`sims`: an `R` package for Computing Semantic Similarities

Jose Luis Mosquera and Alex Sánchez

August 31, 2014

Contents

1 Introduction

An ontology is a way for annotating concepts of a certain domain. It allows the comparison between entities through their associated concepts, and which otherwise would not be comparable. The structure of the vocabulary of an ontology is arranged as a rooted directed acyclic graph (DAG). That is, an ontology is a hierarchy with a single "highest" term called the *root*. All other descendant terms are connected by either one or a several directed links (i.e. the links point upwards) to the root, an these links are acyclic (i.e. cycles are not allowed in the graph).

One of the most successful ontologies for annotating biological vocabularies is the Gene Ontology (GO). It is an annotation resource created and maintained by a public consortium, http://geneontology.org/page/go-consortium-contributors-list [18]. The main goal of the consortium is *citing their mission,*

to produce a controlled vocabulary that can be applied to all organisms even as knowledge of gene and protein roles in cells is accumulating and changing. It is organized covering three domains: *Cellular Component* (*CC*), *Biological Process* (*BP*), and *Molecular Function* (*MF*). Each ontology domain consists of a high number of terms or categories hierarchically related from least (top) to most (bottom) specialized characteristics.The GO has two types of relationship (i.e. links)between GO terms: the *is-a* and the *part-of*.

Usually, an ontology is used for the interpretation of sets of objects mapped to this ontology. For example, the GO allows annotating genes and their products. Most genes are annotated in one or more GO terms. Annotations are made as specific as possible. As a consequence a gene is associated not only with its annotations but also with all the less specific terms associated with them. Furthermore, a given gene product may represent one or more molecular functions, be used in one or more biological processes and appear in one or more cellular components.

Many applications using ontologies require to determine the relationship between pairs of terms [11, 12]. An appropriate measure of such relationship is the semantic similarity between the terms. Generally speaking, a semantic similarity between two terms is as a function of distance between the terms in the graph corresponding to the underlying ontology [3]. There are different methods a approaches [4], but mainly they are classified into (1) methods based on node-based approaches, (2) methods based on edge-based approaches, and (3) methods based on hybrid-based approaches.

`sims` package provides functions for dealing with arbitrary ontologies, computing semantic similarities between their and comparing lists of objects annotated in these ontologies, particularly focused on the GO.

The present present document is just an introduction to the use of `sims` package.

To start with `sims` package, write the following code

```
library("sims")
help("sims")
```

Functions available in the package are

```
ls("package:sims")
```

```
##  [1] "ancestors"        "commonAncestors"
##  [3] "cosSim"           "depth"
##  [5] "distRada"         "getA"
##  [7] "getGk"            "getGr"
##  [9] "GOANCESTORS"      "goOOC"
## [11] "GOPARENTS"        "gosims"
## [13] "gosimsAvsB"       "gosimsProfiles"
## [15] "ICA"              "inverseIminusG"
## [17] "is.OOC"           "LCAs"
```

```
## [19] "mapEG2GO"          "mappingMatrix"
## [21] "Nt"                "pdHap"
## [23] "pdHax"             "pdHm"
## [25] "pdHx"              "plotGODAG"
## [27] "plotHistSims"      "pseudoDists"
## [29] "refinementMatrix" "resnikSummary"
## [31] "simFaith"          "simJC"
## [33] "simLin"            "simNunivers"
## [35] "simPsec"           "simRada"
## [37] "simRel"            "simRes"
## [39] "simRes.eb"         "simsBetweenGOIDs"
## [41] "sims.eb"           "simsMat"
## [43] "sims.nb"           "summaryMICA"
## [45] "summaryPaths"      "summarySims"
## [47] "summarySimsAvsB"  "termPairs"
## [49] "toMat"             "toOOC"
## [51] "toPairs"
```

2 Semantic Similarities Between Terms of an Arbitrary Ontology Mapped by a List of Objects

To illustrate the usage of basic structures and the computation of semantic similarities between terms of an arbitrary ontology with `sims` package, we are going to make use of an example proposed by Joslyn *et al* [6]. It consists of a 10 object identifiers mapping to terms of an ontology with 12 concepts. Figure 1 shows a representation of the example considered.

In order to deal with the structure we make use of a concept called *Object-Ontology Complex* (*OOC*) introduced by Carey [1], that we will see in next section 2.1. But, previously we need to "translate" the graph structure described above in terms of matrices.

For the inpatient user, load the following dataset into memory in order to compute semantic similarities and goes to subsection 2.2

```
data(joslyn)
help("joslyn")
```

Otherwise, next coding lines provides the process for building the matrix forms associated with the each component of the structure presented above

```
## 1. Vocabulary of the ontology
vocabulary <- c("R", "B", "C", "K", "F", "G", "I", "E", "J", "H",
"A", "D")
```

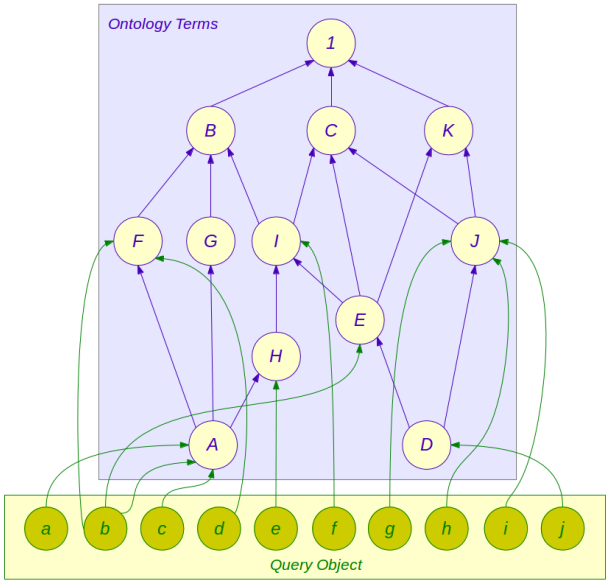


Figure 1: Representation of an ontology with 12 terms and 10 object identifiers annotated in the ontology

```
## 2. Links between terms (structure of the ontology)
origin <- c("B", "C", "K", "F", "G", "I", "I", "E", "J", "E", "J",
"A", "A", "E", "H", "D", "D", "A")
terminus <- c("R", "R", "R", "B", "B", "B", "C", "C", "C", "K",
"K", "F", "G", "I", "I", "E", "J", "H")
links <- data.frame(origin, terminus)
mat.g <- toMat(df = links, rnames = vocabulary,
cnames = vocabulary)

print(mat.g)

##   R B C K F G I E J H A D
## R 0 0 0 0 0 0 0 0 0 0 0 0
## B 1 0 0 0 0 0 0 0 0 0 0 0
## C 1 0 0 0 0 0 0 0 0 0 0 0
## K 1 0 0 0 0 0 0 0 0 0 0 0
## F 0 1 0 0 0 0 0 0 0 0 0 0
## G 0 1 0 0 0 0 0 0 0 0 0 0
## I 0 1 1 0 0 0 0 0 0 0 0 0
## E 0 0 1 1 0 0 1 0 0 0 0 0
## J 0 0 1 1 0 0 0 0 0 0 0 0
## H 0 0 0 0 0 0 1 0 0 0 0 0
```

```
## A 0 0 0 0 1 1 0 0 0 1 0 0
## D 0 0 0 0 0 0 0 1 1 0 0 0

## 3. Objects identifiers that are annotates in the ontology
object.ids <- letters[1:10]

## 4. Mapping from objects to terms (annotation of objects)
object <- c("b", "d", "f", "b", "g", "h", "i", "e", "a", "b", "c",
"j")
term <- c("F", "F", "I", "E", "J", "J", "J", "H", "A", "A", "A",
"D")
map <- data.frame(object, term)
mat.m <- toMat(df = map, rnames = object.ids, cnames = vocabulary)

print(mat.m)

##   R B C K F G I E J H A D
## a 0 0 0 0 0 0 0 0 0 0 1 0
## b 0 0 0 0 1 0 0 1 0 0 1 0
## c 0 0 0 0 0 0 0 0 0 0 1 0
## d 0 0 0 0 1 0 0 0 0 0 0 0
## e 0 0 0 0 0 0 0 0 0 1 0 0
## f 0 0 0 0 0 0 1 0 0 0 0 0
## g 0 0 0 0 0 0 0 0 1 0 0 0
## h 0 0 0 0 0 0 0 0 1 0 0 0
## i 0 0 0 0 0 0 0 0 1 0 0 0
## j 0 0 0 0 0 0 0 0 0 0 0 1
```

2.1 Object-Ontology Complex (OOC) Container

An OOC is *a formalism for working with ontologies for statistical purposes*. It combines the four elements described in previous section 2. That is, (1) the terms of the ontology, (2) the structure of the directed acyclic graph (DAG), (3) the list of objects annotated in the ontology, and (4) how the objects map to the terms.

`sims` package has a class `OOC`, that is used as a general container for Object-Ontology Complexes (OOC).

```
help("OOC")
```

The function `toOOC` facilitates the construction of an object of class `OOC`. This object is merely used as a container of the elements of the OOC. It has four slots `T` (the list of terms or vocabulary of the ontology), `G` (the matrix accessibility matrix or the matrix of 1-step refinement associated with DAF structure of the ontology), `O` (the list of object identifiers), and `M` (the mapping matrix between objects and terms).

```
joslyn.OOC <- toOOC(T = vocabulary, G = mat.g, O = object.ids,
                    M = mat.m)
print(joslyn.OOC)
```

```
## An object of class "OOC"
## Slot "T":
##  [1] "R" "B" "C" "K" "F" "G" "I" "E" "J" "H" "A" "D"
##
## Slot "G":
##   R B C K F G I E J H A D
## R 0 0 0 0 0 0 0 0 0 0 0 0
## B 1 0 0 0 0 0 0 0 0 0 0 0
## C 1 0 0 0 0 0 0 0 0 0 0 0
## K 1 0 0 0 0 0 0 0 0 0 0 0
## F 0 1 0 0 0 0 0 0 0 0 0 0
## G 0 1 0 0 0 0 0 0 0 0 0 0
## I 0 1 1 0 0 0 0 0 0 0 0 0
## E 0 0 1 1 0 0 1 0 0 0 0 0
## J 0 0 1 1 0 0 0 0 0 0 0 0
## H 0 0 0 0 0 0 1 0 0 0 0 0
## A 0 0 0 0 1 1 0 0 0 1 0 0
## D 0 0 0 0 0 0 0 1 1 0 0 0
##
## Slot "O":
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"
##
## Slot "M":
##   R B C K F G I E J H A D
## a 0 0 0 0 0 0 0 0 0 0 1 0
## b 0 0 0 0 1 0 0 1 0 0 1 0
## c 0 0 0 0 0 0 0 0 0 0 1 0
## d 0 0 0 0 1 0 0 0 0 0 0 0
## e 0 0 0 0 0 0 0 0 0 1 0 0
## f 0 0 0 0 0 0 1 0 0 0 0 0
## g 0 0 0 0 0 0 0 0 1 0 0 0
## h 0 0 0 0 0 0 0 0 1 0 0 0
## i 0 0 0 0 0 0 0 0 1 0 0 0
## j 0 0 0 0 0 0 0 0 0 0 0 1
```

2.2 Comptutation of Semantic similarities

In `sims` package there are implemented a total of fourteen measures from different approaches. The following subsections describe main functions to compute semantic similarities between all the pairs of terms of the induced graph (from the ontology) by a list of object identifiers.

2.2.1 Methods of Node-Based Approach

There are implemented seven semantic similarity measures proposed by Resnik [13], Lin [7], Schlicker *et al.* [15], Jiang and Conrath [5], Mazandu and Mulder [9], Pirró and Seco [11], and Pirró and Euzenat [10]. All the methods are based on the concept of *Information Content* (*IC*) proposed by Resnik [13], and the shared information between the two terms being measured is proportional to the IC of the Most Informative Common Ancestor (MICA) in the rooted DAG.

Semantic similarities measures of node-based approach are computed by calling the wrapper function `sims`.

```
help("sims")
```

Three arguments are required by this function: a `list` with the ancestors of each selected term (`at`), a `numeric` vector with the IC of each term (`ic`), and the `method` required (see possibilities in the help).

To obtain the list of ancestors we need to build the accessibility matrix associated with the DAG structure by performing the following computation

```
## Accessibility matrix
inv.IminusG <- inverseIminusG(joslyn.OOC)
A.mat <- getA(inv.IminusG)
print(A.mat)

##       R     B     C     K     F     G     I     E     J
## R FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## B  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## C  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## K  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## F  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## G  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## I  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## E  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE
## J  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
## H  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
## A  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE
## D  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE
##       H     A     D
## R FALSE FALSE FALSE
## B FALSE FALSE FALSE
## C FALSE FALSE FALSE
## K FALSE FALSE FALSE
## F FALSE FALSE FALSE
## G FALSE FALSE FALSE
## I FALSE FALSE FALSE
## E FALSE FALSE FALSE
```

```
## J FALSE FALSE FALSE
## H FALSE FALSE FALSE
## A  TRUE FALSE FALSE
## D FALSE FALSE FALSE

## Ancestors
at <- ancestors(A.mat)
print(at)

## $R
## [1] "R"
##
## $B
## [1] "R" "B"
##
## $C
## [1] "R" "C"
##
## $K
## [1] "R" "K"
##
## $F
## [1] "R" "B" "F"
##
## $G
## [1] "R" "B" "G"
##
## $I
## [1] "R" "B" "C" "I"
##
## $E
## [1] "R" "B" "C" "K" "I" "E"
##
## $J
## [1] "R" "C" "K" "J"
##
## $H
## [1] "R" "B" "C" "I" "H"
##
## $A
## [1] "R" "B" "C" "F" "G" "I" "H" "A"
##
## $D
## [1] "R" "B" "C" "K" "I" "E" "J" "D"
```

Function `resnikSummary` builds a `data.frame` providing the number of times

that each term or any of its refinements appears in the OOC (i.e. $n(t_i)$), the probability of finding the term (i.e. $p(t_i)$), and the Information Content of the term (i.e. $IC(t_i)$). Thus, we can calculate the IC's of each term very easily by performing

```
resnik.sum <- resnikSummary(x = joslyn.OOC)
print(resnik.sum)

##   nt      pt     ic
## R 34 1.00000 0.0000
## B 15 0.44118 0.8183
## C 13 0.38235 0.9614
## K  6 0.17647 1.7346
## F  5 0.14706 1.9169
## G  3 0.08824 2.4277
## I  7 0.20588 1.5805
## E  2 0.05882 2.8332
## J  4 0.11765 2.1401
## H  4 0.11765 2.1401
## A  3 0.08824 2.4277
## D  1 0.02941 3.5264

ic <- resnik.sum[, "ic"]
```

Finally, to compute the semantic similarity we just only indicate the method required

```
## Computation of semantic similarity of Resnik
## (node-based approach)
sims.Res <- sims.nb(at, ic, method = "Res")
head(sims.Res)

##     Resnik
## B-R      0
## C-R      0
## K-R      0
## F-R      0
## G-R      0
## I-R      0

## Computation of all semantic similarities of
## edge-based approach
sims.all <- sims.nb(at, ic, method = "all")
head(sims.all)

##     Resnik Lin Rel     JC Nunivers    Psec Faith
## B-R      0   0   0 0.5500        0 -0.8183     0
```

```
## C-R      0   0   0 0.5098         0 -0.9614     0
## K-R      0   0   0 0.3657         0 -1.7346     0
## F-R      0   0   0 0.3428         0 -1.9169     0
## G-R      0   0   0 0.2917         0 -2.4277     0
## I-R      0   0   0 0.3875         0 -1.5805     0
```

The following function provides a summary of the measures

```
summarySims(sims.all)

##           n NAs     Min Num.Min    Max Num.Max    Mean
## Resnik   66   0  0.0000      24 2.8332       1  0.7998
## Lin      66   0  0.0000      24 1.0000       1  0.3723
## Rel      66   0  0.0000      24 0.9118       1  0.2661
## JC       66   0  0.1796       1 1.0000       1  0.3662
## Nunivers 66   0  0.0000      24 0.8034       1  0.2268
## Psec     66   0 -4.5678       1 2.4277       1 -1.3518
## Faith    66   0  0.0000      24 1.0000       1  0.2804
##          Std.Dev  Median
## Resnik    0.7399  0.8183
## Lin       0.3210  0.4146
## Rel       0.2609  0.2393
## JC        0.1582  0.3271
## Nunivers  0.2098  0.2321
## Psec      1.6449 -1.4746
## Faith     0.2707  0.2616
```

2.2.2 Methods of Edge-Based Approach

With regard to the edge-based approach there are implemented two semantic similarity measures proposed by Resnik [13] and Rada *et al.* [12]. But also, it is a distance measure proposed by Rada [12], and four pseudo-distances proposed by Joslyn *et al* [6].

Semantic similarities measures of edge-based approach are computed by calling the wrapper function `sims.eb`.

```
help("sims.eb")
```

This function depends on four arguments: the `OOC` object (`x`), the name of the root term of the ontology (`root`), the `list` of the ancestors of each selected term (`at`), and the `method` required (see possibilities in the help).

```
## Computation of semantic similarity of Resnik
## (edge-based approach)
Resnik.eb <- sims.eb(x = joslyn.OOC, root = "R", at,
```

```
                      method = "Rada")
head(Resnik.eb)

##       Rada
## B-R 0.5000
## C-R 0.5000
## K-R 0.5000
## F-R 0.3333
## G-R 0.3333
## I-R 0.3333

## Computation of all semantic similarities of
## edge-based approach
sims.eb.all <- sims.eb(x = joslyn.OOC, root = "R", at,
                       method = "all")
head(sims.eb.all)

##       Rada Resnik.eb
## B-R 0.5000         7
## C-R 0.5000         7
## K-R 0.5000         7
## F-R 0.3333         6
## G-R 0.3333         6
## I-R 0.3333         6

## Summary of semantic similarities
summarySims(sims.eb.all)

##            n NAs Min Num.Min Max Num.Max   Mean Std.Dev
## Rada      66   0 0.2       8 0.5      18 0.3399  0.1084
## Resnik.eb 66   0 4.0       8 7.0      18 5.7576  0.9932
##           Median
## Rada      0.3333
## Resnik.eb 6.0000
```

Distance measure of Rada can be computed by calling the function **distRada**.

```
help("distRada")
```

The function requires a **list** of **numeric** vectors with the lengths (in terms of depth) of the number of paths between each pair of terms (**sum.paths**), and the **list** of the ancestors of each selected term (**at**). To obtain the first argument we make use of the function **summaryPaths**

```
sum.paths <- summaryPaths(x = joslyn.OOC, root = "R", len = TRUE)
head(sum.paths, 10)
```

```
##     [,1] [,2] [,3] [,4]
## R-R    0    0    0    0
## B-R    1    0    0    0
## C-R    1    0    0    0
## K-R    1    0    0    0
## F-R    0    2    0    0
## G-R    0    2    0    0
## I-R    0    2    0    0
## E-R    0    2    3    0
## J-R    0    2    0    0
## H-R    0    0    3    0
```

Then, distance is calculated by

```
Rada <- distRada(sum.paths, at)
head(Rada)

##     sp.Rada
## B-R       1
## C-R       1
## K-R       1
## F-R       2
## G-R       2
## I-R       2

summarySims(Rada)

##          n NAs Min Num.Min Max Num.Max  Mean Std.Dev Median
## sp.Rada 66   0   1      18   4       8 2.242  0.9932      2
```

Pseudo-distances implemented in `sims` package can be computed by calling the function `pseudoDists`.

```
help("pseudoDists")
```

This function needs to be fed with the `OOC` object (`x`), the name of the root term of the ontology (`root`), and the `method` required (see possibilities in the help).

```
## Computation of the pseudo-distance of
## the minimum chain length
pd.hm <- pseudoDists(x = joslyn.OOC, root = "R", method = "hm")
head(pd.hm)

##     h.m
## B-R   1
```

```
## C-R   1
## K-R   1
## F-R   2
## G-R   2
## I-R   2

## Computation of all pseudo-distance
pd.all <- pseudoDists(x = joslyn.OOC, root = "R", method = "all")
head(pd.all)

##     h.m h.x h.ax h.ap
## B-R   1   1    1    1
## C-R   1   1    1    1
## K-R   1   1    1    1
## F-R   2   2    2    2
## G-R   2   2    2    2
## I-R   2   2    2    2

## Summary of pseudo-distances
summarySims(pd.all)

##       n NAs Min Num.Min Max Num.Max  Mean Std.Dev Median
## h.m  66  30   1      18 3.0       5 1.639  0.7232   1.50
## h.x  66  30   1      17 4.0       2 1.806  0.9202   2.00
## h.ax 66  30   1      17 3.5       2 1.722  0.8057   1.75
## h.ap 66  30   1      17 3.5       2 1.722  0.8057   1.75
```

3 Semantic Similarities Associated with the GO

The package can manage any ontology, but it is especially focused on the Gene Ontology. In this regard, there are some functions that are particularly adapted for allow building the refinements matrix (i.e. the accessibility matrix) and the mapping matrix (i.e the matrix that maps from Entrez Gene IDs to GO IDs), performing comparisons between lists of semantic similarities, and yield different types of plots (e.g. histograms, diagram bars and DAG's of the induced graphs). Moreover, `sims` package can manage Entrez Gene IDs and GO IDs from any `R` organism package.

In order to explore and compare semantic similarities the package takes advantage of two experimental datasets from two prostate cancer experiments [19] and [16], provided by the `R` package `goProfiles` [14]. Thus, first of all, a dataset with several lists of genes, from two different studies, selected as being differentially expressed in prostate cancer is loaded into memory

```
data(prostateIds)
help("prostateIds")

## No documentation for 'prostateIds' in specified packages and libraries:
## you could try '??prostateIds'
```

Then, two subsets of Entrez Gene ID's are selected from two different lists of genes respectively.

```
## Entrez Gene ID's from Welsh et al. study
eg.we <- welsh01EntrezIDs[1:10]

## Entrez Gene ID's from Singh et al. study
eg.sg <- singh01EntrezIDs[1:10]
```

And finally, provide the name of human `R` organism package

```
pckg <- "org.Hs.eg.db"
```

3.1 Semantic similarities between GO IDs ancestors of terms that have been mapped by Entrez Genes

Function `gosims` allows to compute semantic similarities between all the pairs of GO ID ancestors of terms that annotate the selected Entre Gene ID's.

```
help("gosims")
```

The function requires the list of genes (`eg`), the ontology domain (`ontology`), the name of the organism package (`pckg`), the type of approach (`type`), and the measure used (`method`). In this example are considered all the measures from node-based approach to compute semantic similarities between GO ID's of Molecular Function (`MF`) associated with the subset of genes selected from the Welsh *et al.* study

```
## All semantic similarities of node-based approach
all.nb <- gosims(eg = eg.we, ontology = "MF", pckg = pckg,
                 type = "nb", method = "all")

## Loading required package:  org.Hs.eg.db

summarySims(all.nb)

##             n NAs      Min Num.Min    Max Num.Max     Mean
## Resnik   6903   0  0.00000    4469 4.7005      10  0.45654
## Lin      6903   1  0.00000    4468 1.0000      42  0.11142
```

```
## Rel      6903   1  0.00000    4468 0.9909      10  0.08649
## JC       6903   0  0.09614     452 1.0000      43  0.14564
## Nunivers 6903   0  0.00000    4469 1.0000      10  0.09713
## Psec     6903   0 -9.40096     452 4.7005      10 -6.45551
## Faith    6903   1  0.00000    4468 1.0000      42  0.07772
##          Std.Dev  Median
## Resnik   0.84936  0.0000
## Lin      0.21059  0.0000
## Rel      0.19429  0.0000
## JC       0.09717  0.1202
## Nunivers 0.18070  0.0000
## Psec     2.64582 -7.3215
## Faith    0.17062  0.0000
```

3.2 Semantic similarities profiles

The following functions are though for performing comparisons between two semantic similarity profiles generated according two list of genes.

The reason for comparin two lists of semantic similarities may be to understand functional gene similarities. In order to perform this type of comparison, existing packages (e.gs `GOSim` [2] and `GOSemSim` [20]) propose different approaches based on similarities that yield judgments of orientation, but not magnitudes. `sims` package considers alternative strategies that rely on a more statistical approach. Some functions allow building summaries with magnitude measures and plots for highlighting differences between profiles. The following subsections illustrate the main ideas with an example that considers the two lists of genes subsetted from the studies of Welsh *et al.* and Singh *et al.*

3.2.1 Computation of the semantic similarity profiles

To compute the semantic similarity profiles associated with each list of Entrez Gene ID's we use the function `gosimsAvsB`. It looks for the induced graph given by two lists of Entrez Gene ID's annotated in the ontology domain, and then calculates the semantic similarities between all the pairs of GO ID ancestors associated with the GO ID's that are annotating each list of genes. Figure 2 shows the schematically the idea of this step

```
## Semantic similarity profiles computed with Resnik's measure
## from node-based approach
WEvsSG.nb <- gosimsAvsB(eg1 = eg.we, eg2 = eg.sg, ontology = "MF",
                        pckg = pckg, type = "nb", method = "Res")
```

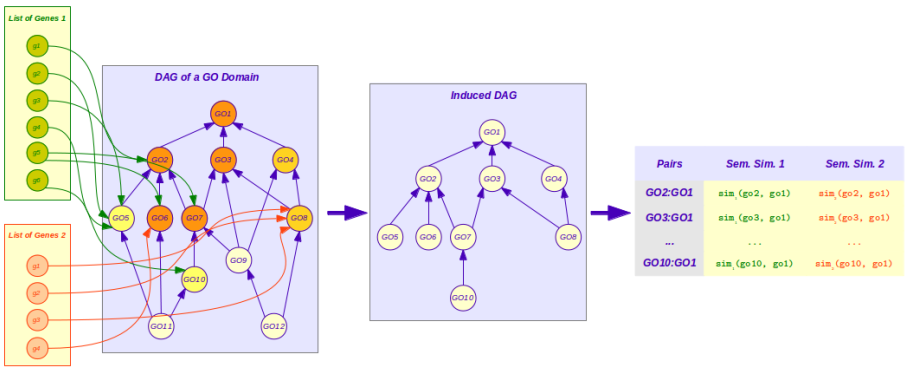


Figure 2: Schema for computing the two semantic similarities profiles associated with the two lists of genes respectively.

3.2.2 Comparison between the semantic similarity profiles

Statistical analysis is performed with the function `summarySimsAvsB`. It yields a summary that consists of (1) an statistic descriptive for each profile of semantic similarity measures, (2) a Mantel's Test [8] for examining the association between the distance matrices (i.e. the similarity matrices), and (3) a Cosine Similarity [17] for determining the similarity between the two semantic similarity profiles.

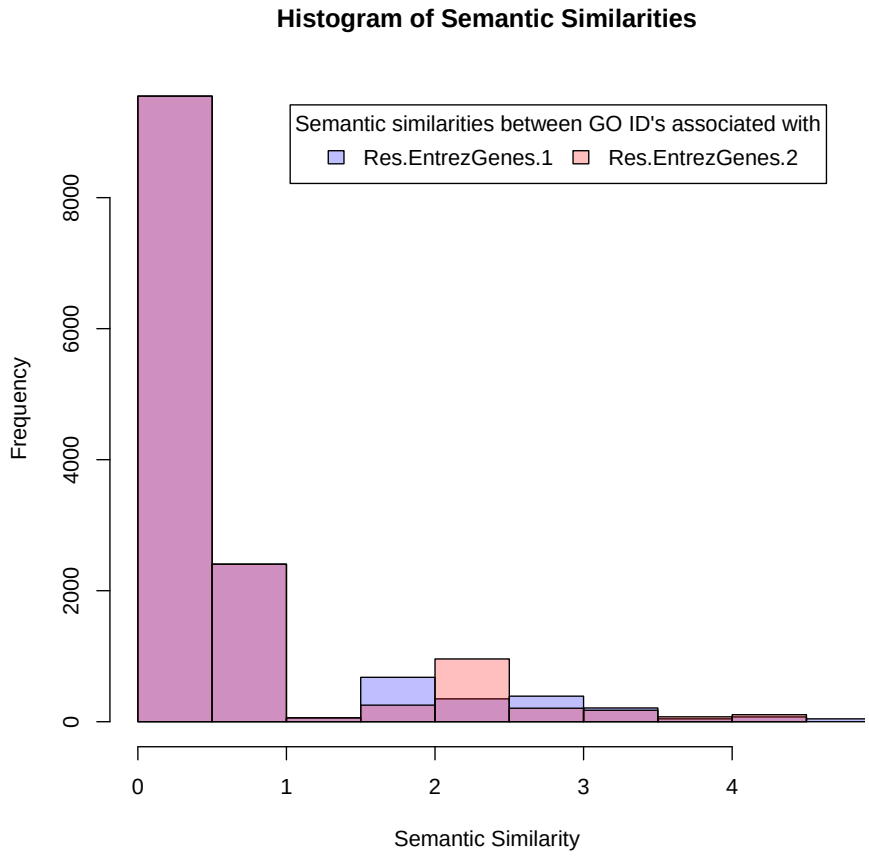
```
summarySimsAvsB(WEvsSG.nb)

## $Summary
##                         n NAs Min Num.Min   Max Num.Max
## Res.EntrezGenes.1 13861  59   0    9550 4.700      44
## Res.EntrezGenes.2 13861  73   0    9550 4.431     107
##                      Mean Std.Dev Median
## Res.EntrezGenes.1 0.4267  0.8552      0
## Res.EntrezGenes.2 0.4294  0.8713      0
##
## $Mantel
##   Mantel.r PValue
## 1   0.9698  0.001
##
## $Similarity
## [1] 0.9757
```

3.2.3 Plots for the semantic similarity profiles

In `sims` package there are three types of plots implemented. They support the statistical summary provided by the function `summaryAvsB`.

First plot is an histogram of the semantic similarity profiles. It shows both "curves" in the same plot.

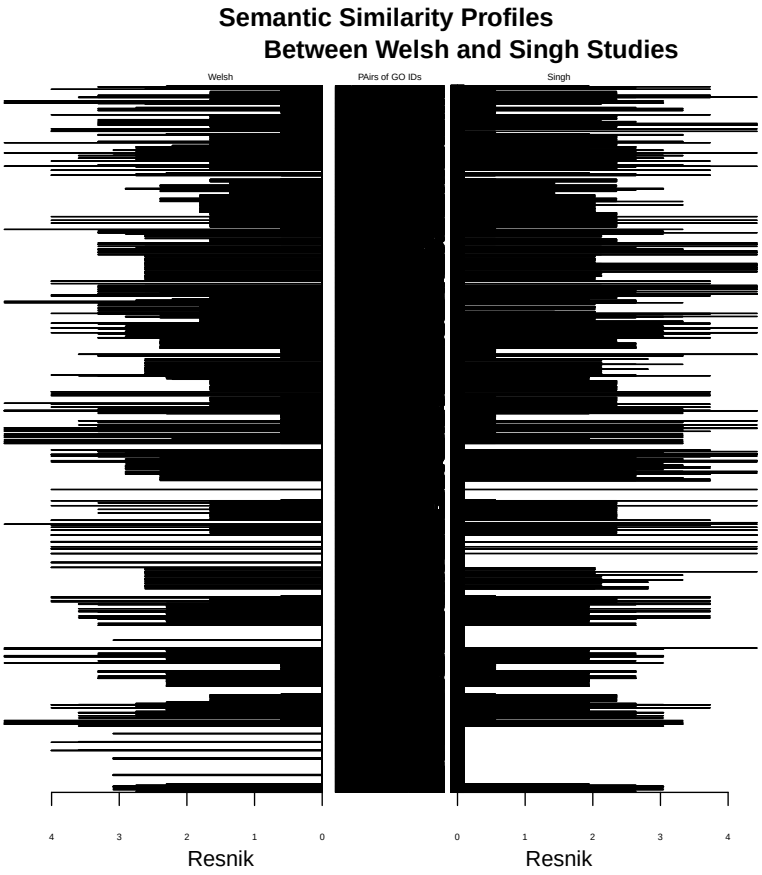
```
plotHistSims(x = WEvsSG.nb, freq = TRUE,
             main = "Histogram of Semantic Similarities",
             xlab = "Semantic Similarity")
```



Second image plots a vertical bar diagram, whose bars are associated with the semantic similarities between each pair of terms. Bars on the left side are the bars corresponding to the first list of genes and bars on the right side are the bars corresponding to the second list of genes.

```
gosimsProfiles(x = WEvsSG.nb,
               col = c("tomato", "blue"), cex = 0.4,
               top.labels = c("Welsh", "PAirs of GO IDs", "Singh"),
               main = "Semantic Similarity Profiles
                       Between Welsh and Singh Studies",
```

```
                xlab = "Resnik")
```

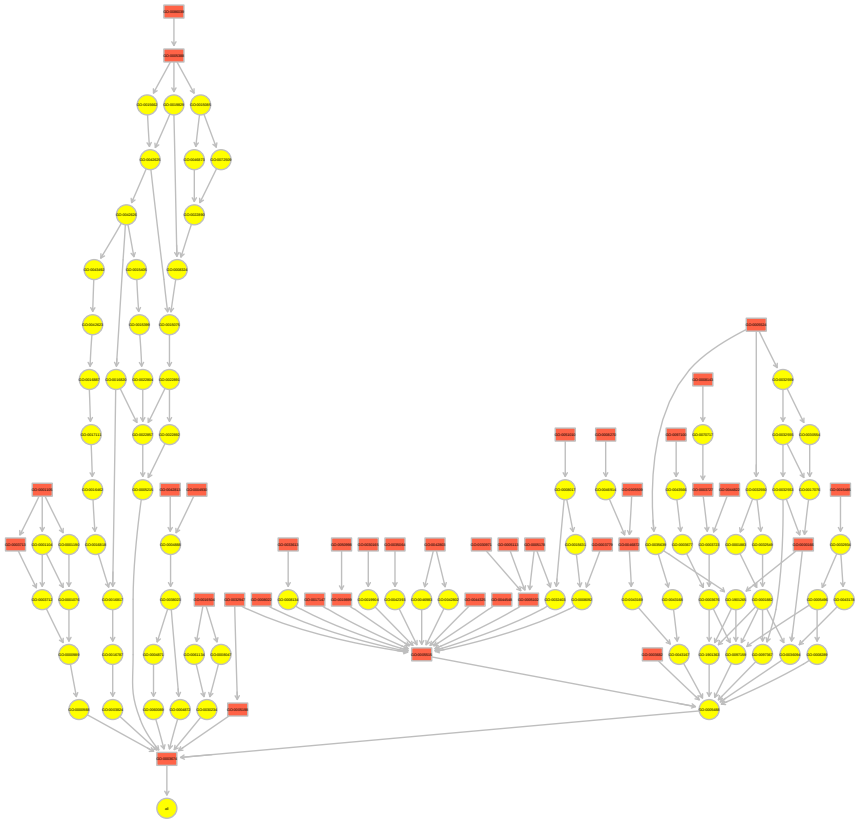



```
## [1] 5.1 4.1 4.1 2.1
```

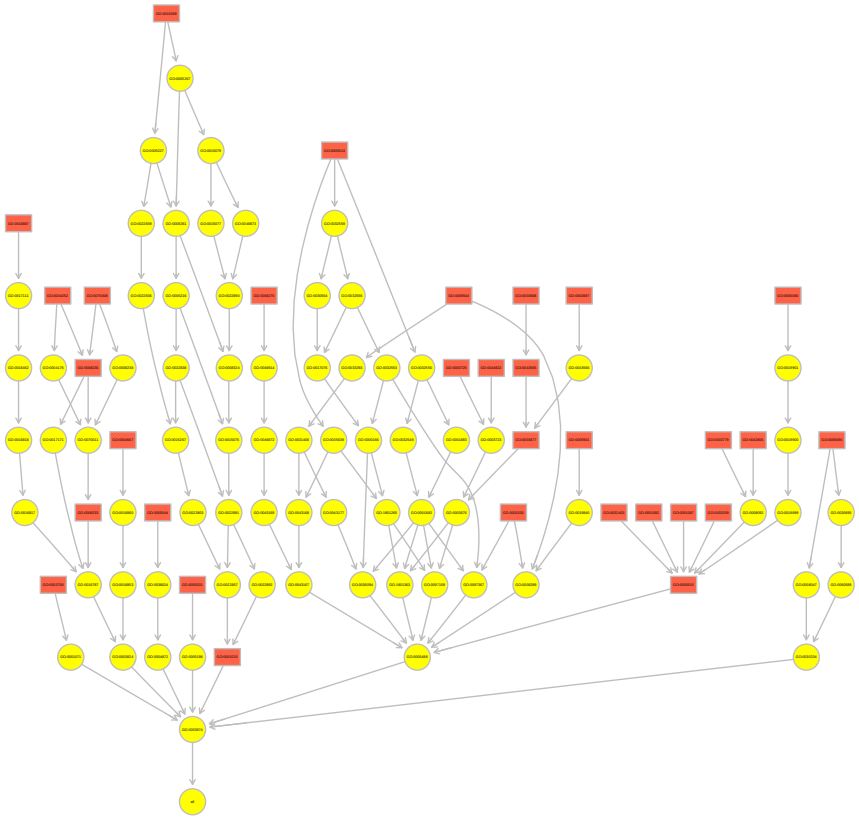
Function `plotDAG` plots the induced subgraph from the GO domain associated with one or two lists of Entrez Gene Identifiers. The subgraph shows two types of shapes for each node. Circles are GO ID's not mapped directly by the genes and rectangles are GO ID's that are mapped directly by the genes. The color of nodes indicate the type of relation with the Entrez Gene IDs. That is, when argument `eg2` is `NULL`, there are two possibilities: nodes mapped directly are shown in `red` color and their ancestors are shown in `yellow` color. But, if argument `eg2` is not `NULL`, then there are six different colors. Nodes mapped directly from the first list of Entrez Gene IDs are shown in `red` color and their ancestors are shown in `yellow` color. Nodes mapped directly from the second list of Entrez Gene IDs are shown in `lightblue` color and their ancestors are shown in `blue` color. Nodes mapped directly from both lists of Entrez Gene IDs

are shown in `magenta` color and their ancestors are shown in `violet` color.

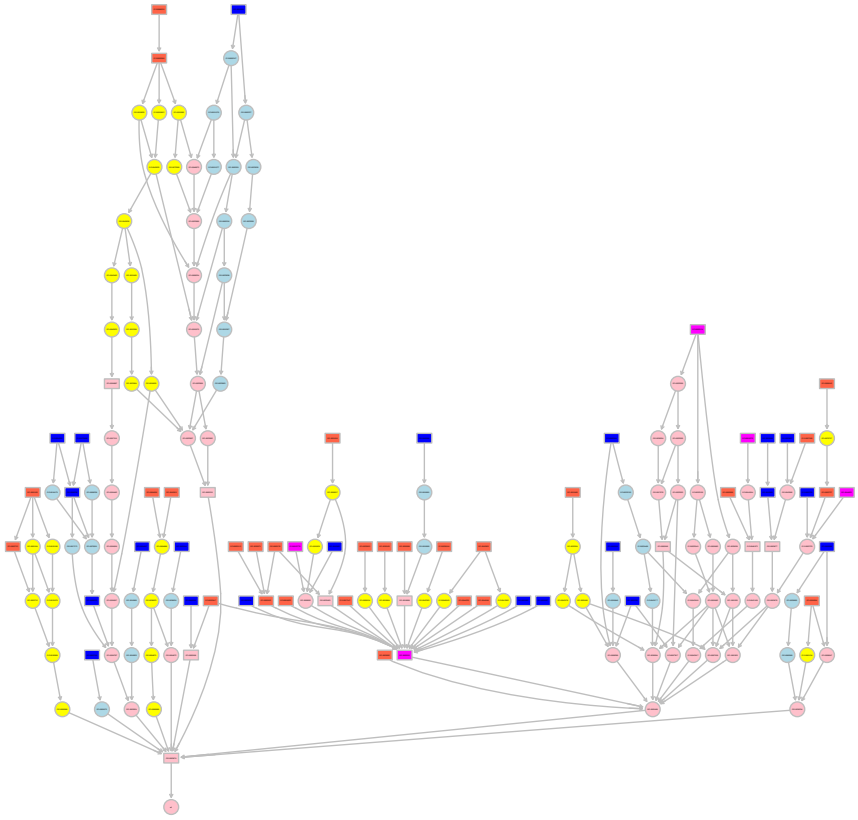
```
## Induced subgraph associated with the list of genes from
## Welsh study
plotGODAG(eg1 = eg.we, eg2 = NULL, pckg = pckg, ontology = "MF")
```



```
## [1] "A graph with 118 nodes."

## Induced subgraph associated with the list of genes from
## Singh study
plotGODAG(eg1 = eg.sg, eg2 = NULL, pckg = pckg, ontology = "MF")
```



```
## [1] "A graph with 105 nodes."

## Induced subgraph associated with both lists of genes
plotGODAG(eg1 = eg.we, eg2 = eg.sg, pckg = pckg, ontology = "MF")
```



```
## [1] "A graph with 167 nodes."
```

References

[1] Vincent J. Carey. Ontology concepts and tools for statistical genomics. *Journal of Multivariate Analysis*, 90:213–228, 2003.

[2] Holger Fröhlich, Nora Speer, Annemarie Poustka, and Tim Beißbarth. GOSim - an R-package for computation of information theoretic GO similarities between terms and gene products. *BMC Bioinformatics*, 8(166), 2007.

[3] Mingxin Gan, Xue Dou, and Rui Jiang. From ontology to semantic similarity: Calculation of ontology-based semantic similarity. *The Scientific World Journal*, (793091):1–11, 2013.

[4] Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. Semantic measures for the comparison of units of language, concepts or entities from text and knowledge base analysis. *arXiv*, (1310.1285), 2013.

[5] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference Research on Computational Linguistics (ROCLING X)*, pages 19–33, 1997. Tapei, Taiwan.

[6] Cliff A. Joslyn, Susan M. Mniszewski, Andy W. Fulmer, and Gary G. Heaton. The gene ontology categorizer. *Bioinformatics*, 20(s1):169–77, 2004.

[7] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann Publishers, 1998.

[8] Nathan Mantel. The detection of disease clustering and a generalized regression approach. *Cancer Research*, 27(2 Part 1):209–220, 1967.

[9] Gaston K. Mazandu and Nicola J. Mulder. Information content-based gene ontology semantic similarity approaches: Toward a unified framework theory. *BioMed Research International*, (292063):1–11, 2013.

[10] Giuseppe Pirró and Jérôme Euzenat. A feature and information theoretic framework for semantic similarity and relatedness. In Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Z. Pan, Ian Horrocks, and Birte Glimm, editors, *The Semantic Web - ISWC 2010*, volume 6496 of *Lecture Notes in Computer Science*, pages 615–630. Springer Berlin Heidelberg, 2010.

[11] Giuseppe Pirró and Nuno Seco. Design, implementation and evaluation of a new semantic similarity metric combining features and intrinsic information content. In Robert Meersman and Zahir Tari, editors, *On the Move to Meaningful Internet Systems: OTM 2008*, volume 5332 of *Lecture Notes in Computer Science*, pages 1271–1288. Springer Berlin Heidelberg, 2008.

[12] Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1):17–30, 1989.

[13] Phillip Resnik. Using information content to evaluate semantic similarity in a taxonomy. pages 448–453. Int. Joint Conf. on Artificial Intelligence, Kaufmann, Morgan, 1995.

[14] Alex Sánchez, Jordi Ocaña, and Miquel Salicrú. *goProfiles: goProfiles: an R package for the statistical analysis of functional profiles*, 2010. R package version 1.24.0.

[15] Andreas Schlicker, Francisco S. Domingues, Jörg Rahnenführer, and Thomas Lengauer. A new measure for functional similarity of gene products based on Gene Ontology. *BMC Bioinformatics*, 7:302, 2006.

[16] Dinesh Singh, Philip K. Febbo, Kenneth Ross, Donald G. Jackson, Judith Manola, Christine Ladd, Pablo Tamayo, Andrew A. Renshaw, Anthony V. DAmico, Jerome P. Richie, Eric S. Lander, Massimo Loda, Philip W. Kantoff, Todd R. Golub, and William R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203–209, 2002.

[17] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., Inc., firt edition edition, 2005.

[18] The Gene Ontology Consortium. The gene ontology (go) database and informatics resource. *Nucleic Acids Research*, 32(Suppl. 1):D258–D261, 2004.

[19] John B. Welsh, Lisa M. Sapinoso, Andrew I. Su, Suzanne G. Kern, Jessica Wang-Rodriguez, Christopher A. Moskaluk, Henry F. Frierson Jr., and Garret M. Hampton. Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer. 61(16):5974–5978, 2001.

[20] Guangchuang 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*, 26(7):976–978, 2010.