Introduction to the GOxploreR package

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

24 March 2022

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

Citation
Contents

Citation .......................................................... 1
Introduction .......................................................... 3
Overview of the functionality of the package ..................... 3
  Gene2GOTermAndLevel ........................................... 3
  Gene2GOTermAndLevel_ON ....................................... 5
  GOTermXXOnLevel ............................................... 5
  Level2GOTermXX ................................................... 6
  Level2LeafNodeXX ................................................ 6
  Level2JumpNodeXX ............................................... 7
  Level2RegularNodeXX ............................................. 7
  Level2NoLeafNodeXX ............................................. 7
  getGOCategory .................................................. 8
  degreeDistXX ..................................................... 8
  GOTermXX2ChildLevel .......................................... 9
  GetLeafNodesXX .................................................. 10
  GO2DecXX ......................................................... 10
  GetDAG .......................................................... 10
  visRDAGXX ....................................................... 11
  visRsubDAGXX .................................................... 14
  distRankingGO .................................................. 15
  scoreRankingGO ................................................ 17
  prioritizedGOTerms ............................................. 18
  GO4Organism .................................................... 18
  Conclusion ...................................................... 19
  References ...................................................... 19

List of Figures

1 Degree distribution of the biological process GO-terms on level 4. ............... 8
2 Degree distribution of the molecular function GO-terms on level 2. ................. 9
3 Degree distribution of the cellular component GO-terms on level 10. ............... 9
4 Visualization of a reduced GO-DAG for Caenorhabditis elegans. ................. 14
5 Visualization of a reduced sub-GO-DAG of BPs for Human. ..................... 15
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. 16

List of Tables

1 Organisms supported by the GOxploreR package. .............................. 3
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.

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

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:

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

The package function can be loaded using:

```r
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.

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

<table>
<thead>
<tr>
<th>Entrezgene ID</th>
<th>GO ID</th>
<th>Domain</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10212 GO:0005634</td>
<td>CC</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>10212 GO:0005737</td>
<td>CC</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>10212 GO:0016607</td>
<td>CC</td>
<td>9</td>
</tr>
</tbody>
</table>
The biological process gene ontology terms will be retrieved and their levels:

```r
Gene2GOTermAndLevel(genes = c(100000642, 30592, 58153, 794484), organism = "Danio rerio")
```

<table>
<thead>
<tr>
<th>Entrezgene ID</th>
<th>GO ID</th>
<th>Domain</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>100000642</td>
<td>GO:0007186</td>
<td>BP</td>
<td>5</td>
</tr>
<tr>
<td>100000642</td>
<td>GO:0050911</td>
<td>BP</td>
<td>7</td>
</tr>
<tr>
<td>30592</td>
<td>GO:0045214</td>
<td>BP</td>
<td>9</td>
</tr>
<tr>
<td>30592</td>
<td>GO:0060047</td>
<td>BP</td>
<td>5</td>
</tr>
<tr>
<td>30592</td>
<td>GO:0060038</td>
<td>BP</td>
<td>9</td>
</tr>
<tr>
<td>30592</td>
<td>GO:0048738</td>
<td>BP</td>
<td>7</td>
</tr>
<tr>
<td>30592</td>
<td>GO:0055005</td>
<td>BP</td>
<td>11</td>
</tr>
<tr>
<td>30592</td>
<td>GO:0055015</td>
<td>BP</td>
<td>10</td>
</tr>
<tr>
<td>30592</td>
<td>GO:0045823</td>
<td>BP</td>
<td>7</td>
</tr>
<tr>
<td>794484</td>
<td>GO:0008150</td>
<td>BP</td>
<td>0</td>
</tr>
</tbody>
</table>
```

The molecular function gene ontology terms will be retrieved and their levels:

```r
Gene2GOTermAndLevel(genes = c(100009600, 18131, 100017), organism = "Mouse", domain = "MF")
```

<table>
<thead>
<tr>
<th>Entrezgene ID</th>
<th>GO ID</th>
<th>Domain</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>100009600</td>
<td>GO:0008270</td>
<td>MF</td>
<td>6</td>
</tr>
<tr>
<td>100009600</td>
<td>GO:0043565</td>
<td>MF</td>
<td>5</td>
</tr>
<tr>
<td>100009600</td>
<td>GO:0046872</td>
<td>MF</td>
<td>4</td>
</tr>
<tr>
<td>100009600</td>
<td>GO:0003677</td>
<td>MF</td>
<td>4</td>
</tr>
<tr>
<td>18131</td>
<td>GO:0005515</td>
<td>MF</td>
<td>2</td>
</tr>
<tr>
<td>18131</td>
<td>GO:0005509</td>
<td>MF</td>
<td>5</td>
</tr>
<tr>
<td>18131</td>
<td>GO:0038023</td>
<td>MF</td>
<td>2</td>
</tr>
<tr>
<td>18131</td>
<td>GO:0042802</td>
<td>MF</td>
<td>3</td>
</tr>
<tr>
<td>18131</td>
<td>GO:0019899</td>
<td>MF</td>
<td>3</td>
</tr>
<tr>
<td>100017</td>
<td>GO:0005515</td>
<td>MF</td>
<td>2</td>
</tr>
<tr>
<td>100017</td>
<td>GO:0035650</td>
<td>MF</td>
<td>3</td>
</tr>
<tr>
<td>100017</td>
<td>GO:0001784</td>
<td>MF</td>
<td>5</td>
</tr>
<tr>
<td>100017</td>
<td>GO:0005102</td>
<td>MF</td>
<td>3</td>
</tr>
<tr>
<td>100017</td>
<td>GO:0005546</td>
<td>MF</td>
<td>7</td>
</tr>
<tr>
<td>100017</td>
<td>GO:0001540</td>
<td>MF</td>
<td>4</td>
</tr>
<tr>
<td>100017</td>
<td>GO:0050750</td>
<td>MF</td>
<td>5</td>
</tr>
<tr>
<td>100017</td>
<td>GO:0030159</td>
<td>MF</td>
<td>4</td>
</tr>
<tr>
<td>100017</td>
<td>GO:0030276</td>
<td>MF</td>
<td>3</td>
</tr>
<tr>
<td>100017</td>
<td>GO:0035612</td>
<td>MF</td>
<td>3</td>
</tr>
<tr>
<td>100017</td>
<td>GO:0035591</td>
<td>MF</td>
<td>3</td>
</tr>
<tr>
<td>100017</td>
<td>GO:0035615</td>
<td>MF</td>
<td>4</td>
</tr>
</tbody>
</table>
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
GOTermBPOnLevel(goterm = goterms)

## Term Level
## 1 GO:0009083 5
## 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
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
GOTermCCOnLevel(goterm = goterms)

## Term Level
## 1 GO:0055044 2
## 2 GO:0030427 2
## 3 GO:0036436 10
## 4 GO:0034980 7
## 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

```r
Level2GOTermBP(level = 1, organism = "Human")
```

```r
#> [1] "GO:0008152" "GO:0032502" "GO:0002376" "GO:0048511" "GO:0043473"
#> [6] "GO:0040011" "GO:0023052" "GO:009987" "GO:0000003" "GO:0007610"
#> [11] "GO:0050896"
```

# Retrieve all the GO-terms from a particular GO MF level

```r
Level2GOTermMF(level = 14, organism = "Rat")
```

```r
#> [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

```r
Level2GOTermCC(level = 14)
```

```r
#> [1] "GO:0044203"
```

**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

```r
Level2LeafNodeBP(level = 2, organism = "Danio rerio")
```

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

# Get all leaf nodes from a GO MF level

```r
Level2LeafNodeMF(level = 12)
```

```r
#> [1] "GO:0015375" "GO:0099635" "GO:1905054" "GO:1905058" "GO:0098695"
#> [6] "GO:0098697" "GO:0005314"
```

# Get all leaf nodes from a GO CC level

```r
Level2LeafNodeCC(level = 10, organism = "Schizosaccharomyces pombe")
```

```r
#> [1] "GO:1902377" "GO:1990342" "GO:0071957" "GO:0031942" "GO:0000124"
#> [6] "GO:0005662" "GO:0016586" "GO:0031618" "GO:0034991" "GO:0035751"
#> [11] "GO:1990707" "GO:0016514" "GO:0030126" "GO:0032221" "GO:0005671"
#> [16] "GO:0005749" "GO:0046695" "GO:0005658" "GO:0061499" "GO:0030875"
#> [21] "GO:0036266" "GO:0031262" "GO:0120104" "GO:0071339" "GO:0061496"
#> [26] "GO:0050331" "GO:0030998" "GO:1904834" "GO:0030958" "GO:0120105"
#> [31] "GO:0033698" "GO:0034990" "GO:0000136" "GO:0030127" "GO:0005742"
#> [36] "GO:0043599" "GO:0071627" "GO:0071958" "GO:0061493" "GO:0061497"
#> [41] "GO:0043625" "GO:0031307" "GO:0070692" "GO:0070985" "GO:1990574"
#> [46] "GO:0043505" "GO:0070691" "GO:1990612" "GO:0033551" "GO:0034044"
#> [51] "GO:0120106" "GO:0032865" "GO:0032585" "GO:0005750" "GO:1990537"
```
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"))

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

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

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 substituted with BP, MF, or CC.

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

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

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

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")
All GO-terms on a particular GO MF level that are not leaf nodes
Level2NoLeafNodeMF(level = 10, organism = "Caenorhabditis elegans")

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

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)

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)

![Degree distribution of the molecular function GO-terms on level 2.](image)

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)

![Degree distribution of the cellular component GO-terms on level 10.](image)

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")

```r
# $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")

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

# Get the level of a GO CC term's children

GOTermCC2ChildLevel(goterm = "GO:0071735")

```r
#> $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 descendant 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")

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

# Molecular function GO-term descendant terms

GO2DecMF(goterm = "GO:0008553")

```r
#> [1] "GO:0008900"
```

# Cellular component GO-term descendant terms

GO2DecCC(goterm = "GO:0031233")

```r
#> [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"))
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))

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


# 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.

```

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

```
# [1] "GO:0022610" "GO:0001775" "GO:0007155" "GO:0002682" "GO:0045321" 
# [6] "GO:0046649" "GO:0050776" "GO:0002684" "GO:0000278" "GO:0051301" 
# [16] "GO:0022411" "GO:0016032" "GO:0032984" "GO:0006259" "GO:0070972"
```
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 functions provide ordering for a given list of GO-terms. The difference is scoreRankingGO ranks the GO-terms according to a score which is computed using Equation 1.

\[
\text{score} = \frac{\text{level}(GO)}{\text{level}_{\text{max}}(GO)} \times \frac{\text{level}(GO)}{\text{level}_{\text{GO-DAG}}(GO)}
\]  

Equation 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-term and a visualisation of the ranking.

```r

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

```r
## $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"
## [16] "GO:0010564" "GO:0000279" "GO:0019080" "GO:0019058" "GO:0032984"
## [21] "GO:0051249" "GO:0000280" "GO:0006259" "GO:0070972" "GO:0043624"
## [26] "GO:0072594" "GO:0000087" "GO:0045087" "GO:0019083" "GO:0072599"
## [31] "GO:0016071" "GO:0006396" "GO:0006401" "GO:0006412" "GO:0034340"
## [36] "GO:0006413" "GO:0006402" "GO:0006415" "GO:0060337" "GO:0045047" "GO:0000184"
## [41] "GO:0006612" "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 24 46 13 45 10 20
## [26] 24 12 47 19 7 27 42 23 28 39 11 21 2 38 16 4 37 6 15 9 8 5

## $score
## [1] 0.002770083 0.011080332 0.011080332 0.013157895 0.015037594 0.024930748
## [7] 0.044321330 0.044321330 0.056140351 0.073099415 0.073099415 0.078947368 0.084210526
## [13] 0.084210526 0.087719298 0.099722992 0.105263158 0.118421053 0.126315789 0.126315789
## [19] 0.126315789 0.140350877 0.157894737 0.164473684 0.171929825
## [25] 0.171929825 0.171929825 0.184210526 0.184210526 0.214912281 0.240601504
## [31] 0.240601504 0.280701754 0.280701754 0.327935223 0.336842105 0.355263158
## [37] 0.424561404 0.424561404 0.438596491 0.541353383 0.635338346
```
prioritizedGOTerms

Given a vector of GO-terms, this function prioritizes the GO-terms by exploiting 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.

```r

# We Prioritize the given biological process GO-terms
prioritizedGOTerms(lst = Terms, organism = "Human", sp = TRUE, domain = "BP")
```

```r
#> $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"
#> [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.

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

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

18
```r
# 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     3
#>  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