Package ‘DeSciDe’

April 29, 2025

Type Package

Title Tool for Unbiased Literature Searching and Gene List Curation

Version 1.0.0

Description Designed for genomic and proteomic data analysis, enabling unbiased PubMed searching, protein interaction network visualization, and comprehensive data summarization. This package aims to help users identify novel targets within their data sets based on protein network interactions and publication precedence of target's association with research context based on literature precedence. Methods in this package are described in detail in: Douglas (Year) <to-be-added DOI or link to the preprint>. Key functionalities of this package also leverage methodologies from previous works, such as:
- Szklarczyk et al. (2023) <doi:10.1093/nar/gkac1000>
- Winter (2017) <doi:10.32614/RJ-2017-066>.

License MIT + file LICENSE

Encoding UTF-8

Imports rentrez, ComplexHeatmap, circlize, STRINGdb, data.table, igraph, ggplot2, openxlsx, dplyr, tidyr, magrittr, tibble, ggrepel

RoxygenNote 7.3.2

URL `https://github.com/camdouglas/DeSciDe`

BugReports `https://github.com/camdouglas/DeSciDe/issues`

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

VignetteBuilder knitr

Config/testthat/edition 3

Depends R (>= 4.0.0)

NeedsCompilation no

Author Cameron Douglas [aut, cre] (<`https://orcid.org/0000-0003-2334-1652`>)

Maintainer Cameron Douglas <`camerondouglas@ufl.edu`>

Repository CRAN

Date/Publication 2025-04-29 09:00:02 UTC

Contents

`combine_summary` *Combine PubMed and STRING Metrics*

Description

Combine PubMed search summary and STRING gene metrics.

Usage

```
combine_summary(
  pubmed_search_results,
  string_results,
  file_directory = NULL,
  export_format = "csv",
  export = FALSE,
  threshold_percentage = 20
)
```

Arguments

`pubmed_search_results`	Data frame with PubMed search results.
`string_results`	Data frame with STRING metrics.
`file_directory`	Directory for saving the output summary. Defaults to NULL.
`export_format`	Format for export, either "csv", "tsv", or "excel".
`export`	Logical indicating whether to export the summary. Defaults to FALSE.
`threshold_percentage`	Percentage threshold for ranking (default is 20%).

Value

A data frame with combined summary including connectivity, precedence, and category.

Examples

```
pubmed_data <- data.frame(Gene = c("Gene1", "Gene2"), PubMed_Rank = c(1, 2))
string_data <- data.frame(Gene = c("Gene1", "Gene2"), Connectivity_Rank = c(2, 1))
combined <- combine_summary(pubmed_data, string_data, export = FALSE)
print(combined)
```

`descide` *Run DeSciDe pipeline*

Description

Run the entire analysis pipeline including PubMed search, STRING database search, and plotting.

Usage

```
descide(
  genes_list,
  terms_list,
  rank_method = "weighted",
  species = 9606,
  network_type = "full",
  score_threshold = 400,
  threshold_percentage = 20,
  export = FALSE,
  file_directory = NULL,
  export_format = "csv"
)
```

Arguments

`genes_list`	A list of gene IDs.
`terms_list`	A list of search terms.
`rank_method`	The method to rank pubmed results, either "weighted" or "total". Weighted ranks results based on order of terms inputted. Total ranks results on total sum of publications across all search term combinations. Defaults to "weighted".
`species`	The NCBI taxon ID of the species. Defaults to 9606 (Homo sapiens).
`network_type`	The type of string network to use, either "full" or "physical". Defaults to "full".
`score_threshold`	The minimum score threshold for string interactions. Defaults to 400.
`threshold_percentage`	Percentage threshold for ranking (default is 20%).
`export`	Logical indicating whether to export the results. Defaults to FALSE.
`file_directory`	Directory for saving the output files. Defaults to NULL.
`export_format`	Format for export, either "csv", "tsv", or "excel".

Value

A list containing the PubMed search results, STRING results, and summary results.

Examples

```
genes <- c("TP53", "BRCA1")
terms <- c("cancer", "tumor")
results <- descide(genes, terms, export = FALSE)
str(results)
```

`plot_clustering` *Plot STRING Interactions*

Description

Plot STRING interactions degree vs. clustering.

Usage

```
plot_clustering(string_results, file_directory = NULL, export = FALSE)
```

Arguments

`string_results`	Data frame with STRING metrics.
`file_directory`	Directory for saving the output plot. Defaults to NULL.
`export`	Logical indicating whether to export the plot. Defaults to FALSE.

Value

Invisibly returns the ggplot object.

Examples

```
# Example data frame
string_results <- data.frame(Degree = c(10, 5), Clustering_Coefficient_Percent = c(20, 10))
plot_clustering(string_results, file_directory = tempdir(), export = FALSE)
```

`plot_connectivity_precedence` *Plot Connectivity vs. Precedence*

Description

Create a scatter plot of Connectivity Rank vs. PubMed Rank.

Usage

```
plot_connectivity_precedence(
  combined_summary,
  file_directory = NULL,
  export = FALSE
)
```

Arguments

`combined_summary`	Data frame with combined summary including categories.
`file_directory`	Directory for saving the output plot. Defaults to NULL.
`export`	Logical indicating whether to export the plot. Defaults to FALSE.

Value

Invisibly returns a ggplot object.

Examples

```
combined_data <- data.frame(Gene = c("Gene1", "Gene2"), Connectivity_Rank = c(1, 2),
                            PubMed_Rank = c(2, 1),
                           Category = c("High Connectivity - High Precedence", "Other"))
plot_connectivity_precedence(combined_data, export = FALSE)
```

`plot_heatmap` *Plot Heatmap*

Description

Create and optionally save a heatmap of the PubMed search results.

Usage

```
plot_heatmap(pubmed_search_results, file_directory = NULL, export = FALSE)
```

Arguments

`pubmed_search_results`	A data frame containing raw search results with genes and terms.
`file_directory`	Directory for saving the output plot. Defaults to NULL.
`export`	Logical indicating whether to export the plot. Defaults to FALSE.

Value

Invisibly returns a `HeatmapList` object.

Examples

```
# Example data frame
data <- data.frame(Gene = c("Gene1", "Gene2"),
                   Term1 = c(10, 20),
                   Term2 = c(5, 15),
                   Total = c(15, 35),
                   PubMed_Rank = c(1, 2))
plot_heatmap(data, file_directory = tempdir(), export = FALSE)
```

`plot_string_network` *Plot STRING Network*

Description

Plot STRING network interactions using STRINGdb.

Usage

```
plot_string_network(
  string_db,
  string_ids,
  file_directory = NULL,
  export = FALSE
)
```

Arguments

`string_db`	A STRINGdb object.
`string_ids`	A list of STRING IDs.
`file_directory`	Directory for saving the output plot. Defaults to NULL.
`export`	Logical indicating whether to export the plot. Defaults to FALSE.

Value

Invisibly returns NULL.

Examples

```
library(STRINGdb)
string_db <- STRINGdb$new(species = 9606)
string_ids <- c("9606.ENSP00000269305", "9606.ENSP00000357940")
plot_string_network(string_db, string_ids, file_directory = tempdir(), export = FALSE)
```

`rank_search_results` *Rank Search Results*

Description

Rank search results based on a chosen method.

Usage

```
rank_search_results(data, terms_list, rank_method = "weighted")
```

Arguments

`data`	A data frame containing search results.
`terms_list`	A list of search terms.
`rank_method`	The method to rank pubmed results, either "weighted" or "total". Weighted ranks results based on order of terms inputted. Total ranks results on total sum of publications across all search term combinations. Defaults to "weighted".

Value

A data frame with ranked search results, which includes the genes and their corresponding ranks based on the search method.

Examples

```
# Example data frame
data <- data.frame(Gene = c("Gene1", "Gene2"),
                   Term1 = c(10, 20),
                   Term2 = c(5, 15))
terms_list <- c("Term1", "Term2")
ranked_results <- rank_search_results(data, terms_list, rank_method = "weighted")
print(ranked_results)
```

`search_pubmed` *Search PubMed with Multiple Genes and Terms*

Description

Perform a PubMed search for multiple genes and terms.

Usage

```
search_pubmed(genes_list, terms_list, rank_method = "weighted", verbose = TRUE)
```

Arguments

`genes_list`	A list of gene IDs.
`terms_list`	A list of search terms.
`rank_method`	The method to rank results, either "weighted" or "total". Defaults to "weighted".
`verbose`	Logical flag indicating whether to display messages. Default is TRUE.

Value

A data frame with search results, including genes, terms, and their corresponding publication counts and ranks.

Examples

```
genes <- c("TP53", "BRCA1")
terms <- c("cancer", "tumor")
search_results <- search_pubmed(genes, terms, rank_method = "weighted", verbose = FALSE)
print(search_results)
```

`search_string_db` *Search STRING Database*

Description

Search the STRING database for protein interactions.

Usage

```
search_string_db(
  genes_list,
  species = 9606,
  network_type = "full",
  score_threshold = 400
)
```

Arguments

`genes_list`	A list of gene IDs.
`species`	The NCBI taxon ID of the species. Defaults to 9606 (Homo sapiens).
`network_type`	The type of network to use, either "full" or "physical". Defaults to "full".
`score_threshold`	The minimum score threshold for string interactions. Defaults to 400.

Value

A list containing the following elements:

string_results A data frame with STRING interaction metrics.

string_db The STRINGdb object used.

string_ids The STRING IDs for the input genes.

Examples

```
## Not run:
library(STRINGdb)
genes <- c("TP53", "BRCA1")
results <- search_string_db(genes)
print(results)

## End(Not run)
```

`single_pubmed_search` *Search PubMed*

Description

Perform a PubMed search for a given gene and term.

Usage

```
single_pubmed_search(gene, term)
```

Arguments

`gene`	A character string representing the gene symbol.
`term`	A character string representing the search term.

Value

An integer representing the number of PubMed articles found from the search query in PubMed.

Examples

```
# Perform a PubMed search for gene 'TP53' with term 'cancer'
result <- single_pubmed_search("TP53", "cancer")
print(result)
```

Index