or academic). • "id": Exact species ID match.

Value

A data frame containing information for all matching species. Throws an error if no species are found.

Examples

```
# Search all names for "Human"
search_species(query = "Human", name_type = "all")

# Search primary names for "Human"
search_species(query = "Human", name_type = "primary")

# Search academic names for "Homo sapiens"
search_species(query = "Homo sapiens", name_type = "academic")

# Search by species ID
search_species(query = 96, name_type = "id")
```

T2G_prep *TERM2GENE Data Prep for iDEP Database*

Description

Prepares background genes for enrichment analysis functions in the format of TERM2GENE data, using pathway information from various databases. Requires ID for a species, and can filter for specific vector of genes.

Usage

```
T2G_prep(species_id = NULL, category = "GOBP", genes = NULL)
```

Arguments

species_id	Numeric. The ID of a desired species from database, found using `srch_species()`
category	Character. A vector or character constant of pathway categories/databases (e.g. KEGG, GOBP, GOCC, etc.). It is not recommended to use all categories, as some species have many, leading to performance issues
genes	Character. A character vector of genes to add to query

Value

A data frame containing TERM2GENE Data (pathways to genes)

Examples

```
# CAUTION: The human database is very large, running these examples require
# the download of the human database.

# Prepare background genes mapping for Hypoxia dataset
# Useful for pathway enrichment analysis of our data
bg_genes <- T2G_prep(
  species_id = 96,
  category = "KEGG",
  genes = rownames(hypoxia_deseq)
)
head(bg_genes)
```

Index