Package ‘GOSemSim’

May 19, 2026

Type Package

Title GO-terms Semantic Similarity Measures

Version 2.39.0

Maintainer Guangchuang Yu <guangchuangyu@gmail.com>

Description The semantic comparisons of Gene Ontology (GO) annotations provide quantitative ways to compute similarities between genes and gene groups, and have became important basis for many bioinformatics analysis approaches. GOSemSim is an R package for semantic similarity computation among GO terms, sets of GO terms, gene products and gene clusters. GOSemSim implemented five methods proposed by Resnik, Schlicker, Jiang, Lin and Wang respectively.

Depends R (>= 4.2.0)

LinkingTo Rcpp

Imports AnnotationDbi, DBI, digest, GO.db, methods, rlang, stats, utils, yulab.utils (>= 0.2.3)

Suggests AnnotationHub, BiocManager, clusterProfiler, DOSE, knitr, org.Hs.eg.db, prettydoc, readr, rmarkdown, testthat, tidyr, tidyselect, ROCR

VignetteBuilder knitr

ByteCompile true

License Artistic-2.0

Encoding UTF-8

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

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

biocViews Annotation, GO, Clustering, Pathways, Network, Software

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

git_url https://git.bioconductor.org/packages/GOSemSim

git_branch devel

git_last_commit e3e9db0

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-05-18

Author Guangchuang Yu [aut, cre],
Alexey Stukalov [ctb],
Pingfan Guo [ctb],
Chuanle Xiao [ctb],
Lluís Revilla Sancho [ctb]

Contents

GOSemSim-package 2
buildGOmap 3
clusterSim 3
combineScores 4
geneSim 5
get_organism 6
godata 6
GOSemSimDATA-class 7
goSim 8
go_term_table 8
infoContentMethod 9
load_onto 9
mclusterSim 10
mgeneSim 11
mgoSim 12
read.blast2go 13
read.gaf 13
reexports 14
tcss_cutoff 14
termSim 15
wangMethod_internal 16

Index 17

`GOSemSim-package` *GOSemSim: GO-terms Semantic Similarity Measures*

Description

The semantic comparisons of Gene Ontology (GO) annotations provide quantitative ways to compute similarities between genes and gene groups, and have became important basis for many bioinformatics analysis approaches. GOSemSim is an R package for semantic similarity computation among GO terms, sets of GO terms, gene products and gene clusters. GOSemSim implemented five methods proposed by Resnik, Schlicker, Jiang, Lin and Wang respectively.

Author(s)

Maintainer: Guangchuang Yu <`guangchuangyu@gmail.com`>

Other contributors:

- Alexey Stukalov <`astukalov@gmail.com`> [contributor]
- Pingfan Guo <`1178431277@qq.com`> [contributor]
- Chuanle Xiao <`xiaochuanle@126.com`> [contributor]
- Lluís Revilla Sancho <`lluis.revilla@gmail.com`> [contributor]

See Also

Useful links:

- https://yulab-smu.top/biomedical-knowledge-mining-book/
- Report bugs at https://github.com/YuLab-SMU/GOSemSim/issues

---

`buildGOmap` *buildGOmap*

Description

Addding indirect GO annotation

Usage

```
buildGOmap(TERM2GENE)
```

Arguments

`TERM2GENE`	data.frame with two or three columns of GO TERM, GENE and ONTOLOGY (optional)

Details

provided by a data.frame of GO TERM (column 1), GENE (column 2) and ONTOLOGY (optional) that describes GO direct annotation, this function will add indirect GO annotation of genes.

Value

data.frame, GO annotation with direct and indirect annotation

Author(s)

Yu Guangchuang

`clusterSim` *Semantic similarity between two gene clusters*

Description

Semantic similarity between two gene clusters

Usage

```
clusterSim(
  cluster1,
  cluster2,
  semData,
  measure = "Wang",
  drop = "IEA",
  combine = "BMA"
)
```

Arguments

`cluster1`	A set of gene IDs
`cluster2`	Another set of gene IDs
`semData`	GOSemSimDATA object
`measure`	One of "Resnik", "Lin", "Rel", "Jiang", "TCSS" and "Wang" methods.
`drop`	Evidence codes to drop; use `NULL` to keep all GO annotations
`combine`	One of "max", "avg", "rcmax", "BMA" methods, used to combine multiple term scores.

Value

similarity

Author(s)

Guangchuang Yu `https://yulab-smu.top`

See Also

`goSim()` `mgoSim()` `geneSim()` `mgeneSim()` `clusterSim()` `mclusterSim()`

Examples

```
d <- godata('org.Hs.eg.db', ont = "MF", computeIC = FALSE)
cluster1 <- c("835", "5261", "241", "994")
cluster2 <- c("307", "308", "317", "321", "506", "540", "378", "388", "396")
clusterSim(cluster1, cluster2, semData = d, measure = "Wang")
```

`combineScores` *combining similarity matrix to similarity score*

Description

Functions for combining similarity matrix to similarity score

Usage

```
combineScores(SimScores, combine)
```

Arguments

`SimScores`	similarity matrix
`combine`	combine method

Value

similarity value

Author(s)

Guangchuang Yu `https://yulab-smu.top`

`geneSim` *Semantic similarity between two genes*

Description

Given two genes, calculate their semantic similarity and return the score with corresponding GO terms.

Usage

```
geneSim(gene1, gene2, semData, measure = "Wang", drop = "IEA", combine = "BMA")
```

Arguments

`gene1`	Entrez gene ID
`gene2`	Another Entrez gene ID
`semData`	GOSemSimDATA object
`measure`	One of "Resnik", "Lin", "Rel", "Jiang", "TCSS" and "Wang" methods.
`drop`	Evidence codes to drop; use `NULL` to keep all GO annotations
`combine`	One of "max", "avg", "rcmax", "BMA" methods, used to combine multiple term scores.

Value

A list containing similarity value and corresponding GO terms

Author(s)

Guangchuang Yu `https://yulab-smu.top`

See Also

`goSim()` `mgoSim()` `mgeneSim()` `clusterSim()` `mclusterSim()`

Examples

```
d <- godata('org.Hs.eg.db', ont = "MF", computeIC = FALSE)
geneSim("241", "251", semData = d, measure = "Wang")
```

`get_organism` *Get organism name from OrgDb object*

Description

Get organism name from OrgDb object

Usage

```
get_organism(object)
```

Arguments

`object`	OrgDb object or OrgDb package name

Value

Organism name

Author(s)

Guangchuang Yu

`godata` *godata*

Description

prepare GO DATA for measuring semantic similarity

Usage

```
godata(
  OrgDb = NULL,
  annoDb = NULL,
  keytype = "ENTREZID",
  ont,
  computeIC = TRUE,
  processTCSS = FALSE,
  cutoff = NULL
)
```

Arguments

`OrgDb`	OrgDb object (will be removed in future, please use annoDb instead)
`annoDb`	GO annotation database, can be OrgDb or a data.frame contains three columns of 'GENE', 'GO' and 'ONTOLOGY'.
`keytype`	keytype
`ont`	one of 'BP', 'MF', 'CC'
`computeIC`	logical, whether computer IC
`processTCSS`	logical, whether to process TCSS
`cutoff`	cutoff of TCSS

Value

GOSemSimDATA object

Author(s)

Guangchuang Yu

`GOSemSimDATA-class` *Class "GOSemSimDATA" This class stores IC and gene to go mapping for semantic similarity measurement*

Description

Class "GOSemSimDATA" This class stores IC and gene to go mapping for semantic similarity measurement

Slots

`keys`	gene ID
`ont`	ontology
`IC`	IC data
`geneAnno`	gene to GO mapping
`tcssdata`	tcssdata
`metadata`	metadata

`goSim` *Semantic similarity between two GO terms*

Description

Given two GO IDs, calculate their semantic similarity.

Usage

```
goSim(GOID1, GOID2, semData, measure = "Wang")
```

Arguments

`GOID1`	GO ID 1
`GOID2`	GO ID 2
`semData`	GOSemSimDATA object
`measure`	One of "Resnik", "Lin", "Rel", "Jiang", "TCSS" and "Wang" methods.

Value

similarity

Author(s)

Guangchuang Yu `https://yulab-smu.top`

See Also

`goSim()` `mgoSim()` `geneSim()` `mgeneSim()` `clusterSim()` `mclusterSim()`

Examples

```
d <- godata('org.Hs.eg.db', ont = "MF", computeIC = FALSE)
goSim("GO:0004022", "GO:0005515", semData = d, measure = "Wang")
```

`go_term_table` *Information content of GO terms*

Description

These datasets are the information contents of GOterms.

References

Yu et al. (2010) GOSemSim: an R package for measuring semantic similarity among GO terms and gene products *Bioinformatics* (Oxford, England), 26:7 976–978, April 2010. ISSN 1367-4803 `http://bioinformatics.oxfordjournals.org/cgi/content/abstract/26/7/976` PMID: 20179076

`infoContentMethod` *information content based methods*

Description

Information Content Based Methods for semantic similarity measuring

Usage

```
infoContentMethod(ID1, ID2, method, godata)
```

Arguments

`ID1`	Ontology Term
`ID2`	Ontology Term
`method`	one of "Resnik", "Jiang", "Lin" and "Rel", "TCSS".
`godata`	GOSemSimDATA object

Details

implemented for methods proposed by Resnik, Jiang, Lin and Schlicker.

Value

semantic similarity score

Author(s)

Guangchuang Yu `https://yulab-smu.top`

`load_onto` *Load Ontology Database*

Description

Load Ontology Database

Usage

```
load_onto(onto = "HDO")
```

Arguments

`onto`	character. The ontology to load (e.g., "HDO").

Value

An `AnnotationDb` object.

`mclusterSim` *Pairwise semantic similarities for a list of gene clusters*

Description

Calculate pairwise semantic similarities for a list of gene clusters.

Usage

```
mclusterSim(clusters, semData, measure = "Wang", drop = "IEA", combine = "BMA")
```

Arguments

`clusters`	A list of gene clusters
`semData`	GOSemSimDATA object
`measure`	One of "Resnik", "Lin", "Rel", "Jiang", "TCSS" and "Wang" methods.
`drop`	Evidence codes to drop; use `NULL` to keep all GO annotations
`combine`	One of "max", "avg", "rcmax", "BMA" methods, used to combine multiple term scores.

Value

similarity matrix

Author(s)

Guangchuang Yu `https://yulab-smu.top`

See Also

`goSim()` `mgoSim()` `geneSim()` `mgeneSim()` `clusterSim()` `mclusterSim()`

Examples

```
d <- godata('org.Hs.eg.db', ont = "MF", computeIC = FALSE)
cluster1 <- c("835", "5261", "241")
cluster2 <- c("578", "582")
cluster3 <- c("307", "308", "317")
clusters <- list(a = cluster1, b = cluster2, c = cluster3)
mclusterSim(clusters, semData = d, measure = "Wang")
```

`mgeneSim` *Pairwise semantic similarity for a list of genes*

Description

Calculate pairwise semantic similarities for a given list of genes.

Usage

```
mgeneSim(
  genes,
  semData,
  measure = "Wang",
  drop = "IEA",
  combine = "BMA",
  verbose = TRUE
)
```

Arguments

`genes`	A list of Entrez gene IDs
`semData`	GOSemSimDATA object
`measure`	One of "Resnik", "Lin", "Rel", "Jiang", "TCSS" and "Wang" methods.
`drop`	Evidence codes to drop; use `NULL` to keep all GO annotations
`combine`	One of "max", "avg", "rcmax", "BMA" methods, used to combine multiple term scores.
`verbose`	Whether to show a progress bar

Value

similarity matrix

Author(s)

Guangchuang Yu `https://yulab-smu.top`

See Also

`goSim()` `mgoSim()` `geneSim()` `mgeneSim()` `clusterSim()` `mclusterSim()`

Examples

```
d <- godata('org.Hs.eg.db', ont = "MF", computeIC = FALSE)
mgeneSim(c("835", "5261", "241"), semData = d, measure = "Wang")
```

`mgoSim` *Semantic similarity between two GO term sets*

Description

Given two sets of GO terms, calculate their semantic similarity.

Usage

```
mgoSim(GO1, GO2, semData, measure = "Wang", combine = "BMA")
```

Arguments

`GO1`	A set of GO terms
`GO2`	Another set of GO terms
`semData`	GOSemSimDATA object
`measure`	One of "Resnik", "Lin", "Rel", "Jiang", "TCSS" and "Wang" methods.
`combine`	One of "max", "avg", "rcmax", "BMA" methods, used to combine multiple term scores.

Value

similarity

Author(s)

Guangchuang Yu https://yulab-smu.top

See Also

`goSim()` `mgoSim()` `geneSim()` `mgeneSim()` `clusterSim()` `mclusterSim()`

Examples

```
d <- godata('org.Hs.eg.db', ont = "MF", computeIC = FALSE)
go1 <- c("GO:0004022", "GO:0004024", "GO:0004023")
go2 <- c("GO:0009055", "GO:0020037")
mgoSim("GO:0003824", go2, semData = d, measure = "Wang")
mgoSim(go1, go2, semData = d, measure = "Wang")
```

`read.blast2go` *read.blast2go*

Description

given a BLAST2GO file, this function extracts the information from it and make it use for TERM2GENE.

Usage

```
read.blast2go(file, add_indirect_GO = FALSE)
```

Arguments

`file`	BLAST2GO file
`add_indirect_GO`	whether add indirect GO annotation

Value

a data frame with three columns: GENE, GO and ONTOLOGY

`read.gaf` *read.gaf*

Description

parse GAF files

Usage

```
read.gaf(file, asis = FALSE, add_indirect_GO = FALSE)

parse_gff(file, asis = FALSE, add_indirect_GO = FALSE)
```

Arguments

`file`	GAF file
`asis`	logical, whether output the original contains of the file and only works if 'add_indirect_GO = FALSE'
`add_indirect_GO`	whether to add indirect GO annotation

Details

given a GAF file, this function extracts the information from it

Value

A data.frame. Original table if 'asis' works, otherwise contains 3 conlumns of 'GENE', 'GO' and 'ONTOLOGY'

`reexports` *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

yulab.utils `load_OrgDb`

`tcss_cutoff` *determine the topological cutoff for TCSS method*

Description

determine the topological cutoff for TCSS method

Usage

```
tcss_cutoff(
  OrgDb = NULL,
  keytype = "ENTREZID",
  ont,
  combine_method = "max",
  ppidata
)
```

Arguments

`OrgDb`	OrgDb object
`keytype`	keytype
`ont`	ontology : "BP", "MF", "CC"
`combine_method`	"max", "BMA", "avg", "rcmax", "rcmax.avg"
`ppidata`	A data.frame contains positive set and negative set. Positive set is PPI pairs that already verified. ppidata has three columns, column 1 and 2 are character, column 3 must be logical value:TRUE/FALSE.

Value

numeric, topological cutoff for given parameters

Examples

```
## Not run:
    library(org.Hs.eg.db)
    library(STRINGdb)

    string_db <- STRINGdb$new(version = "11.0", species = 9606,
    score_threshold = 700)
    string_proteins <- string_db$get_proteins()

    #get relationship
    ppi <- string_db$get_interactions(string_proteins$protein_external_id)

    ppi$from <- vapply(ppi$from, function(e)
                       strsplit(e, "9606.")[[1]][2], character(1))
    ppi$to <- vapply(ppi$to, function(e)
                       strsplit(e, "9606.")[[1]][2], character(1))
    len <- nrow(ppi)

    #select length
    s_len <- 100
    pos_1 <- sample(len, s_len, replace = T)
    #negative set
    pos_2 <- sample(len, s_len, replace = T)
    pos_3 <- sample(len, s_len, replace = T)
    #union as ppidata
    ppidata <- data.frame(pro1 = c(ppi$from[pos_1], ppi$from[pos_2]),
     pro2 = c(ppi$to[pos_1], ppi$to[pos_3]),
     label = c(rep(TRUE, s_len), rep(FALSE, s_len)),
     stringsAsFactors = FALSE)

    cutoff <- tcss_cutoff(OrgDb = org.Hs.eg.db, keytype = "ENSEMBLPROT",
    ont = "BP", combine_method = "max", ppidata)

## End(Not run)
```

`termSim` *termSim*

Description

Measure similarities between two term vectors.

Usage

```
termSim(
  t1,
  t2,
  semData,
  method = c("Wang", "Resnik", "Rel", "Jiang", "Lin", "TCSS")
)
```

Arguments

`t1`	Term vector
`t2`	Term vector
`semData`	GOSemSimDATA object
`method`	One of "Wang", "Resnik", "Rel", "Jiang", "Lin", "TCSS"

Details

Provide two term vectors, this function calculates their similarities.

Value

score matrix

Author(s)

Guangchuang Yu `https://yulab-smu.top`

`wangMethod_internal` *wangMethod*

Description

Method Wang for semantic similarity measuring

Usage

```
wangMethod_internal(ID1, ID2, ont = "BP")
```

Arguments

`ID1`	Ontology Term
`ID2`	Ontology Term
`ont`	Ontology

Value

semantic similarity score

Author(s)

Guangchuang Yu `https://yulab-smu.top`

Index

∗ **classes**
    GOSemSimDATA-class, 7
∗ **datasets**
    go_term_table, 8
∗ **internal**
    GOSemSim-package, 2
    load_onto, 9
    reexports, 14

buildGOmap, 3

clusterSim, 3
clusterSim(), *4, 5*, *8*, *10–12*
combineScores, 4

geneSim, 5
geneSim(), *4*, *8*, *10–12*
get_organism, 6
GO *(*go_term_table*)*, 8
go_term_table, 8
godata, 6
GOSemSim *(*GOSemSim-package*)*, 2
GOSemSim-package, 2
GOSemSimDATA-class, 7
goSim, 8
goSim(), *4, 5*, *8*, *10–12*
gotbl *(*go_term_table*)*, 8

infoContentMethod, 9

load_onto, 9
load_OrgDb, *14*
load_OrgDb *(*reexports*)*, 14

mclusterSim, 10
mclusterSim(), *4, 5*, *8*, *10–12*
mgeneSim, 11
mgeneSim(), *4, 5*, *8*, *10–12*
mgoSim, 12
mgoSim(), *4, 5*, *8*, *10–12*

parse_gff *(*read.gaf*)*, 13

read.blast2go, 13
read.gaf, 13
reexports, 14

show,GOSemSimDATA-method
        *(*GOSemSimDATA-class*)*, 7

tcss_cutoff, 14
termSim, 15

wangMethod_internal, 16