

# Package ‘ViSEAGO’

October 17, 2020

**Version** 1.2.0

**Title** ViSEAGO: a Bioconductor package for clustering biological functions using Gene Ontology and semantic similarity

**Author** Aurelien Brionne [aut, cre],  
Amelie Juanchich [aut],  
Christelle hennequet-antier [aut]

**Maintainer** Aurelien Brionne <aurelien.brionne@inra.fr>

**Imports** data.table, AnnotationDbi, AnnotationForge, biomaRt, dendextend, DiagrammeR, DT, dynamicTreeCut, GOSemSim, ggplot2, GO.db, grDevices, heatmaply, htmlwidgets, igraph, methods, plotly, topGO, RColorBrewer, R.utils, scales, stats, UpSetR, utils, webshot

**Depends** R (>= 3.6)

**Suggests** htmltools, org.Mm.eg.db, limma, Rgraphviz, BiocStyle, knitr, rmarkdown, corplot, remotes, BiocManager

**Description** The main objective of ViSEAGO package is to carry out a data mining of biological functions and establish links between genes involved in the study. We developed ViSEAGO in R to facilitate functional Gene Ontology (GO) analysis of complex experimental design with multiple comparisons of interest. It allows to study large-scale datasets together and visualize GO profiles to capture biological knowledge. The acronym stands for three major concepts of the analysis: Visualization, Semantic similarity and Enrichment Analysis of Gene Ontology. It provides access to the last current GO annotations, which are retrieved from one of NCBI Entrez-Gene, Ensembl or Uniprot databases for several species. Using available R packages and novel developments, ViSEAGO extends classical functional GO analysis to focus on functional coherence by aggregating closely related biological themes while studying multiple datasets at once. It provides both a synthetic and detailed view using interactive functionalities respecting the GO graph structure and ensuring functional coherence supplied by semantic similarity. ViSEAGO has been successfully applied on several datasets from different species with a variety of biological questions. Results can be easily shared between bioinformaticians and biologists, enhancing reporting capabilities while maintaining reproducibility.

**VignetteBuilder** knitr

**License** GPL-3

**Repository** [forgemia.inra.fr](http://forgemia.inra.fr), [bioconductor.org](http://bioconductor.org)

**URL** <https://www.bioconductor.org/packages/release/bioc/html/ViSEAGO.html>,  
<https://forgemia.inra.fr/UMR-BOA/ViSEAGO>

**BugReports** <https://forgemia.inra.fr/UMR-BOA/ViSEAGO/issues>

**biocViews** Software, Annotation, GO, GeneSetEnrichment,  
MultipleComparison, Clustering, Visualization

**RoxygenNote** 7.0.2

**Collate** 'genomic\_ressource.R' 'Bioconductor2GO.R' 'Ensembl2GO.R'  
'EntrezGene2GO.R' 'EntrezGene\_orthologs.R' 'enrich\_GO\_terms.R'  
'GO\_SS.R' 'GO\_clusters.R' 'GOclusters\_heatmap.R' 'GOcount.R'  
'GOTerms\_heatmap.R' 'MDSplot.R' 'Uniprot2GO.R' 'Upset.R'  
'ViSEAGO.R' 'annotate.R' 'available\_organisms.R' 'gene2GO.R'  
'build\_GO\_SS.R' 'clusters\_cor.R' 'compare\_clusters.R'  
'compute\_SS\_distances.R' 'create\_topGOdata.R' 'custom2GO.R'  
'datasets.R' 'merge\_enrich\_terms.R' 'overlapper.R'  
'pkgdiagram.R' 'show\_heatmap.R' 'show\_table.R' 'taxonomy.R'

**git\_url** <https://git.bioconductor.org/packages/ViSEAGO>

**git\_branch** RELEASE\_3\_11

**git\_last\_commit** 22f848a

**git\_last\_commit\_date** 2020-04-27

**Date/Publication** 2020-10-16

## R topics documented:

annotate	3
available_organisms	5
Bioconductor2GO	6
build_GO_SS	7
clusters_cor	8
compare_clusters	11
compute_SS_distances	14
create_topGOdata	16
Custom2GO	18
enrich_GO_terms-class	19
Ensembl2GO	19
EntrezGene2GO	20
gene2GO-class	21
genomic_ressource-class	22
GOclusters_heatmap	22
GOcount	25
GOTerms_heatmap	26
GO_clusters-class	29
GO_SS-class	29
MDSplot	30
merge_enrich_terms	31
myGOs	34
show_heatmap	35
show_table	37
Uniprot2GO	39
Upset	40
ViSEAGO	41

**Index** 42

---

annotate	<i>Retrieve GO annotations for a specie from genomic ressource database.</i>
----------	--

---

### Description

This method retrieves and stores GO annotations for the organism of interest from one of genomic ressource database (Bioconductor, EntrezGene, Ensembl, Uniprot).

### Usage

```
annotate(id, object, ortholog = FALSE)

## S4 method for signature 'character,genomic_ressource'
annotate(id, object, ortholog = FALSE)
```

### Arguments

id	identifiant corresponding to the organism of interest. This id name is referenced in the first column of the database used (see <a href="#">available_organisms</a> ).
object	a required <a href="#">genomic_ressource-class</a> object created by <a href="#">Bioconductor2GO</a> , <a href="#">EntrezGene2GO</a> , <a href="#">Ensembl2GO</a> , or <a href="#">Uniprot2GO</a> methods.
ortholog	logical (default to FALSE). Only available for vertebrates organisms and for object created by <a href="#">EntrezGene2GO</a> method (see Details).

### Details

This method uses a [genomic\\_ressource-class](#) object to retrieve GO annotations for the organism of interest. The stored annotations are structured in 3 slots corresponding to the 3 GO categories: MF (Molecular Function), BP (Biological Process), and CC (Cellular Component). Each slot contains GO terms with associated [evidence code](#).

The [genomic\\_ressource-class](#) object is created by one of the four available methods: [Bioconductor2GO](#), [EntrezGene2GO](#), [Ensembl2GO](#), or [Uniprot2GO](#).

In the case of vertebrates, setting ortholog argument to TRUE is required if you need to add GO terms with experimental [evidence codes](#) from orthologs genes when using [EntrezGene2GO](#) method. To display organisms supported by NCBI EntrezGene orthologs pipeline, set the arguments id=NULL and ortholog=TRUE. This approach is highly similar to the strategy developed by Uniprot-GOA consortium for the Electronic Annotation Method using [Ensembl Compara](#).

### Value

annotate produces an object of [gene2GO-class](#) required by [build\\_GO\\_SS](#) method.

### References

- Durinck S, Spellman P, Birney E and Huber W (2009). Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt. *Nature Protocols*, 4, pp. 1184-1191.
- Durinck S, Moreau Y, Kasprzyk A, Davis S, De Moor B, Brazma A and Huber W (2005). BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinformatics*, 21, pp. 3439-3440.

Fong, JH, Murphy, TD, Pruitt, KD (2013). Comparison of RefSeq protein-coding regions in human and vertebrate genomes. *BMC Genomics*, 14:654.

Henrik Bengtsson (2016). R.utils: Various Programming Utilities. R package version 2.5.0. <https://CRAN.R-project.org/package=R.utils>.

Herve Pages, Marc Carlson, Seth Falcon and Nianhua Li (2017). AnnotationDbi: Annotation Database Interface. R package version 1.38.0.

Matt Dowle and Arun Srinivasan (2017). data.table: Extension of data.frame. R package version 1.10.4. <https://CRAN.R-project.org/package=data.table>.

### See Also

Other genomic\_ressource: [Bioconductor2GO\(\)](#), [Custom2GO\(\)](#), [Ensembl2GO\(\)](#), [EntrezGene2GO\(\)](#), [Uniprot2GO\(\)](#), [available\\_organisms\(\)](#), [genomic\\_ressource-class](#), [taxonomy\(\)](#)

Other GO\_terms: [GOcount\(\)](#), [GOterms\\_heatmap\(\)](#), [create\\_topGOdata\(\)](#), [gene2GO-class](#), [merge\\_enrich\\_terms\(\)](#)

### Examples

```
## Not run:
## load Mus musculus (mouse) GO annotations

# from Bioconductor
Bioconductor<-ViSEAGO::Bioconductor2GO()
myGENE2GO<-ViSEAGO::annotate(
  id="org.Mm.eg.db",
  object=Bioconductor
)

# from EntrezGene
EntrezGene<-ViSEAGO::EntrezGene2GO()
myGENE2GO<-ViSEAGO::annotate(
  id="10090",
  object=EntrezGene
)

# from EntrezGene
Ensembl<-ViSEAGO::Ensembl2GO()
myGENE2GO<-ViSEAGO::annotate(
  id="mmusculus_gene_ensembl",
  object=Ensembl
)

# from Uniprot
Uniprot<-ViSEAGO::Uniprot2GO()
myGENE2GO<-ViSEAGO::annotate(
  id="mouse",
  object=Uniprot
)

## from Custom GO annotation file
Custom<-ViSEAGO::Custom2GO(system.file("extdata/customfile.txt",package = "ViSEAGO"))
myGENE2GO<-ViSEAGO::annotate(
  id="myspecies1",
  object=Custom
)
```

```

## specific options for EntrezGene database

# Chicken GO annotations without adding orthologs
EntrezGene<-ViSEAGO::EntrezGene2GO()
myGENE2GO<-ViSEAGO::annotate(
  id="9031",
  object=EntrezGene
)

# Chicken GO annotation with the add of orthologs GO annotations
EntrezGene<-ViSEAGO::EntrezGene2GO()
myGENE2GO<-ViSEAGO::annotate(
  id="9031",
  object=EntrezGene,
  ortholog=TRUE
)

# display organisms supported by NCBI EntrezGene orthologs pipeline
EntrezGene<-ViSEAGO::EntrezGene2GO()
ViSEAGO::annotate(
  id="NULL",
  object=EntrezGene,
  ortholog=TRUE
)

## End(Not run)

```

---

available\_organisms    *Display available organisms from a specified database.*

---

## Description

Display an interactive table with available organisms from a genomic resource database (Bioconductor, EntrezGene, Ensembl, Uniprot).

## Usage

```

available_organisms(object)

## S4 method for signature 'genomic_resource'
available_organisms(object)

```

## Arguments

object            a [genomic\\_resource-class](#) object created by [Bioconductor2GO](#), [EntrezGene2GO](#), [Ensembl2GO](#), or [Uniprot2GO](#) methods.

## Details

an interactive datatable.

## Value

javascript datatable

## References

Yihui Xie (2016). DT: A Wrapper of the JavaScript Library 'DataTables'. R package version 0.2. <https://CRAN.R-project.org/package=DT>

## See Also

Other genomic\_ressource: [Bioconductor2GO\(\)](#), [Custom2GO\(\)](#), [Ensembl2GO\(\)](#), [EntrezGene2GO\(\)](#), [Uniprot2GO\(\)](#), [annotate\(\)](#), [genomic\\_ressource-class](#), [taxonomy\(\)](#)

Other visualization: [GOclusters\\_heatmap\(\)](#), [GOcount\(\)](#), [GOterms\\_heatmap\(\)](#), [Upset\(\)](#), [overLapper\(\)](#), [show\\_heatmap\(\)](#), [show\\_table\(\)](#)

## Examples

```
# display Bioconductor table
Bioconductor<-ViSEAGO::Bioconductor2GO()
ViSEAGO::available_organisms(Bioconductor)
## Not run:

# display EntrezGene table
EntrezGene<-ViSEAGO::EntrezGene2GO()
ViSEAGO::available_organisms(EntrezGene)

# display Ensembl table
Ensembl<-ViSEAGO::Ensembl2GO()
ViSEAGO::available_organisms(Ensembl)

# display Uniprot table
Uniprot<-ViSEAGO::Uniprot2GO()
ViSEAGO::available_organisms(Uniprot)

## End(Not run)
```

---

Bioconductor2GO

*Check available organisms databases at Bioconductor.*

---

## Description

Retrieve the Bioconductor **OrgDb** available organisms databases packages.

## Usage

```
Bioconductor2GO()
```

## Details

This function gives genome wide annotation for available organisms databases packages from **Bioconductor OrgDb**. It uses `loadAnnDbPkgIndex` from **AnnotationForge** package.

## Value

a [genomic\\_ressource-class](#) object required by [annotate](#) method.

## References

Carlson M and Pages H (2017). AnnotationForge: Code for Building Annotation Database Packages. R package version 1.18.0.

## See Also

Other genomic\_ressource: [Custom2GO\(\)](#), [Ensembl2GO\(\)](#), [EntrezGene2GO\(\)](#), [Uniprot2GO\(\)](#), [annotate\(\)](#), [available\\_organisms\(\)](#), [genomic\\_ressource-class](#), [taxonomy\(\)](#)

## Examples

```
# Check Bioconductor OrgDb available organisms
Bioconductor<-ViSEAG0::Bioconductor2GO()
```

---

build_GO_SS	<i>build GO Semantic Similarity object.</i>
-------------	---

---

## Description

Compute the Information content (IC) on the given ontology, and create a [GO\\_SS-class](#) object required by [compute\\_SS\\_distances](#) method to compute GO semantic similarity between enriched GO terms or groups of terms.

## Usage

```
build_GO_SS(gene2GO, enrich_GO_terms)

## S4 method for signature 'gene2GO,enrich_GO_terms'
build_GO_SS(gene2GO, enrich_GO_terms)
```

## Arguments

`gene2GO` a [gene2GO-class](#) object from [annotate](#) method.  
`enrich_GO_terms` a [enrich\\_GO\\_terms-class](#) from [merge\\_enrich\\_terms](#) method.

## Details

This method use [annotate](#) and [merge\\_enrich\\_terms](#) output objects (see Arguments), and compute the Information content (IC) using the internal code of [godata](#) function from **GOSemSim** package.

## Value

a [GO\\_SS-class](#) object required by [compute\\_SS\\_distances](#).

## References

Alexa A, Rahnenfuhrer J, Lengauer T. Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics* 2006; 22:1600-1607.

Guangchuan Yu, Fei Li, Yide Qin, Xiaochen Bo, Yibo Wu and Shengqi Wang. GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. *Bioinformatics* 2010 26(7):976-978.

Herve Pages, Marc Carlson, Seth Falcon and Nianhua Li (2017). AnnotationDbi: Annotation Database Interface. R package version 1.38.0.

## See Also

Other GO\_semantic\_similarity: [GO\\_SS-class](#), [compute\\_SS\\_distances\(\)](#)

## Examples

```
## Not run:
# initialise object for compute GO Semantic Similarity
myGOs<-ViSEAGO::build_GO_SS(
  myGENE2GO,
  BP_sResults
)

## End(Not run)
# load data example
utils::data(
  myGOs,
  package="ViSEAGO"
)
```

---

clusters\_cor

*Compute distance matrix between dendrograms partitions.*

---

## Description

Build a distance or correlation matrix between partitions from dendrograms.

## Usage

```
clusters_cor(clusters, method = "adjusted.rand")

## S4 method for signature 'list,character'
clusters_cor(clusters, method = "adjusted.rand")
```

## Arguments

clusters	a list of <a href="#">GO_clusters-class</a> objects, from <a href="#">GOterms_heatmap</a> or <a href="#">GOclusters_heatmap</a> , named as character.
method	available methods ("vi", "nmi", "split.join", "rand", or "adjusted.rand") from <b>igraph</b> package <a href="#">compare</a> function.



**Value**

a distance or a correlation matrix.

**References**

Csardi G, Nepusz T: The igraph software package for complex network research, InterJournal, Complex Systems 1695. 2006. <http://igraph.org>.

**See Also**

Other GO\_clusters: [GO\\_clusters-class](#), [G0clusters\\_heatmap\(\)](#), [compare\\_clusters\(\)](#), [show\\_heatmap\(\)](#), [show\\_table\(\)](#)

**Examples**

```
# load example object
data(
  myGOs,
  package="ViSEAGO"
)

## Not run:
# compute Semantic Similarity (SS)
myGOs<-ViSEAGO::compute_SS_distances(
  myGOs,
  distance=c("Resnik", "Lin", "Rel", "Jiang", "Wang")
)

# Resnik distance GO terms heatmap
Resnik_clusters_wardD2<-ViSEAGO::G0terms_heatmap(
  myGOs,
  showIC=TRUE,
  showGOlabels=TRUE,
  GO.tree=list(
    tree=list(
      distance="Resnik",
      aggreg.method="ward.D2"
    ),
    cut=list(
      dynamic=list(
        deepSplit=2,
        minClusterSize =2
      )
    )
  ),
  samples.tree=NULL
)

# Lin distance GO terms heatmap
Lin_clusters_wardD2<-ViSEAGO::G0terms_heatmap(
  myGOs,
  showIC=TRUE,
  showGOlabels=TRUE,
  GO.tree=list(
    tree=list(
      distance="Lin",
```

```
        aggreg.method="ward.D2"
    ),
    cut=list(
        dynamic=list(
            deepSplit=2,
            minClusterSize =2
        )
    )
),
samples.tree=NULL
)

# Resnik distance GO terms heatmap
Rel_clusters_wardD2<-ViSEAGO::GOterms_heatmap(
    myGOs,
    showIC=TRUE,
    showGOlabels=TRUE,
    GO.tree=list(
        tree=list(
            distance="Rel",
            aggreg.method="ward.D2"
        ),
        cut=list(
            dynamic=list(
                deepSplit=2,
                minClusterSize =2
            )
        )
    ),
    samples.tree=NULL
)

# Resnik distance GO terms heatmap
Jiang_clusters_wardD2<-ViSEAGO::GOterms_heatmap(
    myGOs,
    showIC=TRUE,
    showGOlabels=TRUE,
    GO.tree=list(
        tree=list(
            distance="Jiang",
            aggreg.method="ward.D2"
        ),
        cut=list(
            dynamic=list(
                deepSplit=2,
                minClusterSize =2
            )
        )
    ),
    samples.tree=NULL
)

# Resnik distance GO terms heatmap
Wang_clusters_wardD2<-ViSEAGO::GOterms_heatmap(
    myGOs,
    showIC=TRUE,
    showGOlabels=TRUE,
```

```

GO.tree=list(
  tree=list(
    distance="Wang",
    aggreg.method="ward.D2"
  ),
  cut=list(
    dynamic=list(
      deepSplit=2,
      minClusterSize =2
    )
  )
),
samples.tree=NULL
)

## End(Not run)
# clusters to compare
clusters<-list(
  Resnik="Resnik_clusters_wardD2",
  Lin="Lin_clusters_wardD2",
  Rel="Rel_clusters_wardD2",
  Jiang="Jiang_clusters_wardD2",
  Wang="Wang_clusters_wardD2"
)

## Not run:
# global dendrogram clustering correlation
clust_cor<-ViSEAGO::clusters_cor(
  clusters,
  method="adjusted.rand"
)

## End(Not run)

```

---

compare\_clusters      *Heatmap to compare partitions*

---

## Description

Build an interactive heatmap of the common GO terms frequency between several partitions.

## Usage

```

compare_clusters(clusters)

## S4 method for signature 'list'
compare_clusters(clusters)

```

## Arguments

clusters      a list of named `GO_clusters-class` objects, from `GOterms_heatmap` or `GOclusters_heatmap` methods.

**Details**

Build an interactive heatmap of common GO terms frequency between partitions from several `GO_clusters-class` objects.

**Value**

an interactive javascript heatmap.

**References**

Carson Sievert, Chris Parmer, Toby Hocking, Scott Chamberlain, Karthik Ram, Marianne Corvellec and Pedro Despouy (2017). `plotly`: Create Interactive Web Graphics via 'plotly.js'. R package version 4.6.0. <https://CRAN.R-project.org/package=plotly>

**See Also**

Other `GO_clusters`: `GO_clusters-class`, `G0clusters_heatmap()`, `clusters_cor()`, `show_heatmap()`, `show_table()`

**Examples**

```
# load example object
data(
  myGOs,
  package="ViSEAGO"
)

## Not run:
# compute Semantic Similarity (SS)
myGOs<-ViSEAGO::compute_SS_distances(
  myGOs,
  distance=c("Resnik", "Lin", "Rel", "Jiang", "Wang")
)

# Resnik distance GO terms heatmap
Resnik_clusters_wardD2<-ViSEAGO::G0terms_heatmap(
  myGOs,
  showIC=TRUE,
  showGOlabels=TRUE,
  GO.tree=list(
    tree=list(
      distance="Resnik",
      aggreg.method="ward.D2"
    ),
    cut=list(
      dynamic=list(
        deepSplit=2,
        minClusterSize =2
      )
    )
  ),
  samples.tree=NULL
)

# Lin distance GO terms heatmap
Lin_clusters_wardD2<-ViSEAGO::G0terms_heatmap(
```

```
myGOs,
showIC=TRUE,
showGOlabels=TRUE,
GO.tree=list(
  tree=list(
    distance="Lin",
    aggreg.method="ward.D2"
  ),
  cut=list(
    dynamic=list(
      deepSplit=2,
      minClusterSize =2
    )
  )
),
samples.tree=NULL
)

# Resnik distance GO terms heatmap
Rel_clusters_wardD2<-ViSEAGO::GOterms_heatmap(
  myGOs,
  showIC=TRUE,
  showGOlabels=TRUE,
  GO.tree=list(
    tree=list(
      distance="Rel",
      aggreg.method="ward.D2"
    ),
    cut=list(
      dynamic=list(
        deepSplit=2,
        minClusterSize =2
      )
    )
  ),
  samples.tree=NULL
)

# Resnik distance GO terms heatmap
Jiang_clusters_wardD2<-ViSEAGO::GOterms_heatmap(
  myGOs,
  showIC=TRUE,
  showGOlabels=TRUE,
  GO.tree=list(
    tree=list(
      distance="Jiang",
      aggreg.method="ward.D2"
    ),
    cut=list(
      dynamic=list(
        deepSplit=2,
        minClusterSize =2
      )
    )
  ),
  samples.tree=NULL
)
```

```

# Resnik distance GO terms heatmap
Wang_clusters_wardD2<-ViSEAGO::GOterms_heatmap(
  myGOs,
  showIC=TRUE,
  showGOlabels=TRUE,
  GO.tree=list(
    tree=list(
      distance="Wang",
      aggreg.method="ward.D2"
    ),
    cut=list(
      dynamic=list(
        deepSplit=2,
        minClusterSize =2
      )
    )
  ),
  samples.tree=NULL
)

## End(Not run)

# clusters to compare
clusters<-list(
  Resnik="Resnik_clusters_wardD2",
  Lin="Lin_clusters_wardD2",
  Rel="Rel_clusters_wardD2",
  Jiang="Jiang_clusters_wardD2",
  Wang="Wang_clusters_wardD2"
)

## Not run:
# clusters content comparisons
clusters_comp<-ViSEAGO::compare_clusters(clusters)

## End(Not run)

```

---

compute\_SS\_distances    *Compute distance between GO terms or GO clusters based on semantic similarity.*

---

### Description

This method computes distance between GO terms or GO clusters based on semantic similarity.

### Usage

```

compute_SS_distances(object, distance)

## S4 method for signature 'ANY,character'
compute_SS_distances(object, distance)

```

**Arguments**

object	a <a href="#">GO_SS-class</a> , or <a href="#">GO_clusters-class</a> objects created by <a href="#">build_GO_SS</a> or <a href="#">G0terms_heatmap</a> methods, respectively.
distance	The available methods for calculating GO terms Semantic Similarity (SS) are "Resnik", "Rel", "Lin", and "Jiang" which are based on Information Content (IC), and "Wang" which is based on graph topology. The available methods for calculating clusters of GO terms SS are "max", "avg", "rmax", and "BMA".

**Details**

This method computes semantic similarity distances between all GO terms provided by [GO\\_SS-class](#) object.

This method also computes semantic similarity distances between all GO clusters provided by [GO\\_clusters-class](#) object.

Semantic Similarity computations are based on [mgoSim](#) method from the **GoSemSim** package.

**Value**

a [GO\\_SS-class](#), or a [GO\\_clusters-class](#) object (same class as input object).

**References**

Marc Carlson (2017). *GO.db: A set of annotation maps describing the entire Gene Ontology*. R package version 3.4.1.

Guangchuan Yu, Fei Li, Yide Qin, Xiaochen Bo, Yibo Wu and Shengqi Wang. *GOSemSim: an R package for measuring semantic similarity among GO terms and gene products*. *Bioinformatics* 2010 26(7):976-978

Herve Pages, Marc Carlson, Seth Falcon and Nianhua Li (2017). *AnnotationDbi: Annotation Database Interface*. R package version 1.38.0.

**See Also**

Other `GO_semantic_similarity`: [GO\\_SS-class](#), [build\\_GO\\_SS\(\)](#)

**Examples**

```
# load data example
data(
  myGOs,
  package="ViSEAGO"
)

## Not run:
# compute GO terms Semantic Similarity distances
myGOs<-ViSEAGO::compute_SS_distances(
  myGOs,
  distance=c("Resnik","Lin","Rel","Jiang","Wang")
)

# G0termsHeatmap with default parameters
Wang_clusters_wardD2<-ViSEAGO::G0terms_heatmap(
```

```

myGOs,
showIC=TRUE,
showGOlabels=TRUE,
GO.tree=list(
  tree=list(
    distance="Wang",
    aggreg.method="ward.D2",
    rotate=NULL
  ),
  cut=list(
    dynamic=list(
      pamStage=TRUE,
      pamRespectsDendro=TRUE,
      deepSplit=2,
      minClusterSize=2
    )
  )
),
samples.tree=NULL
)

# compute clusters of GO terms Semantic Similarity distances
Wang_clusters_wardD2<-ViSEAGO::compute_SS_distances(
  Wang_clusters_wardD2,
  distance=c("max","avg","rcmax","BMA")
)

## End(Not run)

```

---

create\_topGOdata

*Create topGOdata object for enrichment test with topGO package.*


---

## Description

This method create a [topGOdata-class](#) object required by **topGO** package in order to perform GO enrichment test.

## Usage

```
create_topGOdata(geneSel, allGenes, geneList = NULL, gene2GO, ont, nodeSize)
```

```
## S4 method for signature 'ANY,ANY,ANY,gene2GO,character,numeric'
```

```
create_topGOdata(geneSel, allGenes, geneList = NULL, gene2GO, ont, nodeSize)
```

## Arguments

geneSel	genes of interest.
allGenes	customized background genes.
geneList	logical factor (1: genes of interest, 0: genes background, and gene identifiants in names) (default value to NULL).
gene2GO	a <a href="#">gene2GO-class</a> object created by <a href="#">annotate</a> method.
ont	the ontology used is "MF" (Molecular Function), "BP" (Biological Process), or "CC" (Cellular Component).
nodeSize	the minimum number of genes for each GO term.



## Details

This method is a convenient wrapper building a [topGOdata-class](#) object using a given ontology category (ont argument) in order to perform GO enrichment test. The complete GO annotation is required (gene2GO argument) and also the list of genes of interest (geneSel argument) against the corresponding background (allGenes argument) separately, or grouped together in a factor (geneList argument).

## Value

a [topGOdata-class](#) object required by runTest from **topGO** package.

## References

Alexa A, Rahnenfuhrer J, Lengauer T. Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics* 2006; 22:1600-1607.

## See Also

Other GO\_terms: [GOcount\(\)](#), [GOterms\\_heatmap\(\)](#), [annotate\(\)](#), [gene2GO-class](#), [merge\\_enrich\\_terms\(\)](#)

## Examples

```
# load genes identifiers (GeneID,ENS...) background (Expressed genes)
background<-scan(
  system.file(
    "extdata/data/input",
    "background_L.txt",
    package = "ViSEAGO"
  ),
  quiet=TRUE,
  what=""
)

# load Differentially Expressed (DE) gene identifiers from files
pregnantvslactateDE<-scan(
  system.file(
    "extdata/data/input",
    "pregnantvslactateDE.txt",
    package = "ViSEAGO"
  ),
  quiet=TRUE,
  what=""
)

## Not run:
# create topGOdata for BP for each list of DE genes
BP_L_pregnantvslactate<-ViSEAGO::create_topGOdata(
  geneSel=pregnantvslactateDE,
  allGenes=background,
  gene2GO=myGENE2GO,
  ont="BP",
  nodeSize=5
)

## End(Not run)
```

---

Custom2GO

*Store organisms GO annotations from custom database file.*

---

### Description

Store the available species and current GO annotations from a custom table file

### Usage

```
Custom2GO(file)
```

### Arguments

file                    custom GO annotation file

### Details

This function load a custom GO annotation database table that must contain columns:

**taxid** custom taxonomic identifiants

**gene\_id** custom gene identifiants

**gene\_symbol** custom gene symbols

**GOID** Known GO identifiants (see `select(GO.db, columns=columns(GO.db), keys=keys(GO.db))`)

**evidence** Known GO **evidence codes**

### Value

a [genomic\\_ressource-class](#) object required by [annotate](#).

### References

Matt Dowle and Arun Srinivasan (2017). `data.table`: Extension of ‘`data.frame`’. R package version 1.10.4. <https://CRAN.R-project.org/package=data.table>.

### See Also

Other `genomic_ressource`: [Bioconductor2GO\(\)](#), [Ensembl2GO\(\)](#), [EntrezGene2GO\(\)](#), [Uniprot2GO\(\)](#), [annotate\(\)](#), [available\\_organisms\(\)](#), [genomic\\_ressource-class](#), [taxonomy\(\)](#)

### Examples

```
# Download custom GO annotations
Custom<-ViSEAGO::Custom2GO(
  system.file(
    "extdata/customfile.txt",
    package = "ViSEAGO"
  )
)
```

---

enrich\_GO\_terms-class *enrich\_GO\_terms class object definition.*

---

### Description

This class is invoked by [merge\\_enrich\\_terms](#) method in order to store the merged data.table and associated metadata.

### Slots

same\_genes\_background logical.

input a list containing named elements. Each element must contain the name of [topGOdata-class](#) object created by [create\\_topGOdata](#) method and the associated [topGOresult-class](#) object(s) to combine (see examples in [merge\\_enrich\\_terms](#)).

ont ontology used "MF", "BP", or "CC".

topGO a list with topGO objects summary informations.

data a merged data.table of enriched GO terms (p<0.01) in at least once with GO descriptions and statistical values.

### See Also

Other enrich\_GO\_terms: [Upset\(\)](#), [overLapper\(\)](#), [show\\_heatmap\(\)](#), [show\\_table\(\)](#)

---

Ensembl2GO

*Check available organisms datasets at Ensembl.*

---

### Description

List Ensembl referenced organisms datasets from the current (NULL) or archive (number in character) annotation version.

### Usage

```
Ensembl2GO(biomart = "ensembl", host = "www.ensembl.org", version = NULL)
```

### Arguments

biomart	the biomart name (eg. "ensembl", the default) available with <b>biomaRt</b> package <a href="#">listEnsembl</a> .
host	the Ensembl host adress for <b>vertebrate</b> ("www.ensembl.org", the default value), <b>plants</b> ("plants.ensembl.org"), <b>metazoa</b> ("metazoa.ensembl.org"), or <b>fungi</b> ("fungi.ensembl.org").
version	the annotation version to use (eg. NULL for the default current version, or a version number in character)

### Details

This function gives referenced organisms genomes at [Ensembl](#). It uses the [useEnsembl](#) and [listDatasets](#) from **biomaRt** package.

**Value**

a `genomic_ressource-class` object required by `annotate`.

**References**

Durinck S, Spellman P, Birney E and Huber W (2009). Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package `biomaRt`. *Nature Protocols*, 4, pp. 1184-1191.

Durinck S, Moreau Y, Kasprzyk A, Davis S, De Moor B, Brazma A and Huber W (2005). BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinformatics*, 21, pp. 3439-3440.

Matt Dowle and Arun Srinivasan (2017). `data.table`: Extension of `data.frame`. R package version 1.10.4. <https://CRAN.R-project.org/package=data.table>.

**See Also**

Other `genomic_ressource`: `Bioconductor2GO()`, `Custom2GO()`, `EntrezGene2GO()`, `Uniprot2GO()`, `annotate()`, `available_organisms()`, `genomic_ressource-class`, `taxonomy()`

**Examples**

```
# host address to use for Ensembl vertebrate/Plants/Protists/Metazoa/Fungi
# host="www.ensembl.org" # vertebrate
# host="plants.ensembl.org"
# host="protists.ensembl.org"
# host="metazoa.ensembl.org"
# host="fungi.ensembl.org"
# biomart is not available for bacteria

## Not run:
# check the Ensembl available biomart (if not known)
# for Animals (host="www.ensembl.org", the default)
biomaRt::listEnsembl()

# List Ensembl available organisms
Ensembl<-ViSEAGO::Ensembl2GO(
  biomart="ensembl",
  host="www.ensembl.org",
  version=NULL
)

## End(Not run)
```

---

EntrezGene2GO

*Store available organisms GO annotations at EntrezGene.*


---

**Description**

Store the available species and current GO annotations from the [gene2go.gz](http://gene2go.gz) nfile available at NCBI [EntrezGene ftp](http://EntrezGene ftp).

**Usage**

```
EntrezGene2GO()
```

**Details**

This function downloads the [gene2go.gz](#) file from [EntrezGene ftp](#) which contains available organisms (taxid) with the corresponding GO annotations.

**Value**

a [genomic\\_ressource-class](#) object required by [annotate](#).

**References**

Matt Dowle and Arun Srinivasan (2017). [data.table](#): Extension of ‘data.frame’. R package version 1.10.4. <https://CRAN.R-project.org/package=data.table>.

Eric Sayers (2013). [Entrez Programming Utilities Help](#).

#’ Henrik Bengtsson (2016). [R.utils](#): Various Programming Utilities. R package version 2.5.0. <https://CRAN.R-project.org/package=R.utils>.

Maglott, D, Ostell, J, Pruitt, KD, Tatusova, T (2011). [Entrez Gene: gene-centered information at NCBI](#). *Nucleic Acids Res.*, 39, Database issue:D52-7.

**See Also**

Other [genomic\\_ressource](#): [Bioconductor2GO\(\)](#), [Custom2GO\(\)](#), [Ensembl2GO\(\)](#), [Uniprot2GO\(\)](#), [annotate\(\)](#), [available\\_organisms\(\)](#), [genomic\\_ressource-class](#), [taxonomy\(\)](#)

**Examples**

```
## Not run:
# Download EntrezGene available organisms GO annotations
EntrezGene<-ViSEAGO::EntrezGene2GO()

## End(Not run)
```

---

<code>gene2GO-class</code>	<i>gene2GO class object definition.</i>
----------------------------	---

---

**Description**

This class is invoked by [annotate](#) method in order to store GO annotations for each category (MF, BP, CC).

**Slots**

db database source in character.  
stamp date of stamp in character.  
organism target species GO annotation in character.  
MF a list containing GO terms for Molecular Function (MF) category for each gene element.  
BP a list containing GO terms for Biological Process (BP) category for each gene element.  
CC a list containing GO terms for Cellular Component (CC) category for each gene element.

**See Also**

Other [GO\\_terms](#): [GOcount\(\)](#), [GOterms\\_heatmap\(\)](#), [annotate\(\)](#), [create\\_topGOdata\(\)](#), [merge\\_enrich\\_terms\(\)](#)

---

genomic\_ressource-class

*genomic\_ressource class object definition.*

---

### Description

This class stores the annotations and associated metadata obtained by [Bioconductor2GO](#), [EntrezGene2GO](#), [Ensembl2GO](#), or [Uniprot2GO](#).

### Slots

db name of database used (Bioconductor, EntrezGene, Ensembl, or Uniprot).

stamp date of stamp (for Bioconductor, EntrezGene, and Uniprot), or annotation version for Ensembl database.

data GO annotations from [EntrezGene2GO](#) method.

organisms informations about species/datasets availables.

mart Ensembl mart from [Ensembl2GO](#) method.

### See Also

Other genomic\_ressource: [Bioconductor2GO\(\)](#), [Custom2GO\(\)](#), [Ensembl2GO\(\)](#), [EntrezGene2GO\(\)](#), [Uniprot2GO\(\)](#), [annotate\(\)](#), [available\\_organisms\(\)](#), [taxonomy\(\)](#)

---

GOclusters\_heatmap *Build a clustering heatmap on GO groups.*

---

### Description

This method computes a clustering heatmap based on GO groups semantic similarity.

### Usage

```
GOclusters_heatmap(
  object,
  tree = list(distance = "BMA", aggreg.method = "ward.D2", rotate = NULL)
)

## S4 method for signature 'GO_clusters,list'
GOclusters_heatmap(
  object,
  tree = list(distance = "BMA", aggreg.method = "ward.D2", rotate = NULL)
)
```

**Arguments**

- object** a `GO_clusters-class` object from `compute_SS_distances`.
- tree** a named list with:
- distance ("BMA" by default)** distance computed from the semantic similarity for GO groups which could be "max", "avg", "rcmax", or "BMA".
  - aggreg.method ("ward.D2" by default)** aggregation method criteria from `hclust` ("ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median", or "centroid") to build a dendrogram.
  - rotate** sort the branches of the tree based on a vector - either of labels order or the labels in their new order

**Details**

This method computes a clustering heatmap based on GO groups semantic similarity (computed with `compute_SS_distances`).

The heatmap color intensity corresponds to the number of GO terms in each GO group.

GO group description is defined as the first common GO ancestor with the cluster identifier in brackets.

The dendrogram branches are colored according to GO terms clusters.

**Value**

a `GO_clusters-class` object.

**References**

Matt Dowle and Arun Srinivasan (2017). `data.table`: Extension of 'data.frame'. R package version 1.10.4. <https://CRAN.R-project.org/package=data.table>.

Tal Galili (2015). `dendextend`: an R package for visualizing, adjusting, and comparing trees of hierarchical clustering. *Bioinformatics*. DOI:10.1093/bioinformatics/btv428.

Tal Galili (2017). `heatmaply`: Interactive Cluster Heat Maps Using 'plotly'. R package version 0.9.1. <https://CRAN.R-project.org/package=heatmaply>.

Erich Neuwirth (2014). `RColorBrewer`: ColorBrewer Palettes. R package version 1.1-2. <https://CRAN.R-project.org/package=RColorBrewer>.

Carson Sievert, Chris Parmer, Toby Hocking, Scott Chamberlain, Karthik Ram, Marianne Corvellec and Pedro Despouy (2017). `plotly`: Create Interactive Web Graphics via 'plotly.js'. R package version 4.6.0. <https://CRAN.R-project.org/package=plotly>.

H. Wickham. `ggplot2`: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2009.

**See Also**

Other `GO_clusters`: `GO_clusters-class`, `clusters_cor()`, `compare_clusters()`, `show_heatmap()`, `show_table()`

Other semantic\_similarity: `GOterms_heatmap()`

Other visualization: `GOcount()`, `GOterms_heatmap()`, `Upset()`, `available_organisms()`, `overLapper()`, `show_heatmap()`, `show_table()`

**Examples**

```

# load data example
utils::data(
  myGOs,
  package="ViSEAGO"
)
## Not run:
# compute GO terms Semantic Similarity distances
myGOs<-ViSEAGO::compute_SS_distances(
  myGOs,
  distance="Wang"
)

# GOtermsHeatmap with default parameters
Wang_clusters_wardD2<-ViSEAGO::GOterms_heatmap(
  myGOs,
  showIC=TRUE,
  showGOlabels=TRUE,
  GO.tree=list(
    tree=list(
      distance="Wang",
      aggreg.method="ward.D2",
      rotate=NULL
    ),
    cut=list(
      dynamic=list(
        pamStage=TRUE,
        pamRespectsDendro=TRUE,
        deepSplit=2,
        minClusterSize =2
      )
    )
  ),
  samples.tree=NULL
)

# compute clusters of GO terms Semantic Similarity distances
Wang_clusters_wardD2<-ViSEAGO::compute_SS_distances(
  Wang_clusters_wardD2,
  distance="BMA"
)

# GOclusters heatmap
Wang_clusters_wardD2<-ViSEAGO::GOclusters_heatmap(
  Wang_clusters_wardD2,
  tree=list(
    distance="BMA",
    aggreg.method="ward.D2",
    rotate=NULL
  )
)

## End(Not run)

```



---

GOcount

*Barplot for the count of GO terms.*

---

### Description

This method displays in barplot the count of GO terms splitted in two categories (significant or not) for each result of GO enrichment tests.

### Usage

```
GOcount(object, file = NULL)
```

```
## S4 method for signature 'ANY'  
GOcount(object, file = NULL)
```

### Arguments

**object** an `enrich_GO_terms-class` object from `merge_enrich_terms` method.  
**file** the name of the output file (default to NULL for interactive screen display).

### Details

This method displays an interactive barplot, using **plotly** package, from a `merge_enrich_terms` output object.  
A static image (in png) could be printed by setting file argument.

### Value

a barplot.

### References

Carson Sievert, Chris Parmer, Toby Hocking, Scott Chamberlain, Karthik Ram, Marianne Corvellec and Pedro Despouy (2017). `plotly`: Create InteractiveWeb Graphics via 'plotly.js'. R package version 4.6.0. <https://CRAN.R-project.org/package=plotly>.

### See Also

Other GO\_terms: `GOterms_heatmap()`, `annotate()`, `create_topGOdata()`, `gene2GO-class`, `merge_enrich_terms()`  
Other visualization: `GOclusters_heatmap()`, `GOterms_heatmap()`, `Upset()`, `available_organisms()`, `overLapper()`, `show_heatmap()`, `show_table()`

### Examples

```
# load object  
utils::data(  
  myGOs,  
  package="ViSEAGO"  
)  
  
# barplot for the count of GO terms  
ViSEAGO::GOcount( myGOs)
```

---

GOterms\_heatmap      *Build a clustering heatmap on GO terms.*

---

### Description

This method computes a clustering heatmap based on GO terms semantic similarity.

### Usage

```
GOterms_heatmap(
  myGOs,
  showIC = TRUE,
  showGOlabels = TRUE,
  heatmap_colors = c("#ffffff", "#99000D"),
  GO.tree = list(tree = list(distance = "Wang", aggreg.method = "ward.D2", rotate =
    NULL), cut = list(dynamic = list(pamStage = TRUE, pamRespectsDendro = TRUE, deepSplit
    = 2, minClusterSize = 2))),
  samples.tree = NULL
)

## S4 method for signature 'GO_SS'
GOterms_heatmap(
  myGOs,
  showIC = TRUE,
  showGOlabels = TRUE,
  heatmap_colors = c("#ffffff", "#99000D"),
  GO.tree = list(tree = list(distance = "Wang", aggreg.method = "ward.D2", rotate =
    NULL), cut = list(dynamic = list(pamStage = TRUE, pamRespectsDendro = TRUE, deepSplit
    = 2, minClusterSize = 2))),
  samples.tree = NULL
)
```

### Arguments

myGOs	a <a href="#">GO_SS-class</a> object from <a href="#">compute_SS_distances</a> .
showIC	logical (default to TRUE) to display the GO terms Information Content (IC) side bar.
showGOlabels	logical (default to TRUE) to display the GO terms ticks on y axis.
heatmap_colors	pvalues color range with white to Sangria collors by default (c("#ffffff", "#99000D")).
GO.tree	a named list of parameters to build and cut the GO terms dendrogram. <b>tree (a named list with:)</b> <b>distance ("Wang" by default)</b> distance computed from the semantic similarity which could be IC-based ("Resnik", "Rel", "Lin", or "Jiang") or graph-based ("Wang"). <b>aggreg.method ("ward.D2" by default)</b> aggregation method criteria from <a href="#">hclust</a> ("ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median", or "centroid") to build a dendrogram. <b>rotate</b> sort the branches of the tree based on a vector - either of labels order or the labels in their new order

**cut (a named list with:)** **static (default to NULL)** a numeric value that is the height (between 0 and 1), or the number of clusters (value > 1) to cut the dendrogram.

**dynamic (a named list which only contains cutreeDynamic options values below)** **pamS** second (PAM-like) stage will be performed.

**pamRespectsDendro (default to TRUE)** PAM stage will respect the dendrogram in the sense that objects and small clusters will only be assigned to clusters that belong to the same branch that the objects or small clusters being assigned belong to.

**deepSplit (default to 2)** provides a rough control over sensitivity for cluster splitting (range 0 to 4). The higher the value (or if TRUE), the more and smaller clusters will be produced.

**minClusterSize (default to 2)** minimum cluster size.

**samples.tree** a named list of parameters to build and cut the samples dendrogram (default to NULL).

**tree (a named list with:)** **distance ("pearson" by default)** distance computed that could be correlation ("abs.pearson", "pearson", "kendall", or "spearman"), or dist method (euclidean, "maximum", "manhattan", "canberra", "binary", or "minkowski).

**aggreg.method ("average" by default)** same options than for GO.tree argument

**cut** same options than for GO.tree argument.

## Details

This method computes a clustering heatmap based on GO terms semantic similarity (computed with [compute\\_SS\\_distances](#)).

The dendrogram produced could be cutted in static or dynamic mode.

1. build dendrograms on GO terms and optionally on samples.
2. cut in static or dynamic mode and color the dendrogram branches.
3. build an interactive clustering heatmap based on [heatmaply](#).

## Value

a [GO\\_clusters-class](#) object.

## References

Matt Dowle and Arun Srinivasan (2017). data.table: Extension of 'data.frame'. R package version 1.10.4. <https://CRAN.R-project.org/package=data.table>.

Tal Galili (2015). dendextend: an R package for visualizing, adjusting, and comparing trees of hierarchical clustering. Bioinformatics. DOI:10.1093/bioinformatics/btv428.

Tal Galili (2017). heatmaply: Interactive Cluster Heat Maps Using 'plotly'. R package version 0.9.1. <https://CRAN.R-project.org/package=heatmaply>.

Peter Langfelder, Bin Zhang and with contributions from Steve Horvath (2016). dynamicTreeCut: Methods for Detection of Clusters in Hierarchical Clustering Dendrograms. R package version 1.63-1. <https://CRAN.R-project.org/package=dynamicTreeCut>.

Erich Neuwirth (2014). RColorBrewer: ColorBrewer Palettes. R package version 1.1-2. <https://CRAN.R-project.org/package=RColorBrewer>.

Carson Sievert, Chris Parmer, Toby Hocking, Scott Chamberlain, Karthik Ram, Marianne Corvellec and Pedro Despouy (2017). plotly: Create Interactive Web Graphics via 'plotly.js'. R package version 4.6.0. <https://CRAN.R-project.org/package=plotly>.

Hadley Wickham (2016). scales: Scale Functions for Visualization. R package version 0.4.1. <https://CRAN.R-project.org/package=scales>.

H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2009.

### See Also

Other GO\_terms: [GOcount\(\)](#), [annotate\(\)](#), [create\\_topGOdata\(\)](#), [gene2GO-class](#), [merge\\_enrich\\_terms\(\)](#)

Other semantic\_similarity: [GOclusters\\_heatmap\(\)](#)

Other visualization: [GOclusters\\_heatmap\(\)](#), [GOcount\(\)](#), [Upset\(\)](#), [available\\_organisms\(\)](#), [overLapper\(\)](#), [show\\_heatmap\(\)](#), [show\\_table\(\)](#)

### Examples

```
# load data example
utils::data(
  myGOs,
  package="ViSEAGO"
)
## Not run:
# compute GO terms Semantic Similarity distances
myGOs<-ViSEAGO::compute_SS_distances(
  myGOs,
  distance="Wang"
)

# GOtermsHeatmap with default parameters
Wang_clusters_wardD2<-ViSEAGO::GOterms_heatmap(
  myGOs,
  showIC=TRUE,
  showGOlabels=TRUE,
  GO.tree=list(
    tree=list(
      distance="Wang",
      aggreg.method="ward.D2",
      rotate=NULL
    ),
    cut=list(
      dynamic=list(
        pamStage=TRUE,
        pamRespectsDendro=TRUE,
        deepSplit=2,
        minClusterSize =2
      )
    )
  ),
  samples.tree=NULL
)

## End(Not run)
```

---

GO\_clusters-class      *GO\_clusters class object*

---

### Description

This class is invoked by [GOterms\\_heatmap](#) and [GOclusters\\_heatmap](#) methods to store all results produced.

### Slots

db database source.  
stamp date of stamp.  
organism target species.  
topGO topGO objects summary.  
ont ontology (MF, BP, or CC).  
IC Information Content (IC).  
enrich\_GOs [enrich\\_GO\\_terms-class](#) object.  
terms\_dist distance between GO terms based on semantic similiarity.  
clusters\_dist distance between GO groups based on semantic similiarity.  
hcl\_params Hierarchical clustering parameters used.  
dendrograms GO terms and samples dendrograms.  
samples\_gp samples groups.  
heatmap GO terms and GO groups heatmaps.

### See Also

Other GO\_clusters: [GOclusters\\_heatmap\(\)](#), [clusters\\_cor\(\)](#), [compare\\_clusters\(\)](#), [show\\_heatmap\(\)](#), [show\\_table\(\)](#)

---

GO\_SS-class      *GO\_SS class object definition.*

---

### Description

This class is invoked by [build\\_GO\\_SS](#) method in order to store [enrich\\_GO\\_terms-class](#) object, Information Content (IC), and GO terms or groups distances objects based on semantic similarity.

### Slots

db should be "Bioconductor", "EntrezGene", "Ensembl", or "Uniprot" ressource name.  
stamp date of stamp or annotation version if available in character.  
organism target species GO anotation in character.  
ont used ontology with "MF", "BP", or "CC".  
topGO list with topGO objects summary informations.  
IC Information Content (IC)  
enrich\_GOs [merge\\_enrich\\_terms](#) output object ([enrich\\_GO\\_terms-class](#) object).  
terms\_dist list of GO terms or groups distances objects based on semantic similarity.

**See Also**

Other GO\_semantic\_similarity: [build\\_GO\\_SS\(\)](#), [compute\\_SS\\_distances\(\)](#)

---

MDSplot

*Multi Dimensional Scale (MDS) plot*


---

**Description**

Generate a Multi Dimensional Scale (MDS) plot from distance objects.

**Usage**

```
MDSplot(object, type = "GOterms", file = NULL)
```

```
## S4 method for signature 'ANY'
```

```
MDSplot(object, type = "GOterms", file = NULL)
```

**Arguments**

**object** a [GO\\_SS-class](#) or [GO\\_clusters-class](#) objects from distances computed with [compute\\_SS\\_distances](#).

**type** could be "GOterms" to display GOterms MDSplot, or "GOclusters" to display GOclusters MDSplot.

**file** static image output file name (default to NULL).

**Details**

This method build and display the javascript MDSplot (if file=NULL) from [GO\\_SS-class](#) or [GO\\_clusters-class](#) objects.

A static png image could be printed by setting file argument.

**Value**

a MDS plot.

**Examples**

```
# load data example
utils::data(
  myGOs,
  package="ViSEAGO"
)
## Not run:
# compute GO terms Semantic Similarity distances
myGOs<-ViSEAGO::compute_SS_distances(
  myGOs,
  distance="Wang"
)

# build MDS plot for a GO_SS-class distance object
ViSEAGO::MDSplot(myGOs)
```

```

# GOtermsHeatmap with default parameters
Wang_clusters_wardD2<-ViSEAGO::GOterms_heatmap(
  myGOs,
  showIC=TRUE,
  showGOlabels=TRUE,
  GO.tree=list(
    tree=list(
      distance="Wang",
      aggreg.method="ward.D2",
      rotate=NULL
    ),
    cut=list(
      dynamic=list(
        pamStage=TRUE,
        pamRespectsDendro=TRUE,
        deepSplit=2,
        minClusterSize =2
      )
    )
  ),
  samples.tree=NULL
)

# build MDS plot for a GO_clusters-class distance object, highlighting GO terms clusters.
ViSEAGO::MDSplot(
  Wang_clusters_wardD2,
  "GOterms"
)

# compute clusters of GO terms Semantic Similarity distances
Wang_clusters_wardD2<-ViSEAGO::compute_SS_distances(
  Wang_clusters_wardD2,
  distance="BMA"
)

# GOclusters heatmap
Wang_clusters_wardD2<-ViSEAGO::GOclusters_heatmap(
  Wang_clusters_wardD2,
  tree=list(
    distance="BMA",
    aggreg.method="ward.D2",
    rotate=NULL
  )
)

# build MDS plot for a GO_clusters-class distance object, highlighting GO groups clusters.
ViSEAGO::MDSplot(
  Wang_clusters_wardD2,
  "GOclusters"
)

## End(Not run)

```

**Description**

combine results from GO enrichment tests obtained with **topGO** package, for a given ontology (MF, BP, or CC).

**Usage**

```
merge_enrich_terms(Input, envir = .GlobalEnv)

## S4 method for signature 'list'
merge_enrich_terms(Input, envir = .GlobalEnv)
```

**Arguments**

Input	a list containing named elements. Each element must contain the name of <a href="#">topGOdata-class</a> object created by <a href="#">create_topGOdata</a> method and the associated <a href="#">topGOresult-class</a> object(s).
envir	objects environment (default to .GlobalEnv).

**Details**

This method extracts for each result of GO enrichment test ([topGOresult-class](#) object) and corresponding GO annotations ([topGOdata-class](#) object): informations about GO term (identifiant, name, and description), gene frequency (number of significant genes / Annotated genes), pvalue,  $-\log_{10}(\text{pvalue})$ , significant genes identifiants (GeneID, or Ensembl ID, or uniprot accession), and gene symbols. At the last, this method builds a merged data.table of enriched GO terms ( $p < 0.01$ ) at least once and provides all mentionned columns.

**Value**

an [enrich\\_GO\\_terms-class](#) object.

**References**

Alexa A and Rahnenfuhrer J (2016). topGO: Enrichment Analysis for Gene Ontology. R package version 2.28.0.

Matt Dowle and Arun Srinivasan (2017). data.table: Extension of data.frame. R package version 1.10.4. <https://CRAN.R-project.org/package=data.table>

Herve Pages, Marc Carlson, Seth Falcon and Nianhua Li (2017). AnnotationDbi: Annotation Database Interface. R package version 1.38.0.

**See Also**

Other GO\_terms: [GOcount\(\)](#), [GOterms\\_heatmap\(\)](#), [annotate\(\)](#), [create\\_topGOdata\(\)](#), [gene2GO-class](#)

**Examples**

```
# load genes identifiants (GeneID,ENS...) universe/background (Expressed genes)
background_L<-scan(
  system.file(
    "extdata/data/input",
    "background_L.txt",
    package = "ViSEAGO"
  ),
```



```
    quiet=TRUE,
    what=""
  )

# load Differentially Expressed (DE) gene identifiers from files
PregnantvslactateDE<-scan(
  system.file(
    "extdata/data/input",
    "pregnantvslactateDE.txt",
    package = "ViSEAGO"
  ),
  quiet=TRUE,
  what=""
)

VirginvslactateDE<-scan(
  system.file(
    "extdata/data/input",
    "virginvslactateDE.txt",
    package = "ViSEAGO"
  ),
  quiet=TRUE,
  what=""
)

VirginvspregnantDE<-scan(
  system.file(
    "extdata/data/input",
    "virginvspregnantDE.txt",
    package="ViSEAGO"
  ),
  quiet=TRUE,
  what=""
)

## Not run:
# create topGOdata for BP for each list of DE genes
BP_Pregnantvslactate<-ViSEAGO::create_topGOdata(
  geneSel=PregnantvslactateDE,
  allGenes=background_L,
  gene2GO=myGENE2GO,
  ont="BP",
  nodeSize=5
)

BP_Virginvslactate<-ViSEAGO::create_topGOdata(
  geneSel=VirginvslactateDE,
  allGenes=background_L,
  gene2GO=myGENE2GO,
  ont="BP",
  nodeSize=5
)

BP_Virginvspregnant<-ViSEAGO::create_topGOdata(
  geneSel=VirginvspregnantDE,
  allGenes=background_L,
  gene2GO=myGENE2GO,
  ont="BP",
```

```

    nodeSize=5
  )

  # perform TopGO tests
  elim_BP_Pregnantvslactate<-topGO::runTest(
    BP_L_pregnantvslactate,
    algorithm = "elim",
    statistic = "fisher"
  )

  elim_BP_Virginvslactate<-topGO::runTest(
    BP_L_virginvslactate,
    algorithm = "elim",
    statistic = "fisher"
  )

  elim_BP_Virginvspregnant<-topGO::runTest(
    BP_L_virginvspregnant,
    algorithm = "elim",
    statistic = "fisher"
  )

  # merge topGO results
  BP_sResults<-ViSEAGO::merge_enrich_terms(
    Input=list(
      Pregnantvslactate=c("BP_Pregnantvslactate", "elim_BP_Pregnantvslactate"),
      Virginvslactate=c("BP_Virginvslactate", "elim_BP_Virginvslactate"),
      Virginvspregnant=c("BP_Virginvspregnant", "elim_BP_Virginvspregnant")
    )
  )

  ## End(Not run)

```

---

 myGOs

*myGOs dataset*


---

### Description

an example of object returned by [build\\_GO\\_SS](#) method from mouse functional analysis of mouse mammary gland RNA-Seq (2\_mouse\_bioconductor vignette)

### Usage

```
data(myGOs, package="ViSEAGO")
```

### Format

An object of class [GO\\_SS-class](#).

---

show_heatmap	<i>Display an interactive or static heatmap.</i>
--------------	--

---

### Description

Display a heatmap in interactive or static mode.

### Usage

```
show_heatmap(object, type, file = NULL)
```

```
## S4 method for signature 'GO_clusters,character'
show_heatmap(object, type, file = NULL)
```

### Arguments

object	a <a href="#">GO_clusters-class</a> object from <a href="#">GOterms_heatmap</a> or <a href="#">GOclusters_heatmap</a> .
type	could be "GOterms" to display GOterms clustering heatmap, or "GOclusters" to display GOclusters heatmap.
file	static image output file name (default to NULL).

### Details

This method displays an interactive heatmap (if file=NULL) from [GO\\_clusters-class](#) object for "GOterms" or "GOclusters" type.

A static png image could be printed by setting file argument.

### Value

display or print heatmap.

### See Also

Other enrich\_GO\_terms: [Upset\(\)](#), [enrich\\_GO\\_terms-class](#), [overLapper\(\)](#), [show\\_table\(\)](#)

Other GO\_clusters: [GO\\_clusters-class](#), [GOclusters\\_heatmap\(\)](#), [clusters\\_cor\(\)](#), [compare\\_clusters\(\)](#), [show\\_table\(\)](#)

Other visualization: [GOclusters\\_heatmap\(\)](#), [GOcount\(\)](#), [GOterms\\_heatmap\(\)](#), [Upset\(\)](#), [available\\_organisms\(\)](#), [overLapper\(\)](#), [show\\_table\(\)](#)

### Examples

```
# load data example
data(
  myGOs,
  package="ViSEAGO"
)
## Not run:
# compute GO terms Semantic Similarity distances
myGOs<-ViSEAGO::compute_SS_distances(
  myGOs,
  distance="Wang"
```

```

)

# build MDS plot for a GO_SS-class distance object
ViSEAGO::MDSplot(myGOs)

# GOtermsHeatmap with default parameters
Wang_clusters_wardD2<-ViSEAGO::GOterms_heatmap(
  myGOs,
  showIC=TRUE,
  showGOlabels=TRUE,
  GO.tree=list(
    tree=list(
      distance="Wang",
      aggreg.method="ward.D2",
      rotate=NULL
    ),
    cut=list(
      dynamic=list(
        pamStage=TRUE,
        pamRespectsDendro=TRUE,
        deepSplit=2,
        minClusterSize =2
      )
    )
  ),
  samples.tree=NULL
)

# Display GO terms heatmap
ViSEAGO::show_heatmap(
  Wang_clusters_wardD2,
  "GOterms"
)

# Print GO terms heatmap
ViSEAGO::show_heatmap(
  Wang_clusters_wardD2,
  "GOterms",
  "GOterms_heatmap.png"
)

# compute clusters of GO terms Semantic Similarity distances
Wang_clusters_wardD2<-ViSEAGO::compute_SS_distances(
  Wang_clusters_wardD2,
  distance="BMA"
)

# GOclusters heatmap
Wang_clusters_wardD2<-ViSEAGO::GOclusters_heatmap(
  Wang_clusters_wardD2,
  tree=list(
    distance="BMA",
    aggreg.method="ward.D2",
    rotate=NULL
  )
)

```

```
# Display GO clusters heatmap
ViSEAGO::show_heatmap(
  Wang_clusters_wardD2,
  "GOclusters"
)

# Print GO clusters heatmap
ViSEAGO::show_heatmap(
  Wang_clusters_wardD2,
  "GOclusters",
  "GOclusters_heatmap.png"
)

## End(Not run)
```

---

show_table	<i>Display an interactive or static table.</i>
------------	--

---

## Description

This method is used to display or print the table for [enrich\\_GO\\_terms-class](#) or [GO\\_clusters-class](#) objects.

## Usage

```
show_table(object, file = NULL)

## S4 method for signature 'ANY'
show_table(object, file = NULL)
```

## Arguments

object	an <a href="#">enrich_GO_terms-class</a> object from <a href="#">merge_enrich_terms</a> , or <a href="#">GO_clusters-class</a> object from <a href="#">GOterms_heatmap</a> .
file	table output file name (default to NULL).

## Details

This method displays an interactive table (if file=NULL) from [enrich\\_GO\\_terms-class](#) or [GO\\_clusters-class](#) objects.

The table could be printed by setting file argument.

## Value

display or print table

## References

Yihui Xie (2016). DT: A Wrapper of the JavaScript Library 'DataTables'. R package version 0.2. <https://CRAN.R-project.org/package=DT>

**See Also**

Other enrich\_GO\_terms: [Upset\(\)](#), [enrich\\_GO\\_terms-class](#), [overLapper\(\)](#), [show\\_heatmap\(\)](#)

Other GO\_clusters: [GO\\_clusters-class](#), [GOclusters\\_heatmap\(\)](#), [clusters\\_cor\(\)](#), [compare\\_clusters\(\)](#), [show\\_heatmap\(\)](#)

Other visualization: [GOclusters\\_heatmap\(\)](#), [GOcount\(\)](#), [GOterms\\_heatmap\(\)](#), [Upset\(\)](#), [available\\_organisms\(\)](#), [overLapper\(\)](#), [show\\_heatmap\(\)](#)

**Examples**

```
# load example object
data(
  myGOs,
  package="ViSEAGO"
)

# display merge_enrich_terms output
ViSEAGO::show_table(myGOs)

# print merge_enrich_terms output
ViSEAGO::show_table(
  myGOs,
  "myGOs.txt"
)

## Not run:
# compute GO terms Semantic Similarity distances
myGOs<-ViSEAGO::compute_SS_distances(
  distance="Wang"
)

# GOtermsHeatmap with default parameters
Wang_clusters_wardD2<-ViSEAGO::GOterms_heatmap(
  myGOs,
  showIC=TRUE,
  showGOlabels=TRUE,
  GO.tree=list(
    tree=list(
      distance="Wang",
      aggreg.method="ward.D2",
      rotate=NULL
    ),
    cut=list(
      dynamic=list(
        pamStage=TRUE,
        pamRespectsDendro=TRUE,
        deepSplit=2,
        minClusterSize =2
      )
    )
  ),
  samples.tree=NULL
)

# display table of GO_clusters-class object
ViSEAGO::show_table(Wang_clusters_wardD2)
```

```
# print table of GO_clusters-class object
ViSEAGO::show_table(
  Wang_clusters_wardD2,
  "Wang_clusters_wardD2.txt"
)

## End(Not run)
```

---

Uniprot2GO

*Check available organisms databases at Uniprot.*

---

## Description

Check the **Uniprot-GOA** available organisms.

## Usage

```
Uniprot2GO()
```

## Details

This function downloads the `current_release_numbers` file ([ftp://ftp.ebi.ac.uk/pub/databases/GO/goa/current\\_release\\_numbers](ftp://ftp.ebi.ac.uk/pub/databases/GO/goa/current_release_numbers)) from **Uniprot-GOA** which contains available organisms.

## Value

a `genomic_ressource-class` object required by `annotate`.

## References

Matt Dowle and Arun Srinivasan (2017). `data.table`: Extension of ‘data.frame’. R package version 1.10.4. <https://CRAN.R-project.org/package=data.table>.

Huntley, RP, Sawford, T, Mutowo-Meullenet, P, Shypitsyna, A, Bonilla, C, Martin, MJ, O’Donovan, C (2015). The GOA database: gene Ontology annotation updates for 2015. *Nucleic Acids Res.*, 43, Database issue:D1057-63.

## See Also

Other `genomic_ressource`: [Bioconductor2GO\(\)](#), [Custom2GO\(\)](#), [Ensembl2GO\(\)](#), [EntrezGene2GO\(\)](#), [annotate\(\)](#), [available\\_organisms\(\)](#), `genomic_ressource-class`, [taxonomy\(\)](#)

## Examples

```
# List Uniprot-GOA available organisms
Uniprot<-ViSEAGO::Uniprot2GO()
```

---

Upset

*Enriched GO terms intersections plot.*

---

## Description

This method allows to visualize GO terms intersections between results of enrichment tests.

## Usage

```
Upset(object, file = "./upset.xls")
```

```
## S4 method for signature 'ANY'  
Upset(object, file = "./upset.xls")
```

## Arguments

object	an <a href="#">enrich_GO_terms-class</a> or <a href="#">GO_clusters-class</a> objects.
file	output file name (default to <code>"./upset.xls"</code> )

## Details

This function displays the intersections of enriched GO terms ( $p < 0.01$ ) between all results provided by [enrich\\_GO\\_terms-class](#) or [GO\\_clusters-class](#) objects. The intersections are shown in an upset plot and printed in a table.

## Value

print table and upset.

## See Also

Other [enrich\\_GO\\_terms](#): [enrich\\_GO\\_terms-class](#), [overLapper\(\)](#), [show\\_heatmap\(\)](#), [show\\_table\(\)](#)

Other visualization: [GOclusters\\_heatmap\(\)](#), [GOcount\(\)](#), [GOterms\\_heatmap\(\)](#), [available\\_organisms\(\)](#), [overLapper\(\)](#), [show\\_heatmap\(\)](#), [show\\_table\(\)](#)

## Examples

```
# load example object  
data(  
  myGOs,  
  package="ViSEAGO"  
)  
  
# print upset  
ViSEAGO::Upset(myGOs)
```



**Description**

Easier data mining of biological functions organized into clusters using Gene Ontology and semantic.

**Details**

The main objective of ViSEAGO workflow is to carry out a data mining of biological functions and establish links between genes involved in the study. We developed ViSEAGO in R to facilitate functional Gene Ontology (GO) analysis of complex experimental design with multiple comparisons of interest.

It allows to study large-scale datasets together and visualize GO profiles to capture biological knowledge. The acronym stands for three major concepts of the analysis: Visualization, Semantic similarity and Enrichment Analysis of Gene Ontology ([pkgdiagram](#)).

It provides access to the last current GO annotations ([annotate](#)), which are retrieved from one of NCBI EntrezGene ([Bioconductor2GO](#), [EntrezGene2GO](#)), Ensembl ([Ensembl2GO](#)) or Uniprot ([Uniprot2GO](#)) databases for available species ([available\\_organisms](#)).

ViSEAGO extends classical functional GO analysis ([create\\_topGOdata](#)) to focus on functional coherence by aggregating closely related biological themes while studying multiple datasets at once ([merge\\_enrich\\_terms](#)).

It provides both a synthetic and detailed view using interactive functionalities respecting the GO graph structure ([MDSplot](#), [GOterms\\_heatmap](#), [GOclusters\\_heatmap](#)), and ensuring functional coherence supplied by semantic similarity ([build\\_GO\\_SS](#), [compute\\_SS\\_distances](#)).

ViSEAGO has been successfully applied on several datasets from different species with a variety of biological questions. Results can be easily shared between bioinformaticians and biologists, enhancing reporting capabilities while maintaining reproducibility.

# Index

- \* **GO\_clusters**
  - clusters\_cor, 8
  - compare\_clusters, 11
  - GO\_clusters-class, 29
  - GOclusters\_heatmap, 22
  - show\_heatmap, 35
  - show\_table, 37
- \* **GO\_semantic\_similarity**
  - build\_GO\_SS, 7
  - compute\_SS\_distances, 14
  - GO\_SS-class, 29
- \* **GO\_terms GO\_clusters**
  - semantic\_similarity visualization**
    - MDSplot, 30
- \* **GO\_terms**
  - annotate, 3
  - create\_topGOdata, 16
  - gene2GO-class, 21
  - GOcount, 25
  - GOterms\_heatmap, 26
  - merge\_enrich\_terms, 31
- \* **datasets**
  - myGOs, 34
- \* **enrich\_GO\_terms**
  - enrich\_GO\_terms-class, 19
  - show\_heatmap, 35
  - show\_table, 37
  - Upset, 40
- \* **genomic\_ressource**
  - annotate, 3
  - available\_organisms, 5
  - Bioconductor2GO, 6
  - Custom2GO, 18
  - Ensembl2GO, 19
  - EntrezGene2GO, 20
  - genomic\_ressource-class, 22
  - Uniprot2GO, 39
- \* **semantic\_similarity**
  - GOclusters\_heatmap, 22
  - GOterms\_heatmap, 26
- \* **visualization**
  - available\_organisms, 5
  - GOclusters\_heatmap, 22
  - GOcount, 25
  - GOterms\_heatmap, 26
  - show\_heatmap, 35
  - show\_table, 37
  - Upset, 40
- annotate, 3, 6, 7, 16–18, 20–22, 25, 28, 32, 39, 41
- annotate, character, genomic\_ressource-method (annotate), 3
- available\_organisms, 3, 4, 5, 7, 18, 20–23, 25, 28, 35, 38–41
- available\_organisms, genomic\_ressource-method (available\_organisms), 5
- Bioconductor2GO, 3–6, 6, 18, 20–22, 39, 41
- build\_GO\_SS, 3, 7, 15, 29, 30, 34, 41
- build\_GO\_SS, gene2GO, enrich\_GO\_terms-method (build\_GO\_SS), 7
- clusters\_cor, 8, 12, 23, 29, 35, 38
- clusters\_cor, list, character-method (clusters\_cor), 8
- compare, 8
- compare\_clusters, 9, 11, 23, 29, 35, 38
- compare\_clusters, list-method (compare\_clusters), 11
- compute\_SS\_distances, 7, 8, 14, 23, 26, 27, 30, 41
- compute\_SS\_distances, ANY, character-method (compute\_SS\_distances), 14
- create\_topGOdata, 4, 16, 19, 21, 25, 28, 32, 41
- create\_topGOdata, ANY, ANY, ANY, gene2GO, character, numeric-method (create\_topGOdata), 16
- Custom2GO, 4, 6, 7, 18, 20–22, 39
- cutreeDynamic, 27
- enrich\_GO\_terms-class, 19
- Ensembl2GO, 3–7, 18, 19, 21, 22, 39, 41
- EntrezGene2GO, 3–7, 18, 20, 20, 22, 39, 41
- gene2GO-class, 21
- genomic\_ressource-class, 22
- GO\_clusters-class, 29

GO\_SS-class, 29  
GOclusters\_heatmap, 6, 8, 9, 11, 12, 22, 25, 28, 29, 35, 38, 40, 41  
GOclusters\_heatmap,GO\_clusters,list-method (GOclusters\_heatmap), 22  
GOcount, 4, 6, 17, 21, 23, 25, 28, 32, 35, 38, 40  
GOcount,ANY-method (GOcount), 25  
godata, 7  
GOterms\_heatmap, 4, 6, 8, 11, 15, 17, 21, 23, 25, 26, 29, 32, 35, 37, 38, 40, 41  
GOterms\_heatmap,GO\_SS-method (GOterms\_heatmap), 26

hclust, 23, 26  
heatmaply, 27

listDatasets, 19  
listEnsembl, 19

MDSplot, 30, 41  
MDSplot,ANY-method (MDSplot), 30  
merge\_enrich\_terms, 4, 7, 17, 19, 21, 25, 28, 29, 31, 37, 41  
merge\_enrich\_terms,list-method (merge\_enrich\_terms), 31  
mgoSim, 15  
myGOs, 34

overLapper, 6, 19, 23, 25, 28, 35, 38, 40

pkgdiagram, 41

show\_heatmap, 6, 9, 12, 19, 23, 25, 28, 29, 35, 38, 40  
show\_heatmap,GO\_clusters,character-method (show\_heatmap), 35  
show\_table, 6, 9, 12, 19, 23, 25, 28, 29, 35, 37, 40  
show\_table,ANY-method (show\_table), 37

taxonomy, 4, 6, 7, 18, 20–22, 39

Uniprot2GO, 3–7, 18, 20–22, 39, 41  
Upset, 6, 19, 23, 25, 28, 35, 38, 40  
Upset,ANY-method (Upset), 40  
useEnsembl, 19

ViSEAGO, 41