Introduction to the GOxploreR package

Kalifa Manjang, Shailesh Tripathi, Olli Yli-Harja, Matthias Dehmer & Frank Emmert-Streib

07 June 2021

Predictive Society and Data Analytics Lab, Tampere University, Tampere, Korkeakoulunkatu 10, 33720, Tampere, Finland

Citation

Kalifa Manjang, Shailesh Tripathi, Olli Yli-Harja, Matthias Dehmer, and Frank Emmert-Streib. 2020. Graph-based exploitation of gene ontology using goxplorer for scrutinizing biological significance. Scientific Reports 10, 1: 16672. https://doi.org/10.1038/s41598-020-73326-3 [2]

Contents

List of Figures

List of Tables

Introduction

The GOxploreR package is an R package that provides a simple and efficient way to communicate with the gene ontology (GO) database. The gene ontology is a major bioinformatics initiative by the gene ontology consortium. The goal is to categorize the gene and gene product function. The ontology is structured into three distinct aspects of gene function: molecular function (MF), cellular component (CC), and biological process (BP) together with over 45,000 terms and 130,000 relations whereas the majority of information is centered around ten model organisms [1]. In addition, GO includes annotations by linking specific gene products to GO-terms. Currently, GO is the most comprehensive and widely used knowledgebase concerning functional information about genes.

Organism	Name used as options in the package	Number of genes
Human	"Homo sapiens" / "Human"	19155
Mouse	"Mus musculus" / "Mouse"	20929
Caenorhabditis elegans	"Caenorhabditis elegans" / "Worm"	14697
Drosophila melanogaster	"Drosophila melanogaster" / "Fruit fly"	12683
Rat	"Rattus norvegicus" / "Rat"	19383
Baker's yeast	"Saccharomyces cerevisiae" / "Yeast"	5502
Zebrafish	"Danio rerio" / "Zebrafish"	20718
Arabidopsis thaliana	"Arabidopsis thaliana" / "Cress"	25891
Schizosaccharomyces pombe	"Schizosaccharomyces pombe" / "Fission yeast"	5055
Escherichia coli	"Escherichia coli" / "E.coli"	3449

Table 1: Organisms supported by the GOxploreR package.

This vignette gives an overview of the functionality provided by the GOxploreR package.

The package is freely available on CRAN and can be installed using the following command:

```
install.packages("GOxploreR")
```

The package function can be loaded using:

```
library(GOxploreR)
```

Note that the package needs to be installed to be loaded.

Overview of the functionality of the package

The following is a brief description of the package functionality.

Gene2GOTermAndLevel

The Gene2GOTermAndLevel function provides information associated with a list of genes. Given a gene or a list of genes, an organism, and a domain (BP, MF or CC) the function provides the Gene Ontology terms (GO-terms) associated with the genes and their respective levels of the DAG. The default argument of the domain is BP. For the arguments of the option 'organism' see Table 1.

```
# The cellular component gene ontology terms will be retrieve and their levels
Gene2GOTermAndLevel(genes = c(10212, 9833, 6713), organism = "Homo sapiens", domain = "CC")
```

```
#>    Entrezgene ID      GO ID Domain Level
#> 1          10212 GO:0005634     CC     5
#> 2          10212 GO:0005737     CC     2
#> 3          10212 GO:0016607     CC     9
#> 4          10212 GO:0016020     CC     2
#> 5          10212 GO:0005654     CC     7
#> 6           9833 GO:0016020     CC     2
```

```
#> 7           9833 GO:0005886     CC     3
#> 8           9833 GO:0005938     CC     3
#> 9           9833 GO:0005634     CC     5
#> 10          9833 GO:0005737     CC     2
#> 11          6713 GO:0016021     CC     4
#> 12          6713 GO:0016020     CC     2
#> 13          6713 GO:0005783     CC     5
#> 14          6713 GO:0005789     CC     5
#> 15          6713 GO:0043231     CC     4
```

```
# The biological process gene ontology terms will be retrieve and their levels
Gene2GOTermAndLevel(genes = c(100000642, 30592, 58153, 794484), organism = "Danio rerio")
```

```
#>    Entrezgene ID      GO ID Domain Level
#> 1      100000642 GO:0007186     BP     5
#> 2      100000642 GO:0050911     BP     7
#> 3          30592 GO:0045214     BP     9
#> 4          30592 GO:0060047     BP     5
#> 5          30592 GO:0060038     BP     9
#> 6          30592 GO:0048738     BP     7
#> 7          30592 GO:0055005     BP    11
#> 8          30592 GO:0055015     BP    10
#> 9          30592 GO:0055004     BP    11
#> 10         30592 GO:0045823     BP     7
#> 11        794484 GO:0008150     BP     0
```

```
# The molecular function gene ontology terms will be retrieve and their levels
Gene2GOTermAndLevel(genes = c(100009600, 18131, 100017), organism = "Mouse", domain = "MF")
```

```
#>    Entrezgene ID      GO ID Domain Level
#> 1      100009600 GO:0008270     MF     6
#> 2      100009600 GO:0043565     MF     5
#> 3      100009600 GO:0046872     MF     4
#> 4      100009600 GO:0003677     MF     4
#> 5          18131 GO:0005515     MF     2
#> 6          18131 GO:0005509     MF     5
#> 7          18131 GO:0038023     MF     2
#> 8          18131 GO:0042802     MF     3
#> 9          18131 GO:0019899     MF     3
#> 10        100017 GO:0005515     MF     2
#> 11        100017 GO:0035650     MF     3
#> 12        100017 GO:0001784     MF     5
#> 13        100017 GO:0005102     MF     3
#> 14        100017 GO:0005546     MF     8
#> 15        100017 GO:0001540     MF     4
#> 16        100017 GO:0050750     MF     5
#> 17        100017 GO:0030159     MF     4
#> 18        100017 GO:0030276     MF     3
#> 19        100017 GO:0035612     MF     3
#> 20        100017 GO:0035591     MF     3
#> 21        100017 GO:0035615     MF     4
```

Gene2GOTermAndLevel_ON

This function is similar to the Gene2GOTermAndLevel function, the only difference is that this function queries the Ensembl database online (ON) for GO-terms (making it relatively slow). That means the results

from the Gene2GOTermAndLevel_ON function are always up to date but an internet connection is needed for its execution. This function does not provide support for Escherichia coli.

```
# The cellular component gene ontology terms will be retrieve and their levels
Gene2GOTermAndLevel_ON(genes = c(10212, 9833, 6713), organism = "Homo sapiens", domain ="CC")

# The biological process gene ontology terms will be retrieve and their levels
Gene2GOTermAndLevel_ON(genes = c(100000711, 100000710, 100000277), organism = "Danio rerio")

# The molecular function gene ontology terms will be retrieve and their levels
Gene2GOTermAndLevel_ON(genes = c(100009609, 100017, 100034361), organism = "Mouse", domain = "MF")
```

GOTermXXOnLevel

This function gives the level of a GO-term based on a DAG. The results for organism-specific GO-DAGs are the same as for the general GO-DAG. The XX in the name above should be replaced by either BP, MF, or CC.

```
# Retrieve the level of a GO biological process term
goterms <- c("GO:0009083","GO:0006631","GO:0006629","GO:0014811","GO:0021961")
GOTermBPOnLevel(goterm = goterms)
```

```
#>         Term Level
#> 1 GO:0009083     8
#> 2 GO:0006631     7
#> 3 GO:0006629     3
#> 4 GO:0014811    19
#> 5 GO:0021961    15
```

```
# Retrieve the level of a GO molecular function term
goterms <- c("GO:0005515","GO:0016835","GO:0046976","GO:0015425","GO:0005261")
GOTermMFOnLevel(goterm = goterms)
```

```
#>         Term Level
#> 1 GO:0005515     2
#> 2 GO:0016835     3
#> 3 GO:0046976     9
#> 4 GO:0015425     9
#> 5 GO:0005261     6
```

```
# Retrieve the level of a GO cellular component term
goterms <- c("GO:0055044","GO:0030427","GO:0036436","GO:0034980","GO:0048226")
GOTermCCOnLevel(goterm = goterms)
```

```
#>         Term Level
#> 1 GO:0055044     2
#> 2 GO:0030427     2
#> 3 GO:0036436    12
#> 4 GO:0034980     6
#> 5 GO:0048226     7
```

Level2GOTermXX

This function gives all the GO-terms from a given GO-level. These GO-terms can be from the general GO-DAG or from an organism-specific GO-DAG. If the "organism" argument is given, the GO-terms will be acquired from the organism's (organism supported by the package) DAG level, However, if no value for the "organism" parameter is given then the general GO-DAG is used (default). The XX in the name should be replaced by either BP, MF, or CC.

```
# Retrieve all the GO-terms from a particular GO BP level
Level2GOTermBP(level = 1, organism = "Human")
```

```
#>  [1] "GO:0008152" "GO:0032502" "GO:0002376" "GO:0048511" "GO:0043473"
#>  [6] "GO:0040011" "GO:0023052" "GO:0009987" "GO:0000003" "GO:0007610"
#> [11] "GO:0050896"
```

```
# Retrieve all the GO-terms from a particular GO MF level
Level2GOTermMF(level = 14, organism = "Rat")
```

```
#> [1] "GO:0005391" "GO:0008553" "GO:0086039" "GO:0046961" "GO:1905056"
#> [6] "GO:1905059" "GO:0008900"
```

```
# Retrieve all the GO-terms from the general GO CC level
Level2GOTermCC(level = 16)
```

```
#> NULL
```

Level2LeafNodeXX

This function gives all the leaf nodes from a particular GO-level. Leaf nodes can also be attained from the organism-specific GO-DAG. The "organism" parameter is optional. If supplied, the leaf node from the respective organism's level will be acquired. The default is the general GO-DAG. The XX should be substituted with either BP, MF, or CC.

```
# Get all leaf nodes from a GO BP level
Level2LeafNodeBP(level = 2, organism = "Danio rerio")
```

```
#>  [1] "GO:0006807" "GO:0007586" "GO:0032259" "GO:0030431" "GO:0035176"
#>  [6] "GO:0032504" "GO:1990845" "GO:0036268" "GO:0019835" "GO:0045730"
#> [11] "GO:0007624" "GO:0090618"
```

```
# Get all leaf nodes from a GO MF level
Level2LeafNodeMF(level = 13)
```

```
#> NULL
```

```
# Get all leaf nodes from a GO CC level
Level2LeafNodeCC(level = 15, organism = "Schizosaccharomyces pombe")
```

```
#> [1] "No such level exist for Schizosaccharomyces pombe, the highest level is 12"
```

Level2JumpNodeXX

This function gives for a GO-level the GO-terms which correspond to jump Nodes (JNs). The JNs are GO-terms which have at least one child term not present in the level below the parent term. If no organism is given, the default is the general GO-DAG. The XX in the name should be substituted with BP, MF, or CC.

```
# All jump nodes from the GO BP level
head(Level2JumpNodeBP(level = 2, organism = "Homo sapiens"))
```

```
#> [1] "GO:0007155" "GO:0007568" "GO:0048856" "GO:0007154" "GO:0006955"
#> [6] "GO:0045730"
```

```
# All jump nodes from the GO MF level
head(Level2JumpNodeMF(level = 3, organism = "Homo sapiens"))
```

```
#> [1] "GO:0019239" "GO:0008233" "GO:0035591" "GO:0008047" "GO:0004888"
#> [6] "GO:0042393"
```

```
# All jump nodes from the GO CC level
head(Level2JumpNodeCC(level = 7, organism = "Homo sapiens"))
```

```
#> [1] "GO:0060205" "GO:0005882" "GO:0042641" "GO:0030139" "GO:0033017"
#> [6] "GO:0030659"
```

Level2RegularNodeXX

This function gives for a GO-level the GO-terms which correspond to regular Nodes (RNs). The RNs are those GO-terms whose child terms are all present in the level right below the parent's level. The XX in the name should be subsitituted with BP, MF, or CC.

```
# All regular nodes from the BP level
head(Level2RegularNodeBP(level = 9, organism = "Zebrafish"))
```

```
#> [1] "GO:0002088" "GO:0060396" "GO:0048663" "GO:0071688" "GO:0014866"
#> [6] "GO:0019229"
```

```
# All regular nodes from the MF level
head(Level2RegularNodeMF(level = 7, organism = "Homo sapiens"))
```

```
#> [1] "GO:0005244" "GO:0000976" "GO:0016531" "GO:0004725" "GO:0004983"
#> [6] "GO:0004722"
```

```
# All jump nodes from the CC level
head(Level2RegularNodeCC(level = 7))
```

```
#> [1] "GO:0002102" "GO:0015934" "GO:0015935" "GO:0090533" "GO:0032160"
#> [6] "GO:0032161"
```

Level2NoLeafNodeXX

This function gives all the GO-terms from a GO-level that are not leaf nodes. Similarly, all non-leaf GO-terms from an organism-specific DAG can be returned by providing the organism of interest. The default is the general GO-DAG. The XX in the name should be substituted with either BP, MF or CC.

```
# All GO-terms on a particular GO BP level that are not leaf nodes
Level2NoLeafNodeBP(level = 16, organism = "Homo sapiens")
```

```
#>  [1] "GO:0072540" "GO:0014808" "GO:0060314" "GO:0045623" "GO:0051281"
#>  [6] "GO:0051280" "GO:0045625" "GO:0031585" "GO:0045624" "GO:0021966"
```

```
# All GO-terms on a particular GO MF level that are not leaf nodes
Level2NoLeafNodeMF(level = 11, organism = "Caenorhabditis elegans")
```

```
#> NULL
```

```
# All GO-terms on a particular GO CC level that are not leaf nodes
Level2NoLeafNodeCC(level = 12, organism = "Homo sapiens")
```

```
#> [1] "GO:0098675"
```

getGOcategory

Given a GO-term or a list of GO-terms, this function returns the category of the term. The categories are jump nodes (JN), regular nodes (RN) and leaf nodes (LN).

```
goterm <- c("GO:0009083","GO:0006631","GO:0006629","GO:0016835","GO:0046976","GO:0048226")
```

```
# Returns the categories of the GO-terms in the list
getGOcategory(goterm = goterm)
```

```
#>         Term Category Domain
#> 1 GO:0009083       JN     BP
#> 2 GO:0006631       JN     BP
#> 3 GO:0006629       JN     BP
#> 4 GO:0016835       RN     MF
#> 5 GO:0046976       LN     MF
#> 6 GO:0048226       LN     CC
```

degreeDistXX

This function obtains the degree distribution of the GO-terms on a GO-level. A bar plot is obtained which shows how many nodes in the GO-level have a certain degree k. The XX in the name should be substituted with either BP, MF, or CC.

```
# Degree distribution of the GO-terms on a particular GO BP level
degreeDistBP(level = 4)
```

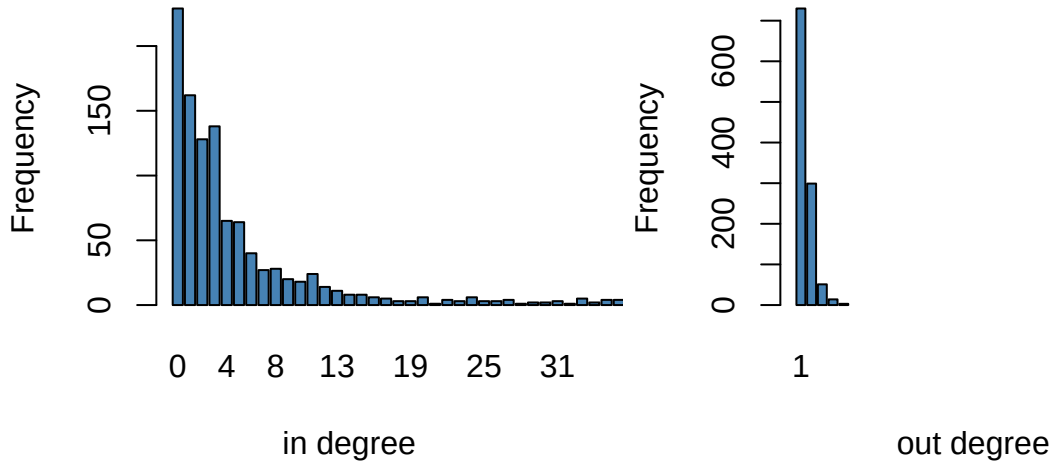

Figure 1: Degree distribution of the biological process GO-terms on level 4.

```
# Degree distribution of the GO-terms on a particular GO MF level
degreeDistMF(level = 2)
```

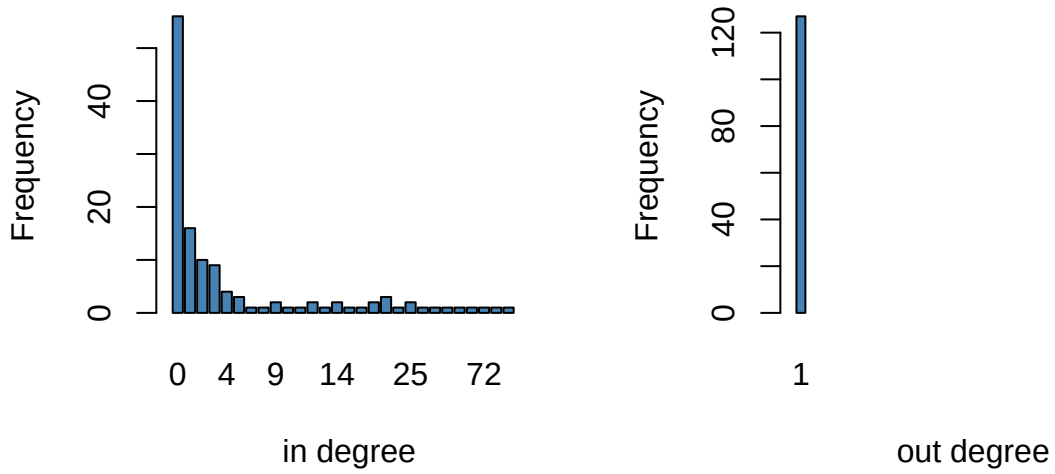

Figure 2: Degree distribution of the molecular function GO-terms on level 2.

```
# Degree distribution of the GO-terms on a particular GO CC level
degreeDistCC(level = 10)
```

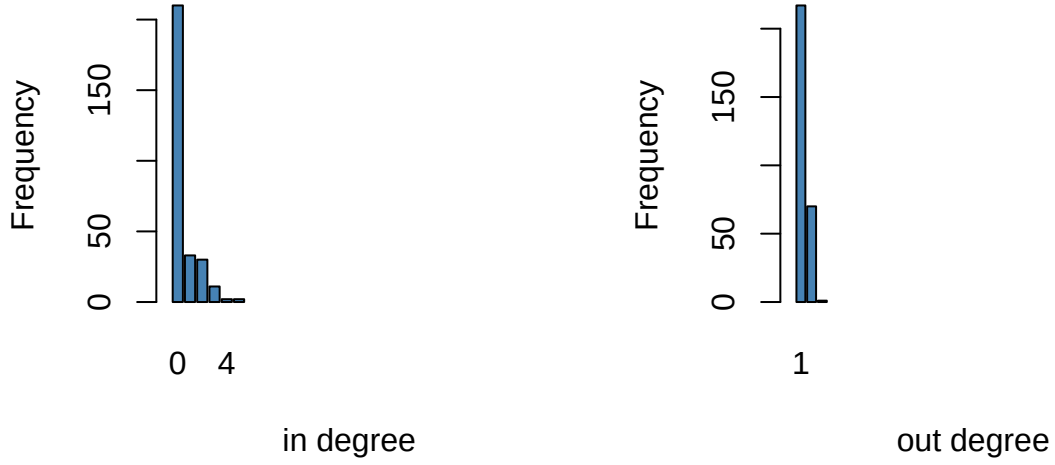

Figure 3: Degree distribution of the cellular component GO-terms on level 10.

GOTermXX2ChildLevel

For a GO-term it's children level are derived. The XX in the name should be substituted with BP, MF, or CC.

```
# Get the level of a GO BP term's children
GOTermBP2ChildLevel(goterm = "GO:0007635")
```

```
#> $Terms
#> [1] "GO:0007636" "GO:0007637" "GO:0042048" "GO:0061366"
#>
#> $Level
#> [1] 5 7 4 6
```

```
# Get the level of a GO MF term's children
GOTermMF2ChildLevel(goterm = "GO:0098632")
```

```
#> $Terms
#> [1] "GO:0086080" "GO:0098641"
#>
#> $Level
#> [1] 6 6
```

```
# Get the level of a GO CC term's children
GOTermCC2ChildLevel(goterm = "GO:0071735")
```

```
#> $Terms
#> [1] "GO:0071736" "GO:0071737"
#>
#> $Level
#> [1] 5 7
```

GetLeafNodesXX

This function gives all the leaf nodes of a certain organism. If the input value is empty or is "BP", "MF"" or "CC"" the default DAG is the general GO-DAG. The value for XX should be subsituted with BP, MF or CC.

```
# All leaf nodes from the GO BP tree
GetLeafNodesBP("BP")

# All leaf nodes from the GO CC tree
GetLeafNodesCC(organism = "Caenorhabditis elegans")
```

GO2DecXX

This function returns all descedant child nodes of a GO-term. That means, we begin from a GO-term and find all the GO-terms of its children and their children until we reach all the way down of the DAG. The XX in the name should be substituted with BP, MF or CC.

```
# Biological process GO-term descendant terms
GO2DecBP(goterm = "GO:0044582")
```

```
#> [1] "GO:1900497" "GO:1900498" "GO:1900499"
```

```
# Molecular function GO-term descendant terms
GO2DecMF(goterm = "GO:0036442")
```

```
#> NULL
```

```
# Cellular component GO-term descendant terms
GO2DecCC(goterm = "GO:0031233")
```

```
#> [1] "GO:0031237" "GO:0031362" "GO:0071575" "GO:1990914"
```

GetDAG

This function gives the GO-terms in the Gene Ontology as an edgelist corresponding to a directed acyclic graph (DAG) for the GO-terms of a certain organism. This can also be obtained for the general GO-DAG (not organism-specific).

```
# Represent all the BP gene association GO-terms for human as an edgelist
head(GetDAG(organism = "Human", domain = "BP"))
```

```
#>      [,1]         [,2]
#> [1,] "GO:0008150" "GO:0000003"
#> [2,] "GO:0008150" "GO:0002376"
```

```
#> [3,] "GO:0008150" "GO:0007610"
#> [4,] "GO:0008150" "GO:0008152"
#> [5,] "GO:0008150" "GO:0009758"
#> [6,] "GO:0008150" "GO:0009987"
```

```
# Represent all the MF gene association GO-terms for Mouse as an edgelist
head(GetDAG(organism = "Mouse", domain = "MF"))
```

```
#>      [,1]         [,2]
#> [1,] "GO:0003674" "GO:0140110"
#> [2,] "GO:0003674" "GO:0003824"
#> [3,] "GO:0003674" "GO:0038024"
#> [4,] "GO:0003674" "GO:0045735"
#> [5,] "GO:0003674" "GO:0005198"
#> [6,] "GO:0003674" "GO:0005215"
```

```
# Represent all the CC gene association GO-terms for Caenorhabditis elegans as an edgelist
head(GetDAG(organism = "Caenorhabditis elegans", domain = "CC"))
```

```
#>      [,1]         [,2]
#> [1,] "GO:0005575" "GO:0005622"
#> [2,] "GO:0005575" "GO:0032991"
#> [3,] "GO:0005575" "GO:0110165"
#> [4,] "GO:0005622" "GO:0000151"
#> [5,] "GO:0005622" "GO:0000159"
#> [6,] "GO:0005622" "GO:0000178"
```

visRDAGXX

The visualization of a GO-DAG is difficult primarily because of the size of the graphs containing thousands of GO-terms. For this reason, we invented a simple method that combines GO-terms with similar characteristics together. This includes a global summary of all GO-terms in the DAG. Every node in the reduced DAG comprises 1 or more GO-terms and these GO-terms can be accessed by using certain information (i.e. the level and what type of node category they represent for example "RN", "JN" or "LN"). This is what we call the reduced GO-DAG for an organism. Furthermore, the function returns a list which contains the GO-terms in each category and the plot of the reduced DAG. To retrieve just the GO-terms in each category, the "plot" argument can be set to "FALSE" . The XX in the name should be substituted with BP, MF or CC. The total number of GO-terms in each node is represented by the node label. For instance, in Figure 4 , Level 0 (i.e "L0 RN") has 1 GO-term present in the node category.

The label "J","R" and "L" on the right side of Figure 4 gives the number of connections between the regular node (RN) on the level and the nodes right below it (RN are nodes that have all their children nodes represented in the next level). For example, on L1, The label J = 5, R = 9 and L = 6 means that the RNs on the level have 5 of it's children nodes as Jump nodes (JN) on L2, 9 of it's descendant are Regular nodes (RN) and 6 of its children GO-terms on L2 are leaf nodes (LN).

```
# The GO-terms in each node category of the reduced Caenorhabditis elegans GO-DAG
head(visRDAGMF(organism = "Caenorhabditis elegans", plot = FALSE))
```

```
#> $`L5 RN`
#>  [1] "GO:0004672" "GO:0000030" "GO:0016791" "GO:0008528" "GO:0004519"
#>  [6] "GO:0050661" "GO:0008417" "GO:0043565" "GO:0003727" "GO:0016174"
#> [11] "GO:0016972" "GO:0050660" "GO:0061630" "GO:0042626" "GO:0004197"
#> [16] "GO:0004180" "GO:0008017" "GO:0035091" "GO:0019901" "GO:0030295"
#> [21] "GO:0003857" "GO:0051377" "GO:0019843" "GO:0008135" "GO:0004104"
#> [26] "GO:0003729" "GO:0016597" "GO:0004190" "GO:0004177" "GO:0016307"
#> [31] "GO:0000828" "GO:0035673" "GO:0015927" "GO:0004540" "GO:0070491"
```

```
#> [36] "GO:0030515" "GO:0051287" "GO:0008374" "GO:0004368" "GO:0004000"
#> [41] "GO:0003725" "GO:0004611" "GO:0017076" "GO:0004407" "GO:0004620"
#> [46] "GO:0019171" "GO:0004864" "GO:0008376" "GO:0008320" "GO:0004527"
#> [51] "GO:0008081" "GO:0008375" "GO:0003997" "GO:0005179" "GO:0004866"
#> [56] "GO:0015165" "GO:0031543" "GO:0036002" "GO:0017069" "GO:0019903"
#> [61] "GO:0031490" "GO:0008378" "GO:0015020" "GO:0003953" "GO:0005160"
#> [66] "GO:0019205" "GO:0070569" "GO:0016462" "GO:0043175" "GO:0008173"
#> [71] "GO:0015929" "GO:0004311" "GO:0004470" "GO:0052742" "GO:0004353"
#> [76] "GO:0008235" "GO:0005092" "GO:0016407" "GO:0003884" "GO:0035596"
#> [81] "GO:0001727" "GO:0018455" "GO:0004731" "GO:0008318" "GO:0004753"
#> [86] "GO:0004576" "GO:0004645" "GO:0072542" "GO:0061629" "GO:0033613"
#> [91] "GO:0034338" "GO:0004084" "GO:0005104" "GO:0046914" "GO:0034979"
#> [96] "GO:0051723" "GO:0008227" "GO:0033744" "GO:0016410"
#>
#> $`L9 LN`
#>  [1] "GO:0052906" "GO:0060002" "GO:0003724" "GO:0031151" "GO:0032143"
#>  [6] "GO:0032142" "GO:0032181" "GO:0015280" "GO:0046975" "GO:0000062"
#> [11] "GO:0004671" "GO:0046961" "GO:0046933" "GO:0008510" "GO:0015616"
#> [16] "GO:0004439" "GO:0008311" "GO:0000979" "GO:0008553" "GO:0015410"
#> [21] "GO:0005391" "GO:0008310" "GO:0022841" "GO:0005005" "GO:0005025"
#> [26] "GO:0042800" "GO:0015279" "GO:0033192" "GO:0001162" "GO:0043813"
#> [31] "GO:0015141" "GO:0052905" "GO:0022889" "GO:0005313" "GO:0003918"
#> [36] "GO:0005366" "GO:0046974" "GO:0042799" "GO:1905502" "GO:0001164"
#> [41] "GO:0005415" "GO:0015183" "GO:0005247" "GO:0019799" "GO:0015131"
#> [46] "GO:0015140" "GO:0005248" "GO:0005229" "GO:0005228" "GO:0052909"
#> [51] "GO:0016316" "GO:0000064" "GO:0015181" "GO:0015189" "GO:0034597"
#> [56] "GO:1990189" "GO:0005436"
#>
#> $`L3 RN`
#>  [1] "GO:0016757" "GO:0004888" "GO:0004497" "GO:0016705" "GO:0003700"
#>  [6] "GO:0016684" "GO:0003676" "GO:0019899" "GO:0005548" "GO:0043021"
#> [11] "GO:0050839" "GO:0016772" "GO:0031369" "GO:0042802" "GO:0016788"
#> [16] "GO:0030276" "GO:0003712" "GO:0016651" "GO:0016849" "GO:0008092"
#> [21] "GO:0016798" "GO:0016765" "GO:0019904" "GO:0046332" "GO:0016746"
#> [26] "GO:0000149" "GO:0016627" "GO:0008134" "GO:0046983" "GO:0016810"
#> [31] "GO:0019825" "GO:0140223" "GO:0051536" "GO:0031072" "GO:0016846"
#> [36] "GO:0042277" "GO:0140312" "GO:0051213" "GO:0019207" "GO:0051018"
#> [41] "GO:0016614" "GO:0016817" "GO:0016638" "GO:0015144" "GO:0099106"
#> [46] "GO:0005539" "GO:0019838" "GO:0005496" "GO:0016701" "GO:0016886"
#> [51] "GO:0016866"
#>
#> $`L10 RN`
#> [1] "GO:0004970" "GO:1904315" "GO:0005283" "GO:0010485" "GO:0015271"
#>
#> $`L7 LN`
#>   [1] "GO:0004715" "GO:0045174" "GO:0004675" "GO:0003924" "GO:0034046"
#>   [6] "GO:0004693" "GO:0003691" "GO:0061575" "GO:0008353" "GO:0032357"
#>  [11] "GO:0070273" "GO:0008198" "GO:0008199" "GO:0032137" "GO:0004385"
#>  [16] "GO:0003688" "GO:0004741" "GO:0072518" "GO:0003873" "GO:1990817"
#>  [21] "GO:0019706" "GO:0015440" "GO:0052591" "GO:0003968" "GO:0004343"
#>  [26] "GO:0043047" "GO:0004320" "GO:0016295" "GO:0016296" "GO:0004313"
#>  [31] "GO:0004340" "GO:0008865" "GO:0019158" "GO:0003980" "GO:0004170"
#>  [36] "GO:0004651" "GO:0004484" "GO:0004709" "GO:0008143" "GO:0008266"
#>  [41] "GO:0032266" "GO:0008107" "GO:0032041" "GO:0008254" "GO:0070260"
```

```
#>  [46] "GO:0004758" "GO:0010314" "GO:0015199" "GO:0047220" "GO:0016531"
#>  [51] "GO:0008441" "GO:0047499" "GO:0004558" "GO:0070140" "GO:0008330"
#>  [56] "GO:0004378" "GO:0008240" "GO:0046525" "GO:0004983" "GO:0000774"
#>  [61] "GO:0004849" "GO:0052918" "GO:0052926" "GO:0004742" "GO:0016263"
#>  [66] "GO:0000014" "GO:0004362" "GO:0003872" "GO:0004017" "GO:0046403"
#>  [71] "GO:0003985" "GO:0052717" "GO:0003983" "GO:0004382" "GO:0045134"
#>  [76] "GO:0052925" "GO:0004135" "GO:0004095" "GO:0004001" "GO:0004798"
#>  [81] "GO:0003841" "GO:0008890" "GO:0008253" "GO:0070037" "GO:0004861"
#>  [86] "GO:0004032" "GO:0016287" "GO:0004662" "GO:0004483" "GO:0004844"
#>  [91] "GO:0036424" "GO:0036425" "GO:0008271" "GO:0004663" "GO:0004377"
#>  [96] "GO:0047710" "GO:1990259" "GO:0016300" "GO:0016435" "GO:0044594"
#> [101] "GO:0005547" "GO:0032034" "GO:0017005" "GO:0046964" "GO:0004024"
#> [106] "GO:0004581" "GO:0004144" "GO:0003747" "GO:0004149" "GO:0015095"
#> [111] "GO:0016922" "GO:0008349" "GO:0052917" "GO:0052824" "GO:0003934"
#> [116] "GO:0008519" "GO:0052692" "GO:0101006" "GO:1903763" "GO:0052650"
#> [121] "GO:0015227" "GO:0004435" "GO:0004780" "GO:0015491" "GO:0004711"
#> [126] "GO:0008466" "GO:0004679" "GO:0008028" "GO:0042392" "GO:0016231"
#> [131] "GO:0004703" "GO:0004687" "GO:0055077" "GO:0034736" "GO:0004092"
#> [136] "GO:0003825" "GO:0046969" "GO:0004656" "GO:0004482" "GO:0018114"
#> [141] "GO:0030378" "GO:0008120" "GO:0005355" "GO:0004694" "GO:0015136"
#> [146] "GO:0030297" "GO:0034431" "GO:0034432" "GO:0004862" "GO:0008458"
#> [151] "GO:0050211" "GO:0001641" "GO:0106050" "GO:0005185"
#>
#> $`L1 RN`
#> [1] "GO:0003824" "GO:0005198" "GO:0060090"
```

```
# RN GO-terms on level 1 can be access as follows
visRDAGMF(organism = "Caenorhabditis elegans", plot = FALSE)$"L1 RN"
```

```
#> [1] "GO:0003824" "GO:0005198" "GO:0060090"
```

```
# JN GO-terms on level 9 can be access as follows
visRDAGMF(organism = "Caenorhabditis elegans", plot = FALSE)$"L9 JN"
```

```
#> [1] "GO:0015379"
```

```
# LN GO-terms on level 14 can be access as follows
visRDAGMF(organism = "Caenorhabditis elegans", plot = FALSE)$"L14 LN"
```

```
#> NULL
```

```
# Represent the molecular function GO-DAG for organism Caenorhabditis elegans
visRDAGMF(organism = "Caenorhabditis elegans", plot = TRUE)[["plot"]]
```

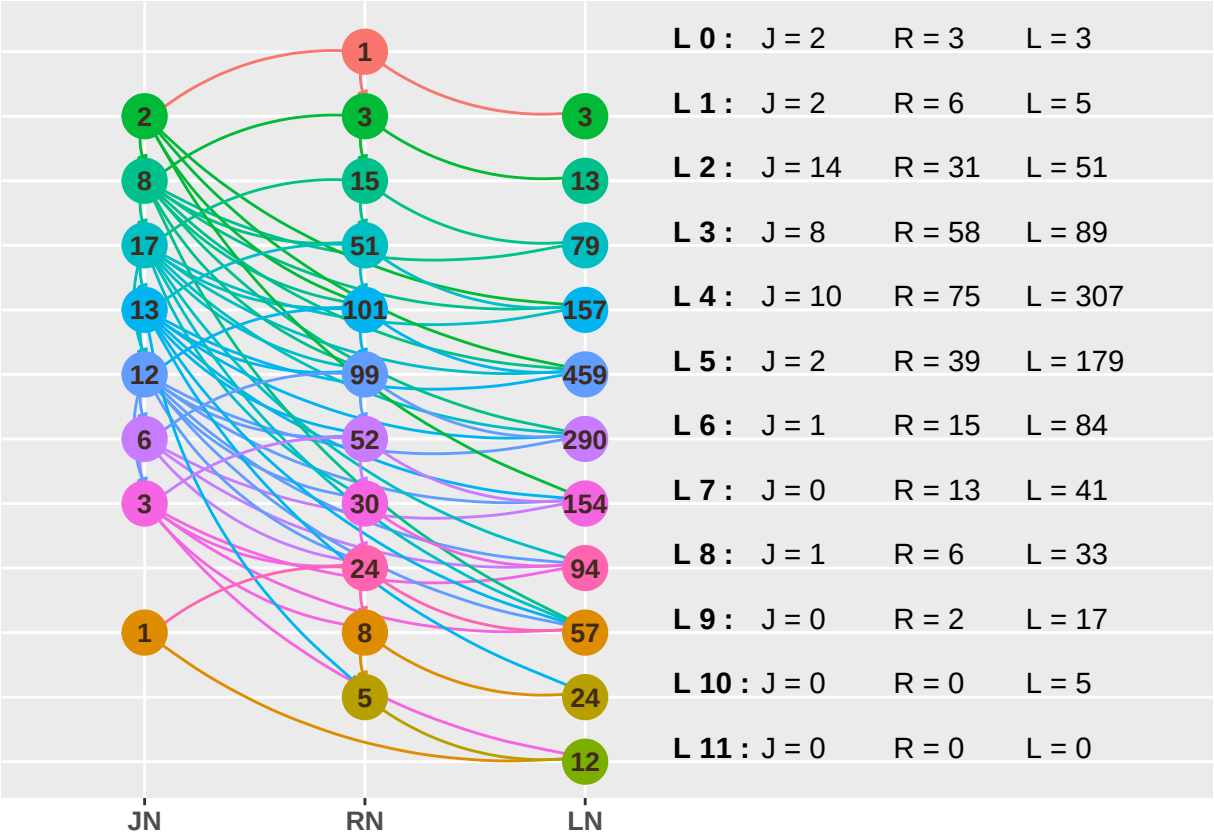

Figure 4: Visualization of a reduced GO-DAG for Caenorhabditis elegans.

visRsubDAGXX

The visRsubDAGXX function is similar to the visRDAGXX function, however, it visualizes an organism-specific sub-GO-DAG. The input of the function is a list of organism-specific GO-terms. If this list contains not all GO-terms of the organism, then category nodes are faded out. The XX in the function can be substituted with BP, MF or CC.

```
Terms <- c("GO:0022403","GO:0000278","GO:0006414","GO:0006415","GO:0006614",
           "GO:0045047","GO:0072599","GO:0006613","GO:0000279","GO:0000087",
           "GO:0070972","GO:0000184","GO:0000280","GO:0007067","GO:0006413",
           "GO:0048285","GO:0006412","GO:0000956","GO:0006612","GO:0019080",
           "GO:0019083","GO:0016071","GO:0006402","GO:0043624","GO:0043241",
           "GO:0006401","GO:0072594","GO:0022904","GO:0019058","GO:0032984",
           "GO:0045333","GO:0006259","GO:0051301","GO:0022900","GO:0006396",
           "GO:0060337","GO:0071357","GO:0034340","GO:0002682","GO:0051320",
           "GO:0045087","GO:0051325","GO:0022411","GO:0016032","GO:0044764",
           "GO:0022415","GO:0051329","GO:0050776","GO:0030198","GO:0043062")

# visualisation the DAG node categories of the given biological process GO-terms
visRsubDAGBP(goterm = Terms, organism = "Human")
```

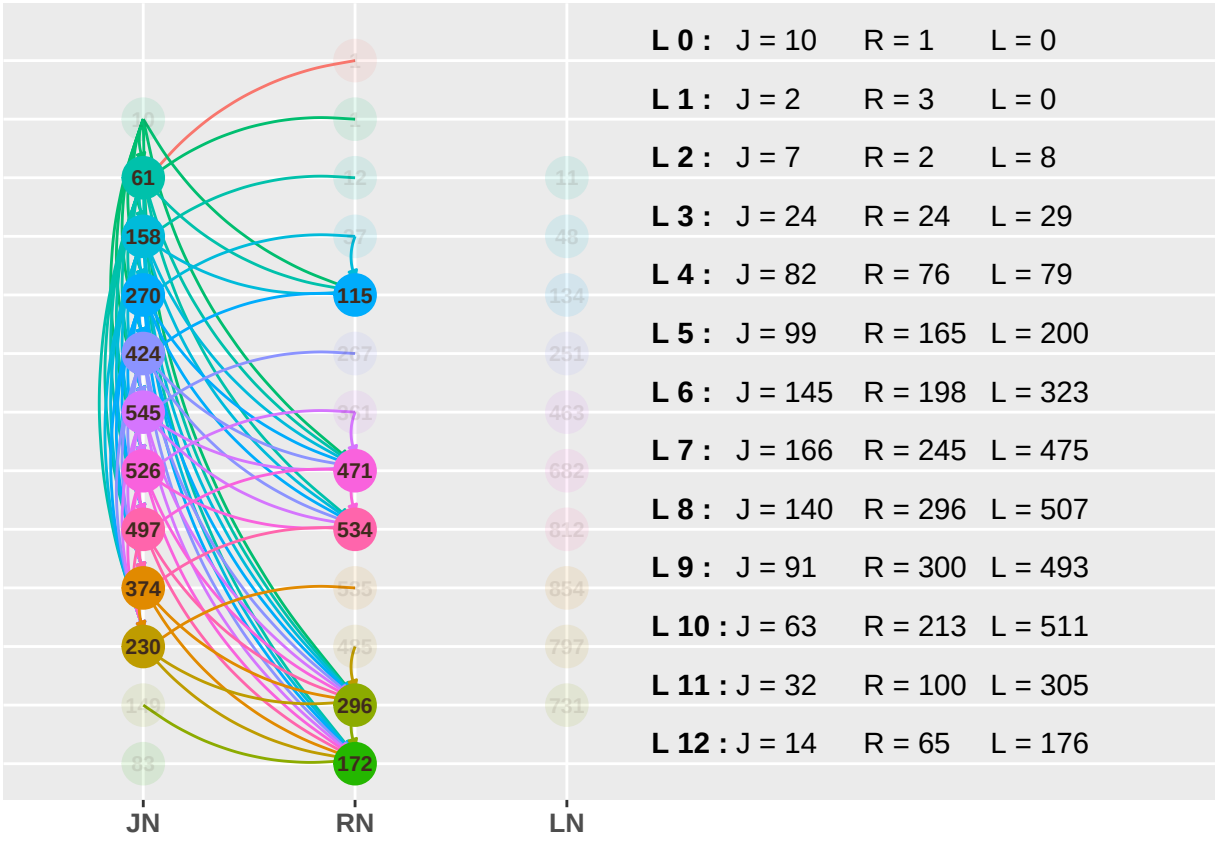

Figure 5: Visualization of a reduced sub-GO-DAG of BPs for Human.

distRankingGO

Given a list of GO-terms, the function provides ranking for the GO-terms according to the distance between the GO-terms hierarchy level and the maximal depth of paths in the GO-DAG passing through these GO-terms. The function provides options for "BP", "MF" and "CC" ontology.

```
Terms <- c("GO:0000278","GO:0006414","GO:0022403","GO:0006415","GO:0006614",
           "GO:0045047","GO:0072599","GO:0006613","GO:0000184","GO:0070972",
           "GO:0006413","GO:0000087","GO:0000280","GO:0000279","GO:0006612",
           "GO:0000956","GO:0048285","GO:0019080","GO:0019083","GO:0043624",
           "GO:0006402","GO:0032984","GO:0006401","GO:0072594","GO:0019058",
           "GO:0051301","GO:0016071","GO:0006412","GO:0002682","GO:0022411",
           "GO:0001775","GO:0046649","GO:0045321","GO:0050776","GO:0007155",
           "GO:0022610","GO:0060337","GO:0071357","GO:0034340","GO:0016032",
           "GO:0044764","GO:0006396","GO:0010564","GO:0002684","GO:0006259",
           "GO:0051249","GO:0045087")

# Ordering of the GO-terms in the list
distRankingGO(goterm = Terms, domain = "BP", plot = TRUE)
```

```
#> $`GO-terms_ranking`
#>  [1] "GO:0022610" "GO:0001775" "GO:0007155" "GO:0002682" "GO:0045321"
#>  [6] "GO:0000278" "GO:0051301" "GO:0046649" "GO:0050776" "GO:0002684"
#> [11] "GO:0048285" "GO:0010564" "GO:0000280" "GO:0044764" "GO:0051249"
#> [16] "GO:0022411" "GO:0016032" "GO:0070972" "GO:0032984" "GO:0006259"
```

```
#> [21] "GO:0043624" "GO:0072594" "GO:0016071" "GO:0006396" "GO:0072599"
#> [26] "GO:0006401" "GO:0006412" "GO:0006413" "GO:0019080" "GO:0006402"
#> [31] "GO:0019058" "GO:0045087" "GO:0019083" "GO:0034340" "GO:0006414"
#> [36] "GO:0022403" "GO:0045047" "GO:0006612" "GO:0000956" "GO:0071357"
#> [41] "GO:0000279" "GO:0060337" "GO:0006415" "GO:0006613" "GO:0000184"
#> [46] "GO:0000087" "GO:0006614"
#>
#> $indices_of_ranking
#>  [1] 36 31 35 29 33  1 26 32 34 44 17 43 13 41 46 30 40 10 22 45 20 24 27 42  7
#> [26] 23 28 11 18 21 25 47 19 39  2  3  6 15 16 38 14 37  4  8  9 12  5
#>
#> $distance
#>  [1] 17 16 16 15 15 14 14 14 14 14 13 13 12 12 12 11 11 10 10 10  9  8  8  8  7
#> [26]  7  7  6  6  6  6  6  5  5  4  4  4  4  4  4  3  3  2  2  2  2  1
#>
#> $plot
```

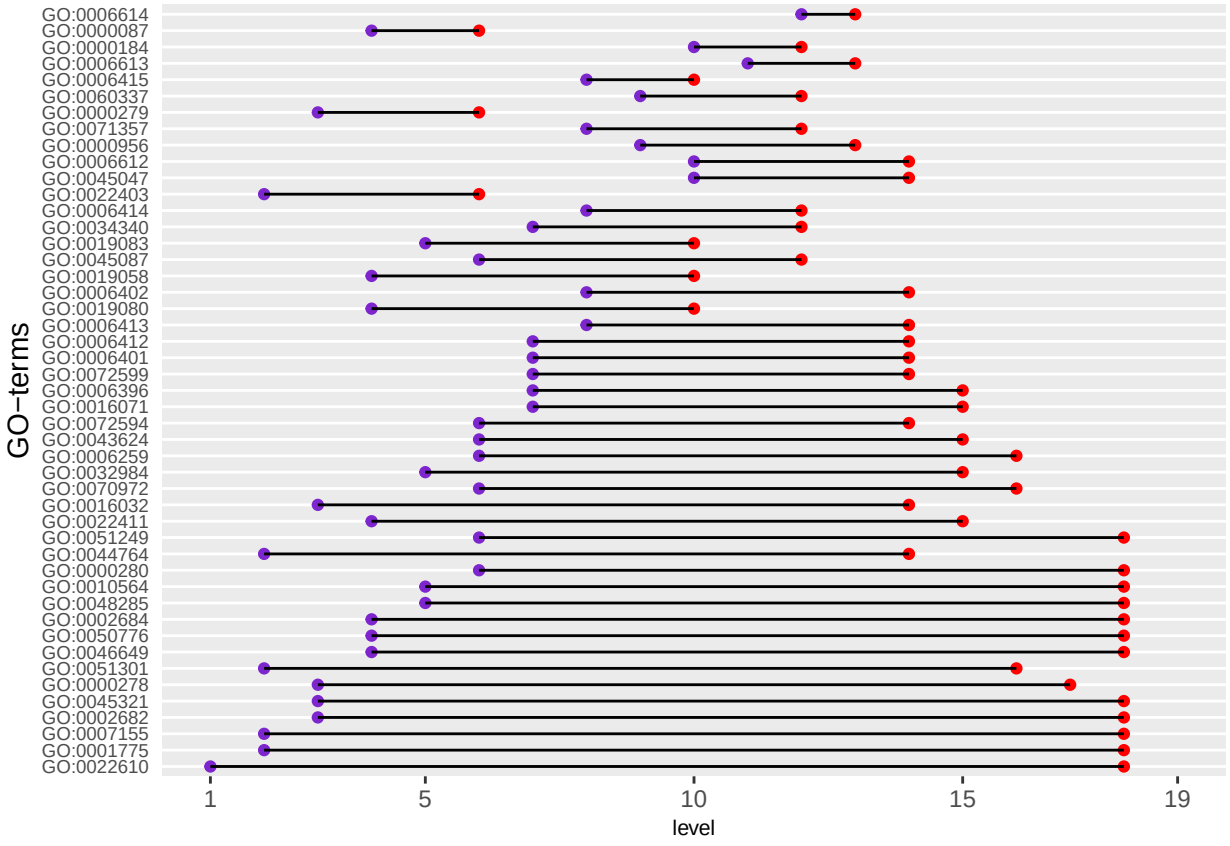

Figure 6: The hierarchy levels for a list of GO-terms (y-axis) are shown in purple and the hierarchy levels for the maximal depth of paths in the GO-DAG passing through these GO-terms is shown in red.

The function produced as output the ranked GO-terms, the indices of the ranking (indices corresponding to the original list), the distance between the GO-terms hierarchy level and the maximal depth of paths in the GO-DAG passing through these GO-terms and a visualisation of the ranking. The GO-terms are ranked according to the distance between the two points (purple and red) shown in Figure 6.

scoreRankingGO

The function scoreRankingGO is similar to the distRankingGO function because both function provide ordering for a given list of GO-terms. The difference is scoreRankingGO rank the GO-terms according to a score which is computed using Equation 1.

$$score = \frac{level(GO)}{level_{max}(GO)} \times \frac{level(GO)}{level_{GO-DAG}(GO)} \tag{1}$$

The function produced as output the ranked GO-terms, the indices of the ranking (indices corresponding to the original list), the scores of each GO-terms and a visualisation of the ranking.

```
Terms <- c("GO:0000278","GO:0006414","GO:0022403","GO:0006415","GO:0006614",
           "GO:0045047","GO:0072599","GO:0006613","GO:0000184","GO:0070972",
           "GO:0006413","GO:0000087","GO:0000280","GO:0000279","GO:0006612",
           "GO:0000956","GO:0048285","GO:0019080","GO:0019083","GO:0043624",
           "GO:0006402","GO:0032984","GO:0006401","GO:0072594","GO:0019058",
           "GO:0051301","GO:0016071","GO:0006412","GO:0002682","GO:0022411",
           "GO:0001775","GO:0046649","GO:0045321","GO:0050776","GO:0007155",
           "GO:0022610","GO:0060337","GO:0071357","GO:0034340","GO:0016032",
           "GO:0044764","GO:0006396","GO:0010564","GO:0002684","GO:0006259",
           "GO:0051249","GO:0045087")

# Ordering of the GO-terms in the list
scoreRankingGO(goterm = Terms, domain = "BP", plot = FALSE)
```

```
#> $GO_terms_ranking
#>  [1] "GO:0022610" "GO:0001775" "GO:0007155" "GO:0051301" "GO:0044764"
#>  [6] "GO:0002682" "GO:0045321" "GO:0000278" "GO:0016032" "GO:0022403"
#> [11] "GO:0046649" "GO:0050776" "GO:0002684" "GO:0022411" "GO:0048285"
#> [16] "GO:0010564" "GO:0000279" "GO:0019080" "GO:0019058" "GO:0032984"
#> [21] "GO:0000280" "GO:0051249" "GO:0070972" "GO:0006259" "GO:0043624"
#> [26] "GO:0019083" "GO:0072594" "GO:0000087" "GO:0045087" "GO:0016071"
#> [31] "GO:0006396" "GO:0072599" "GO:0006401" "GO:0006412" "GO:0034340"
#> [36] "GO:0006413" "GO:0006402" "GO:0006414" "GO:0071357" "GO:0000956"
#> [41] "GO:0006415" "GO:0060337" "GO:0045047" "GO:0006612" "GO:0000184"
#> [46] "GO:0006613" "GO:0006614"
#>
#> $indices_of_ranking
#>  [1] 36 31 35 26 41 29 33  1 40  3 32 34 44 30 17 43 14 18 25 22 13 46 10 45 20
#> [26] 19 24 12 47 27 42  7 23 28 39 11 21  2 38 16  4 37  6 15  9  8  5
#>
#> $score
#>  [1] 0.002923977 0.011695906 0.011695906 0.013157895 0.015037594 0.026315789
#>  [7] 0.026315789 0.027863777 0.033834586 0.035087719 0.046783626 0.046783626
#> [13] 0.046783626 0.056140351 0.073099415 0.073099415 0.078947368 0.084210526
#> [19] 0.084210526 0.087719298 0.105263158 0.105263158 0.118421053 0.118421053
#> [25] 0.126315789 0.131578947 0.135338346 0.140350877 0.157894737 0.171929825
#> [31] 0.171929825 0.184210526 0.184210526 0.184210526 0.214912281 0.240601504
#> [37] 0.240601504 0.280701754 0.280701754 0.327935223 0.336842105 0.355263158
#> [43] 0.375939850 0.375939850 0.438596491 0.489878543 0.582995951
```

prioritizedGOTerms

Given a vector of GO-terms, this function prioritizes the GO-terms by expoiting the structure of a DAG. Starting from the GO-term on the highest level and searching all the paths to the root node iteratively. If the argument "sp" is TRUE, only shortest paths are used, otherwise all paths. If any GO-terms in the input vector are found along this path, these GO-terms are removed. This is because the GO-term at the end of a path is more specific than the GO-terms along the path. For an organism, the GO-terms of that organism are used for the prioritization. If the organism argument is NULL then all the (non-retired) GO-terms from a particular ontology are used in the ranking.

```
Terms <- c("GO:0042254", "GO:0022613", "GO:0034470", "GO:0006364", "GO:0016072",
           "GO:0034660", "GO:0006412", "GO:0006396", "GO:0007005", "GO:0032543",
           "GO:0044085", "GO:0044281", "GO:0044257", "GO:0030163", "GO:0006082",
           "GO:0044248", "GO:0006519", "GO:0009056", "GO:0019752", "GO:0043436")

# We Prioritize the given biological process GO-terms

prioritizedGOTerms(lst = Terms, organism = "Human", sp = TRUE, domain = "BP")
```

```
#> $HF
#> [1] "GO:0006364" "GO:0032543" "GO:0044257" "GO:0007005" "GO:0019752"
#>
#> $rankHF
#> GO:0006364 GO:0032543 GO:0044257 GO:0007005 GO:0019752
#>          8          7          5          4          4
#>
#> $HI
#>  [1] "GO:0042254" "GO:0022613" "GO:0034470" "GO:0006364" "GO:0016072"
#>  [6] "GO:0034660" "GO:0006412" "GO:0006396" "GO:0007005" "GO:0032543"
#> [11] "GO:0044085" "GO:0044257" "GO:0030163" "GO:0006082" "GO:0044248"
#> [16] "GO:0006519" "GO:0019752" "GO:0043436"
#>
#> $rankHI
#> GO:0006364 GO:0034470 GO:0016072 GO:0032543 GO:0034660 GO:0006412 GO:0006396
#>          8          7          7          7          6          6          6
#> GO:0044257 GO:0042254 GO:0007005 GO:0030163 GO:0019752 GO:0022613 GO:0043436
#>          5          4          4          4          4          3          3
#> GO:0044085 GO:0006082 GO:0044248
#>          2          2          2
#>
#> attr(,"class")
#> [1] "GOxploreR" "GPrior"
```

GO4Organism

This function gives all the GO-terms association with an organism and their corresponding GO-levels.

```
# All the biological process gene association GO-terms for Human and their GO-level
head(GO4Organism(organism = "Human", domain = "BP"))
```

```
#>        GO ID Level
#> 1 GO:0008150     0
#> 2 GO:0043312     9
#> 3 GO:0002576     8
#> 4 GO:0006805     5
#> 5 GO:0009168    10
```

```
#> 6 GO:0007155     2
```

```
# All the molecular function gene association GO-terms for Mouse and their GO-level
head(GO4Organism(organism = "Mouse", domain = "MF"))
```

```
#>        GO ID Level
#> 1 GO:0005524     8
#> 2 GO:0004672     5
#> 3 GO:0004679     7
#> 4 GO:0016740     2
#> 5 GO:0000166     4
#> 6 GO:0016301     4
```

```
# All the cellular component gene association GO-terms for Rat and their GO-level
head(GO4Organism(organism = "Rat", domain = "CC"))
```

```
#>        GO ID Level
#> 1 GO:0016020     2
#> 2 GO:0016021     4
#> 3 GO:0005654     7
#> 4 GO:0005737     2
#> 5 GO:0005886     3
#> 6 GO:0005887     5
```

Conclusion

This vignette gave a brief overview of the functionality provided by the GOxploreR package. We showed all functions and how to use them.

References

1. Gene Ontology Consortium. 2018. The gene ontology resource: 20 years and still going strong. *Nucleic acids research* 47, D1: D330–D338.

2. Kalifa Manjang, Shailesh Tripathi, Olli Yli-Harja, Matthias Dehmer, and Frank Emmert-Streib. 2020. Graph-based exploitation of gene ontology using goxplorer for scrutinizing biological significance. *Scientific Reports* 10, 1: 16672. https://doi.org/10.1038/s41598-020-73326-3