Package ‘ontologySimilarity’

March 29, 2024

Type Package
Title Calculating Ontological Similarities
Version 2.7
Encoding UTF-8
Date 2024-03-25
Author Daniel Greene
Maintainer Daniel Greene <dg333@cam.ac.uk>
Description Calculate similarity between ontological terms and sets of ontological terms based on term information content and assess statistical significance of similarity in the context of a collection of terms sets - Greene et al. 2017 <doi:10.1093/bioinformatics/btw763>.
License GPL (>= 2)
Imports Rcpp (>= 1.0.0), ontologyIndex (>= 2.0)
LinkingTo Rcpp
Depends R (>= 3.5.0)
Suggests knitr, rmarkdown, paintmap
VignetteBuilder knitr
RoxygenNote 7.3.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2024-03-29 05:50:02 UTC

R topics documented:

ontologySimilarity-package .................................................. 2
create_sim_index ................................................................. 3
descendants_IC ................................................................. 4
gene_GO_terms ................................................................. 4
get_asym_sim_grid ............................................................. 5
get_profile_sims ............................................................... 5
ontologySimilarity-package

Functions for Calculating Ontological Similarities

Description

Functions for calculating semantic similarities between ontological terms or sets of ontological terms based on term information content and assessing statistical significance of similarity in the context of a collection of sets of ontological terms.

Details

Semantic similarity and similarity significance functions based on Resnik and Lin’s measures of similarity. Computationally intensive functions are written in C++ for performance.

Author(s)

Daniel Greene <dg333@cam.ac.uk>
Maintainer: Daniel Greene <dg333@cam.ac.uk>

References


---

create_sim_index

Create similarity index for list of term sets

Description

Create light-weight similarity index for fast lookups of between term set similarity.

Usage

```r
create_sim_index(
  ontology,
  term_sets,
  information_content = descendants_IC(ontology),
  term_sim_method = "lin",
  combine = "average"
)
```

Arguments

- **ontology**: ontology_index object.
- **term_sets**: List of character vectors of ontological term IDs.
- **information_content**: Numeric vector of information contents of terms (named by term) (optional)
- **term_sim_method**: Character string equalling either "lin" or "resnik" to use Lin or Resnik’s expression for the similarity of terms (optional)
- **combine**: Character string - either "average" or "product", indicating whether to use the best-match-product method, or function accepting two arguments - the first, the similarity matrix obtained by averaging across term sets in term_sets, and the second averaging across those in term_sets2.
Value

Object of class sim_index.

See Also

link{get_sim} get_sim_p sample_group_sim

descendants_IC

Get information content based on number of descendants each term has

Description

Calculate information content of terms based on frequency with which it is an ancestor of other terms. Useful as a default if there is no population frequency information available as it captures the structure of the ontology.

Usage

descendants_IC(ontology)

Arguments

ontology ontology_index object.

Value

Numeric vector of information contents named by term.

gene_GO_terms

Gene Ontology annotation of genes

Description

list object containing character vectors of term IDs of GO terms annotating each gene, named by gene. Users can select a list of annotations for a subset of the annotated genes using a character vector of gene symbols, e.g. gene_GO_terms[c("ACTN1", "TUBB1")], which can then be used in functions for calculating similarities, e.g. get_sim_grid. Note that these annotation vectors contain annotation from all major branches of the Gene Ontology, however one can simply extract the terms only relevant to one by calling the function in the ontologyIndex package: intersection_with_descendants.

Format

List of character vectors.
get_asym_sim_grid

References

get_asym_sim_grid Get asymmetrical similarity matrix

Description
Create a numeric matrix of similarities between two lists of term sets, but only averaging over the terms in sets from A the similarities of the best matches in sets from B.

Usage
get_asym_sim_grid(A, B, ...)

Arguments
A List of term sets.
B List of term sets.
... Other arguments to be passed to get_sim_grid.

Value
Numeric matrix of similarities

See Also
get_sim_grid get_profile_sims

get_profile_sims Get similarities of term sets to profile

Description
Get numeric vector of similarities between each item in a list of term sets and another 'ontological profile', i.e. a single term set. Similarity averaging over terms in term_sets.

Usage
get_profile_sims(profile, term_sets, ...)

get_sim

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>profile</td>
<td>Character vector of term IDs.</td>
</tr>
<tr>
<td>term_sets</td>
<td>List of character vectors of ontological term IDs.</td>
</tr>
<tr>
<td>...</td>
<td>Other arguments to pass to <code>get_sim_grid</code>.</td>
</tr>
</tbody>
</table>

Value

Numeric vector of profile similarities.

See Also

`get_asym_sim_grid`, `get_sim_grid`

Description

Calculates the similarity of a group within a population by applying the function specified by `group_sim` to the pairwise similarities of group members.

Usage

```r
get_sim(pop_sim, ...)  # S3 method for class 'integer'
get_sim(pop_sim, ...)  # S3 method for class 'numeric'
get_sim(pop_sim, group = seq(length(pop_sim)), ...)  # S3 method for class 'matrix'
get_sim(pop_sim, group = seq(nrow(pop_sim)), ...)  # S3 method for class 'matrix'
get_sim(pop_sim, group = seq(pop_sim["N"]), ...)  # S3 method for class 'sim_index'
get_sim(pop_sim, group, type, group_sim = "average", ...)  # Default S3 method:
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pop_sim</td>
<td>An object representing the similarities of an indexed population of objects.</td>
</tr>
<tr>
<td>...</td>
<td>Other arguments to be passed to <code>get_sim</code>.</td>
</tr>
<tr>
<td>group</td>
<td>Character or integer vector specifying names/indices of subgroup for which to calculate a group similarity p-value.</td>
</tr>
</tbody>
</table>
get_similarity_rank_matrix

Description

Given a lower triangular similarity matrix, construct a distance matrix where the rows are the ranks of the column cases with respect to similarity to the row case. If relative similarity is of interest, this rank-transformation may reduce bias in favour of high similarity scores in downstream analysis.

Usage

get_similarity_rank_matrix(similarity_matrix, symmetric = TRUE)

Arguments

similarity_matrix

Lower triangular numeric matrix of similarities, where the rownames and colnames are identical to the case IDs.

symmetric

Logical value determining whether to ‘symmetrify’ resultant matrix by averaging rank similarity of A -> B and B -> A.

Value

Matrix of rank similarities.
get_sim_grid  

Get similarity matrix of pairwise similarities of term sets.

Description

Using either an ontology_index object and numeric vector of information content per term - or a matrix of between-term similarities (e.g. the output of `get_term_sim_mat`), create a numeric matrix of ‘between-term set’ similarities. Either the ‘best-match-average’ or ‘best-match-product’ approach (i.e. where the 2 scores obtained by applying the asymmetric ‘best-match’ similarity function to two term sets in each order are combined by taking the average or the product respectively). Either Lin’s (default) or Resnik’s definition of term similarity can be used. If information_content is not specified, a default value from `descendants_IC` is generated.

Usage

```r
get_sim_grid(
  ontology,   
  information_content,
  term_sim_method,   
  term_sim_mat,   
  term_sets,   
  term_sets2 = term_sets,
  combine = "average"
)
```

Arguments

- **ontology**: ontology_index object.
- **information_content**: Numeric vector of information contents of terms (named by term).
- **term_sim_method**: Character string equalling either "lin" or "resnik" to use Lin or Resnik’s expression for the similarity of terms.
- **term_sim_mat**: Numeric matrix with rows and columns corresponding to (and named by) term IDs, and cells containing the similarity between the row and column term.
- **term_sets**: List of character vectors of ontological term IDs.
- **term_sets2**: Second set of term sets.
- **combine**: Character string - either "average" or "product", indicating whether to use the best-match-product method, or function accepting two arguments - the first, the similarity matrix obtained by averaging across term sets in `term_sets`, and the second averaging across those in `term_sets2`.

Details

Note that if any term set within `term_sets` has 0 terms associated with it, it will get a similarity of 0 to any other set. If you do not want to compare term sets with no annotation, take care to filter out empty sets first, e.g. by `term_sets=term_sets[sapply(term_sets, length) > 0]`.
**Value**

Numeric matrix of pairwise term set similarities.

**See Also**

`get_term_sim_mat` `get_sim_p` `get_asym_sim_grid`

**Examples**

```r
library(ontologyIndex)
data(hpo)
term_sets <- list(
  'case1'=c("HP:0001873", "HP:0011877"),
  'case2'=c("HP:0001872", "HP:0001892"),
  'case3'="HP:0001873")
get_sim_grid(ontology=hpo, term_sets=term_sets)
```

---

**get_sim_p**

*Get similarity p-value*

**Description**

p-value of group similarity, calculated by estimating the proportion by random sampling of groups the same size as group which have at least as great group similarity than does group.

**Usage**

```r
get_sim_p(pop_sim, ...)
```

## S3 method for class 'integer'

```r
get_sim_p(pop_sim, ...)
```

## S3 method for class 'numeric'

```r
get_sim_p(pop_sim, group, ...)
```

## S3 method for class 'matrix'

```r
get_sim_p(pop_sim, group, ...)
```

## S3 method for class 'sim_index'

```r
get_sim_p(pop_sim, group, ...)
```

## Default S3 method:

```r
general_sim_p(
  pop_sim,
  group,
  type,
  min_its = 1000,
```
get_sim_p_from_ontology

```
max_its = 1e+05,
signif = 0.05,
log_dismiss = log(1e-06),
group_sim = "average",
...
```

**Arguments**

- `pop_sim` An object representing the similarities of an indexed population of objects.
- `group` Character or integer vector specifying names/indices of subgroup for which to calculate a group similarity p-value.
- `type` Either "matrix", "sim_index" or "numeric" - the type of the `pop_sim` object.
- `min_its` Minimum number of simulated group similarities to calculate
- `max_its` Maximum number of simulated group similarities to calculate
- `signif` Threshold p-value of statistical significance
- `log_dismiss` Threshold of log probability, below which to trigger return of current estimated p-value
- `group_sim` String Either "average" or "min", determining how to calculate the similarity of a group of term sets over all pairwise combinations of group members

**Value**

p-value.

**See Also**

- `get_sim`
- `sample_group_sim`

---

**get_sim_p_from_ontology**

*Get similarity p-value for subgroup term sets*

**Description**

Compute a similarity p-value by permutation for subgroup of a list of term sets

**Usage**

```
get_sim_p_from_ontology(
    ontology,
    term_sets,
    information_content = descendants_IC(ontology),
    term_sim_method = "lin",
    combine = "average",
    ...
)
```
**get_term_set_to_term_sims**

**Arguments**

- **ontology** ontology_index object.
- **term_sets** List of character vectors of ontological term IDs.
- **information_content** Numeric vector of information contents of terms (named by term).
- **term_sim_method** Character string equalling either "lin" or "resnik" to use Lin or Resnik's expression for the similarity of terms.
- **combine** Character string - either "average" or "product", indicating whether to use the best-match-product’ method, or function accepting two arguments - the first, the similarity matrix obtained by averaging across term sets in term_sets, and the second averaging across those in term_sets2.
- **...** Other arguments to be passed to get_sim_p.

**Value**

Numeric value.

**See Also**

get_sim_p create_sim_index

---

**get_term_set_to_term_sims**

*Get 'term sets to term' similarity matrix*

**Description**

Create a numeric matrix of similarities between term sets and individual terms.

**Usage**

`get_term_set_to_term_sims(term_sets, terms, ...)`

**Arguments**

- **term_sets** List of character vectors of ontological term IDs.
- **terms** Character vector of ontological terms.
- **...** Other arguments to be passed to get_sim_grid.

**Value**

Numeric matrix of term set-to-term similarities

**See Also**

get_sim_grid
get_term_sim_mat

Get term-term similarity matrix

Description

Get matrix of pairwise similarity of individual terms based on Lin’s (default) or Resnik’s information content-based expression.

Usage

get_term_sim_mat(
  ontology,
  information_content,
  method = "lin",
  row_terms = names(information_content),
  col_terms = names(information_content)
)

Arguments

ontology ontology_index object.
information_content Numeric vector of information contents of terms (named by term)
method Character value equalling either "lin" or "resnik" to use Lin or Resnik’s expression for similarity of terms respectively.
row_terms Character vector of term IDs to appear as rows of result matrix.
col_terms Character vector of term IDs to appear as cols of result matrix.

Value

Numeric matrix of pairwise term similarities.

See Also

get_sim_grid resnik, lin
GO_IC

Gene Ontology terms information content.

Description

Numeric vector containing the information content of Gene Ontology terms based on frequencies of annotation data object gene_GO_terms. The object can be derived using the function get_term_info_content and data object go from the ontologyIndex package.

Format

List of character vectors.

group_term_enrichment  Identify enriched terms in subgroup

Description

Create a table of terms ranked by their significance of occurrence in a set of term sets amongst an enclosing set, with p-values computed by permutation. Terms are subselected so that only the minimal set of non-redundant terms at each level of frequency within the group are retained.

Usage

```r
group_term_enrichment(
  ontology,
  term_sets,
  group,
  permutations = 1000L,
  min_terms = 2L,
  mc.cores = NULL
)
```

Arguments

- **ontology**: ontology_index object.
- **term_sets**: List of character vectors of ontological term IDs.
- **group**: Integer/logical/character vector specifying indices/positions/names of subgroup for which to calculate a group similarity p-value.
- **permutations**: Number of permutations to test against, or if NULL, perform no permutations and return the unadjusted p-values for the occurrence of each term.
- **min_terms**: Minimum number of times a term should occur within the given group to be eligible for inclusion in the results.
- **mc.cores**: If not null and greater than on, the number of cores use calculating permutations (passed to mclapply).
Value

data.frame containing columns: term (with the term ID); name (term readable name); in_term (number of sets in the given group containing the term); in_no_term (number of sets in the given group not containing the term); out_term and out_no_term (equivalently for the sets not in the given group); p (the p-values calculated by permutation for seeing a term with such a strong association, measured using Fisher’s exact test, in a group of term sets the size of the given group among term_sets). Rows ordered by significance (i.e. the p columns).

See Also

sample_group_sim create_sim_index

Description

Warning! This function is slow - performing large numbers of ‘between term-set’ similarity calculations should be done using get_sim_grid.

Usage

lin(ontology, information_content, term_set_1, term_set_2)

Arguments

ontology ontology_index object.
information_content Numeric vector of information contents of terms (named by term)
term_set_1 Character vector of terms.
term_set_2 Character vector of terms.

Value

Numeric value.

References


See Also

resnik, get_term_sim_mat
resnik  

Calculate Resnik similarity score of two term sets

Description

Warning! This function is slow - performing large numbers of ‘between term-set’ similarity calculations should be done using get_sim_grid.

Usage

resnik(ontology, information_content, term_set_1, term_set_2)

Arguments

  ontology  ontology_index object.
  information_content  Numeric vector of information contents of terms (named by term)
  term_set_1  Character vector of terms.
  term_set_2  Character vector of terms.

Value

Numeric value.

References


See Also

lin, get_term_sim_mat

sample_group_sim  

Draw sample of group similarities

Description

Draw sample of group similarities of groups of given size
Usage

```r
sample_group_sim(pop_sim, ...)
```

## S3 method for class 'integer'

```r
sample_group_sim(pop_sim, ...)
```

## S3 method for class 'numeric'

```r
sample_group_sim(pop_sim, ...)
```

## S3 method for class 'matrix'

```r
sample_group_sim(pop_sim, ...)
```

## S3 method for class 'sim_index'

```r
sample_group_sim(pop_sim, ...)
```

## Default S3 method:

```r
sample_group_sim(
  pop_sim,
  type,
  group_size,
  group_sim = "average",
  sample_size = 10000,
  ...
)
```

Arguments

- `pop_sim` An object representing the similarities of an indexed population of objects.
- `...` Other arguments to be passed to `sample_group_sim`.
- `type` Either "matrix", "sim_index" or "numeric" - the type of the `pop_sim` object.
- `group_size` Integer giving the number of members of a group.
- `group_sim` String Either "average" or "min", determining how to calculate the similarity of a group of term sets over all pairwise combinations of group members
- `sample_size` Number of samples to draw.

Value

Numeric vector of random group similarities.

See Also

- `get_sim`  
- `get_sim_p`
sample_group_sim_from_ontology

*Draw sample of group similarities*

**Description**
ample of group similarities for random groups of given drawn from the given ontology argument

**Usage**

```r
sample_group_sim_from_ontology(
  ontology,
  term_sets,
  information_content = descendants_IC(ontology),
  term_sim_method = "lin",
  combine = "average",
  ...
)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ontology</td>
<td>ontology_index object.</td>
</tr>
<tr>
<td>term_sets</td>
<td>List of character vectors of ontological term IDs.</td>
</tr>
<tr>
<td>information_content</td>
<td>Numeric vector of information contents of terms (named by term)</td>
</tr>
<tr>
<td>term_sim_method</td>
<td>Character string equalling either &quot;lin&quot; or &quot;resnik&quot; to use Lin or Resnik's expression for the similarity of terms.</td>
</tr>
<tr>
<td>combine</td>
<td>Character string - either &quot;average&quot; or &quot;product&quot;, indicating whether to use the best-match-product method, or function accepting two arguments - the first, the similarity matrix obtained by averaging across term sets in term_sets, and the second averaging across those in term_sets2.</td>
</tr>
<tr>
<td>...</td>
<td>Other arguments to be passed to <code>get_sim_p</code>.</td>
</tr>
</tbody>
</table>

**Value**

Numeric vector of group similarities.

**See Also**

*sample_group_sim create_sim_index*
Index

* GO
  ontologySimilarity-package, 2

* HPO
  ontologySimilarity-package, 2

* ontological similarity
  ontologySimilarity-package, 2

* ontology
  ontologySimilarity-package, 2

* semantic similarity
  ontologySimilarity-package, 2

create_sim_index, 3, 11, 14, 17

descendants_IC, 4, 8

gene_GO_terms, 4
get_asym_sim_grid, 5, 6, 9
get_profile_sims, 5, 5
get_sim, 6, 10, 16
get_sim_grid, 4–6, 8, 11, 12, 14, 15
get_sim_p, 4, 7, 9, 9, 11, 16, 17
get_sim_p_from_ontology, 10
get_similarity_rank_matrix, 7
get_term_set_to_term_sims, 11
get_term_sim_mat, 8, 9, 12, 14, 15
GO_IC, 13
group_term_enrichment, 13

lin, 12, 14, 15

ontologySimilarity
  (ontologySimilarity-package), 2
ontologySimilarity-package, 2

resnik, 12, 14, 15

sample_group_sim, 4, 7, 10, 14, 15, 17
sample_group_sim_from_ontology, 17