How To Use DOSim

Jiang Li

June 21, 2010

Contents

1 Overview

This vignette demonstrates how to easily use the DOSim package. DOSim is used to calculate DO terms similarity and genes similarity based on terms similarity, and meanwhile it provides information for disease ontology and can do DO Enrichment analysis. We take GOSim [1] as a refernece to organize our code.

To start with DOSim package, type following code below:

```
> library(DOSim)
> help(DOSim)
```

2 Calculate DO Terms Similarity

Terms in disease ontology(DO) are organized in Directed Acyclic Graph (DAG). Previous studies have developed many methods to calculate their similarities. DOSim implements two types of approaches for calculation of similarity between terms in Disease Ontology: node-based, in which the main data sources are the nodes and their properties; and edge-based, which use the edges and their types as the data source. Totally thirteen different methods are implemented, ten out of thirteen are node-base and the left threes are edge-based. The function *getTermSim* is the interface for user to calculate DO terms similarity.

An example of how to calculate DO Terms similarity is shown below:

```
> termlist = c("DOID:399", "DOID:1117", "DOID:2313", "DOID:2040")
> tsim <- getTermSim(termlist, method = "relevance", verbose = TRUE)
```

```
> tsim
```

```
           DOID:399 DOID:1117 DOID:2313 DOID:2040
DOID:399  0.9765664 0.3421396 0.9609378         0
DOID:1117 0.3421396 0.9610261 0.3471034         0
DOID:2313 0.9609378 0.3471034 0.9740997         0
DOID:2040 0.0000000 0.0000000 0.0000000         1
```

Detailed information for each method implemented in DOSim is shown below:

2.1 Resnik

Resnik'measure is one of the most common semantic similarity measures, it was originally developed for the WordNet[2]. Resnik measures similiarity between two terms as simply the IC of their most informative common ancestor (MICA), which is defined as follows:

$$Sim_{Resnik}\left(t_1, t_2\right) = IC\left(t_{MICA}\right) \tag{1}$$

2.2 JiangConrath

Jiang and Conrath developed measures that scale the information content of the MICA by the information content of the individual concepts [3]. It is defined as:

$$Sim_{JC}\left(t_1, t_2\right) = 1 - \min\left(1, IC\left(t_1\right) + IC\left(t_2\right) - 2 \times IC\left(t_{MICA}\right)\right) \tag{2}$$

2.3 Lin

Lin's measure is the extension of Resnik's by taking the distance of how distant the terms are from their common ancestor into account[4]. It is defined as:

$$Sim_{Lin}\left(t_1, t_2\right) = \frac{2 \times IC\left(t_{MICA}\right)}{IC\left(t_1\right) + IC\left(t_2\right)} \tag{3}$$

2.4 CoutoEnriched

This measure begin with the semantic distance (inverse of similarity) and is proposed by Couto in 2003 [5]. The semantic distance between t_1 and t_2 when t_1 subsumes t_2 is quantified as follows:

$$\Delta\left(t_1, t_2\right) = IC\left(t_2\right) - IC\left(t_1\right) \tag{4}$$

When two terms t_2 and t_3 without a subsuming relation, the semantic distance is the sum of their semantic distance to their most informative common ancestor t_1. Thus, the semantic distance between term t_2 and t_3 is quantified as follows:

$$\Delta\left(t_2, t_3\right) = \Delta\left(t_1, t_2\right) + \Delta\left(t_1, t_3\right) \tag{5}$$

The distance defined in equations 4 and 5 does not use any conceptual distance factors. Thus, we have to redefine this distance to integrate the node depth and density factors. Considering a term t_0 that subsumes t_n, and the sequence of terms $t_0, \ldots, t_n$ representing the path from t_0 to t_n with length n, the senmantic distance between t_0 and t_n is redefeind as follows:

$$\Delta\left(t_0, t_n\right) = \sum_{i=0}^{n-1} D\left(t_i\right) \times E\left(ti\right) \times \left(IC\left(t_{(i+1)}\right) - IC\left(t_i\right)\right) \tag{6}$$

where $D(t)$ and $E(t)$ represent the depth and density conceptual distance factors for a term t.

$D(t)$ is defined as follows:

$$D(t) = \left(\frac{d(t)+1}{d(t)}\right)^{\alpha} \tag{7}$$

where $d(t)$ denotes the depth of term t in the ontology. The α parameter controls the degree of how much the depth factor contributes in equation 6. When α approaches 0, this contribution becomes less significant, since $D(t)$ will apprach 1.

$E(t)$ is defined as follows:

$$E(t) = (1-\beta) \times \frac{\overline{E}}{e(t)} + \beta \tag{8}$$

where $e(t)$ denotes the local density of the term t, i.e. the number of edges that start from t. $\overline{E}$ represents the average density in the whole ontology, i.e. the number of edges divided by the number of terms in the ontology. The β parameter controls the degree of how much the density factor contributes in equation 6. When β approaches 1, this contribution becomes less significant, since $E(t)$ will approach 1.

By normalizing the distance defined in equataion 6, we finally get the semantic similarity between term t_1 and t_2 as follows:

$$Sim_{CoutoEnriched}(t_1, t_2) = 1 - \min\left(1, \frac{\Delta(t_1, t_2)}{IC(t_0)}\right) \tag{9}$$

where $IC(t_0)$ represents the maximum information content possible in the ontology.

2.5 CoutoResnik

In 2005, Couto et.al [6] proposed the GraSM (Graph-based Similarity Measure) approach by introducing the concept of disjunctive common ancestor (DCA). For a term t, GraSM considers that a_1 and a_2 represent disjunctive ancestors of t if there is a path from a_1 to t not passing through a_2 and a path from a_2 to t not passing through a_1, it is defiend as follows:

$$\begin{aligned} DisjAnc(t) = \{(a_1, a_2) \mid \\ (\exists p : (p \in Paths(a_1, t)) \wedge (a_2 \notin p)) \wedge \\ (\exists p : (p \in Paths(a_2, t)) \wedge (a_1 \notin p))\} \end{aligned} \tag{10}$$

Given two terms t_1 and t_2, their DCAs are the most informative common ancestor of disjunctive ancestors of t_1 and t_2 and is defiend as follows:

$$\begin{aligned} DisjCommonAnc\left(t_1,t_2\right) = \{a_1 \mid & \\ & a_1 \in CommonAnc\left(t_1,t_2\right) \wedge \\ & \forall a_2 : [(a_2 \in CommonAnc\left(t_1,t_2\right)) \wedge (IC\left(a_1\right) \leq IC\left(a_2\right))] \Rightarrow \\ & [(a_1,a_2) \in (DisjAnc\left(t_1\right) \cup DisjAnc\left(t_2\right))]\} \end{aligned} \tag{11}$$

GraSM defines the shared information of two terms as the average of the imformation content of the DCAs compared with the inforamtion content of the MICA.

$$Share_{GraSM}\left(t_1,t_2\right) = \overline{\{IC\left(a\right) \mid a \in DisjCommonAnc\left(t_1,t_2\right)\}} \tag{12}$$

It can overcome the constrains when only looking at the MICA and can capture more interpretations for both terms in a DAG ontolgy. Meanwhile it can be applied to any of the measures previously described (Resnik's, JiangConrath's and Lin's) by replacing the IC of the MICA with the average IC of all DCAs. Method named 'CoutoResnik' in DOSim is similar to Resnik's and it is defined as follows:

$$Sim_{CoutoResnik}\left(t_1,t_2\right) = Share_{GraSM}\left(t_1,t_2\right) \tag{13}$$

2.6 CoutoJiangConrath

It is similar to JiangConrath's by replacing the IC of the MICA with the average IC of all DCAs and defined as follows:

$$Sim_{CoutoJC}\left(t_1,t_2\right) = 1 - \min\left(1, IC\left(t_1\right) + IC\left(t_2\right) - 2 \times Share_{GraSM}\left(t_1,t_2\right)\right) \tag{14}$$

2.7 CoutoLin

It is similar to Lin's by replacing the IC of the MICA with the average IC of all DCAs and defined as follows:

$$Sim_{CoutuLin}\left(t_1,t_2\right) = \frac{2 \times Share_{GraSM}\left(t_1,t_2\right)}{IC\left(t_1\right) + IC\left(t_2\right)} \tag{15}$$

2.8 relevance

As Lin's measure is displaced from the graph [7], Schlicker et al. [8] have proposed the relevance similarity measure, which is based on Lin's measure, but use the probalility of annotation of the MICA as a weighting factor to provide graph placement, which is defined as follows:

$$Sim_{relevance}\left(t_1,t_2\right) = Sim_{Lin}\left(t_1,t_2\right) \times \left(1 - p\left(t_{MICA}\right)\right) \tag{16}$$

where $p\left(t_{MICA}\right) = e^{-IC\left(t_{MICA}\right)}$ in DOSim package.

2.9 GIC

This method is proposed by Presquita et al.[7] and it is an expansion of graph-based similarity measures. GIC stands for 'Graph Information Content' and it is defined as follows:

$$Sim_{GIC}(t_1, t_2) = \frac{\sum_{t \in (Ancestor(t_1) \cap Ancestor(t_2))} IC(t)}{\sum_{t \in (Ancestor(t_1) \cup Ancestor(t_2))} IC(t)} \tag{17}$$

2.10 simIC

This similarity measure is quite like the relevance measure, it is called information coefficient similarity measure which effectively intergrates both the information content and the structural information of terms in a DAG ontolgy[9]. It is defined as follows:

$$Sim_{simIC}(t_1, t_2) = Sim_{Lin} \times \left(1 - \frac{1}{1 + IC(t_{MICA})}\right) \tag{18}$$

2.11 path

This path-length measure is simple and proposed by Wu et al.[10] in 1994, it is defined as follows:

$$Sim_{path}(t_1, t_2) = \frac{1}{p} \tag{19}$$

where p is the number of nodes on the shortest path between two terms in a DAG ontology.

2.12 lch

The Leacock and Chodorow measure (lch) [11] is computed as

$$Sim_{lch}(t_1, t_2) = -\log\left(\frac{p}{2 * depth}\right) \tag{20}$$

where p is the number of nodes on the shortest path between two terms in a DAG ontology and *depth* is the maximum depth of the hierarchy.

2.13 Wang

In Wang's measure, each edge is given a weight according to the type of relationship [12]. For term A, it can be represented as $DAG_A = (A, T_A, E_A)$ where T_A is the set of all ancestor terms of A including A itself and E_A is the set of edges connecting the terms in DAG_A. For any term t in DAG_A, Wang et al. defines the semantic contribution of t

to A as the product of all edge weights in the "best" path from t to A, where the "best" path is the one that maximizes the product. It can be calculated by

$$\begin{cases} S_A(A) = 1 \\ S_A(t) = \max\left\{w_e \times S_A\left(t'\right) \mid t' \in childrenof\,(t)\right\} \quad if\ t \neq A \end{cases} \tag{21}$$

where w_e is the semantic contribution factor of edge $e \in E_A$. The senmantic similarity between two terms is then calculated by summing the semantic contibutions of all common ancestors to each of the terms and diving by the total semantic contribution of each term's ancestors to that term which can be calculated by

$$Sim_{Wang}(A, B) = \frac{\sum_{t \in T_A \cap T_B} (S_A(t) + S_B(t))}{SV(A) + SV(B)} \tag{22}$$

where $SV(A)$ is the total semantic contribution to term A in DAG_A, which is calculated by

$$SV(A) = \sum_{t \in T_A} S_A(t) \tag{23}$$

3 Calculate Genes Similarity

Genes similarity is calculated based on their complete DO annotation. Each gene is represented by its set of direct annotations and semantic similarity is calculated between terms in one set and terms in the other (using one of the approaches defined above). DOSim provides users a function named *getGeneSim* to calculate genes similarity. It provides 8 methods to calculate genes similarity. A basic example is shown below:

```
> genelist <- c("10003", "10008", "10015", "10042", "10036")
> gsim <- getGeneSim(genelist, similarity = "max", similarityTerm = "Lin")

> gsim
```

```
          10003 10008      10015 10042      10036
10003 1.0000000     0 0.00000000     0 0.10239441
10008 0.0000000     1 0.00000000     0 0.00000000
10015 0.0000000     0 1.00000000     0 0.03210972
10042 0.0000000     0 0.00000000     1 0.00000000
10036 0.1023944     0 0.03210972     0 1.00000000
```

Detailed information for each method is described below:

3.1 max

This method is straight forward. Given two genes g and $g^{'}$ annotated with DO terms $t_1, \ldots, t_n$ and $t_1^{'}, \ldots, t_m^{'}$, the functional similarity between two genes g and $g^{'}$ is defined as follows:

$$Sim_{max}\left(g, g^{'}\right) = \max_{\substack{i=1,\ldots,n \\ j=1,\ldots,m}} sim\left(t_i, t_j^{'}\right) \tag{24}$$

3.2 mean

It is similar to max method just by taking averaging the pairwise DO term similarity here and it is defined as follows:

$$Sim_{mean}\left(g, g^{'}\right) = \frac{\sum_{\substack{i=1,\ldots,n \\ j=1,\ldots,m}} sim\left(t_i, t_j^{'}\right)}{m \times n} \tag{25}$$

3.3 funSimMax

This method is proposed by Schlicker et.al[8] using the best pairs technique. Given two genes g and $g^{'}$ annotated with DO terms $t_1, \ldots, t_n$ and $t_1^{'}, \ldots, t_m^{'}$, a similarity matrix S is calculated with any of the methods for DO terms mentioned above (S is a $n \times m$ matrix). The rows and columns of S represent two different directional comparisons, row vetors correspond to a comparison of g to $g^{'}$ and column vectors of $g^{'}$ to g. Simlarity values for the comparison of g to $g^{'}$ ($rowScore$) and the comparison of $g^{'}$ to g ($columnScore$) are defined as follows:

$$rowScore = \frac{1}{n}\sum_{i=1}^{n} \max_{1 \le j \le m} s_{ij} \tag{26}$$

$$columnscore = \frac{1}{m}\sum_{j=1}^{m} \max_{1 \le i \le n} s_{ij} \tag{27}$$

Method *funSimMax* takes the maximum value between $rowScore$ and $columnScore$ as the similarity value for genes g and $g^{'}$, which can be calculated as follows:

$$Sim_{funSimMax}\left(g, g^{'}\right) = \max\{rowScore, columnScore\} \tag{28}$$

3.4 funSimAvg

This method is similar to *funSimMax* just by taking the average value between $rowScore$ and $columnScore$ instead of taking the maximum value[8]. It is defined as follows:

$$Sim_{funSimAvg}\left(g, g^{'}\right) = \frac{rowScore + columnScore}{2} \tag{29}$$

3.5 OA

OA stands for optimal assignment, it means assigning each term of the gene having fewer DO terms to exactly one term of the other gene such that the overal similarity is maximized [1, 13]. It can be state as follows: let π be some permutation of either an n-subset of natural numbers $\{1, \ldots, m\}$ or a m-subset of natural numbers $\{1, \ldots, n\}$, then we are looking for the quantity:

$$Sim_{OA}\left(g, g^{'}\right) = \begin{cases} \max_{\pi} \sum_{i=1}^{n} sim\left(t_i, t^{'}_{\pi(i)}\right) & if\ m \leq n \\ \max_{\pi} \sum_{j=1}^{m} sim\left(t_{\pi(j)}, t^{'}_{j}\right) & otherwise \end{cases} \tag{30}$$

where $sim\left(t_i, t^{'}_{\pi(i)}\right)$ and $sim\left(t_{\pi(j)}, t^{'}_{j}\right)$ are any of the similarity methods for DO terms mentioned above.

3.6 hausdorff

Here we apply the Hausdorff Distance (inverse of similarity) to calculated functional similarity between two genes from their set of DO terms [14]. Hausdorff Distance is defined as the maximum value between any point within one set (A) and the nearest point in the other set (B). Hausdorff Distance from set A to B is defined as follows:

$$Dist^{a \to b}_{hausdorff} = \max_{a \in A}\left\{\min_{b \in B}\left(Dist\left(a, b\right)\right)\right\} \tag{31}$$

where $Dist\left(a, b\right)$ is the distance metric betwee term a and b. Then we defined the Hausdorff Distance between set A to B as follows:

$$Dist_{hausdorff} = \max\left(Dist^{a \to b}_{hausdorff}, Dist^{b \to a}_{hausdorff}\right) \tag{32}$$

Given two genes g and $g^{'}$ with their set of DO terms A and B respectively, together with the simimarity matrix S, we defined the similarity between genes g and $g^{'}$ using the Hausdorff Distance method based on the similarity matrix S as follows:

$$Sim_{hausdorff}\left(g, g^{'}\right) = \min\left(Sim^{a \to b}_{hausdorff}\left(g, g^{'}\right), Sim^{b \to a}_{hausdorff}\left(g, g^{'}\right)\right) \tag{33}$$

where $Sim^{a \to b}_{hausdorff}\left(g, g^{'}\right)$ is the hausdorff similarity from set A to B and it is formulated as:

$$Sim^{a \to b}_{hausdorff}\left(g, g^{'}\right) = \min_{a \in A}\left\{\max_{b \in B}\left(Sim\left(a, b\right)\right)\right\} \tag{34}$$

3.7 dot

First, for each gene g, we construct a feature vector $\phi(g)$ relative to a set of prototype genes $p = (p_1, \ldots, p_n)$ which is defined as:

$$\phi\left(g\right) = \left(sim_{max}\left(g, p_1\right), \ldots, sim_{max}\left(g, p_n\right)\right)^T \tag{35}$$

Then the similarity between genes g and $g^{'}$ is the dot product of their feature vector defined by equation 35 after normalizing which can be formulated as:

$$Sim_{dot}\left(g, g^{'}\right) = \frac{< \phi\left(g\right), \phi(g^{'}) >}{\sqrt{< \phi\left(g\right), \phi(g) > \times < \phi\left(g^{'}\right), \phi(g^{'}) >}} \tag{36}$$

3.8 Wang

It is proposed by Wang et.al [12] used a best-match (best pairs technique) average combination strategy. Given two genes g and $g^{'}$ annotated with DO terms $t_1, \ldots, t_n$ (DO_1) and $t_1^{'}, \ldots, t_m^{'}$ (DO_2), we will find it is quite like the method *funSimAvg* as it is defined as follows:

$$Sim_{wang}\left(g, g^{'}\right) = \frac{\sum\limits_{1 \leq i \leq n} Sim\left(t_i, DO_2\right) + \sum\limits_{1 \leq j \leq m} Sim\left(t_j^{'}, DO_1\right)}{n + m} \tag{37}$$

where $Sim\left(t_i, DO_2\right)$ and $Sim\left(t_j^{'}, DO_1\right)$ are defined as:

$$Sim(t_i, DO_2) = \max_{1 \leq j \leq m} s_{ij}$$
$$Sim(t_j^{'}, DO_1) = \max_{1 \leq i \leq n} s_{ij}$$

4 Get Information of Disease Ontology

The Disease Ontology is a community driven,open source ontology that is designed to link disparate datasets through disease concepts. Terms in DO are organized in Directed Acyclic Graph (DAG). With the work of John D.Osborne in 2009[15], human genes are annotated to DO terms. In DOSim, we provide 7 functions to fetch information of DO terms. They are:

- $getParents$
- $getAncestors$
- $getOffsprings$
- $getChildren$
- $getDoTerm$
- $getDoAnno$
- $getDOGraph$

Basic examples of each of the 7 functions are show in the following sections below:

4.1 getParents

Returns a list of all direct parents associated to each DO term.

```
> terms <- c("DOID:934", "DOID:1579")
> getParents(terms)

[1] "Start to fetch the parents"
$`DOID:934`
[1] "DOID:0050117"

$`DOID:1579`
[1] "DOID:13"
```

4.2 getAncestors

Returns the list of all ancestors associated to each DO term.

```
> terms <- c("DOID:934", "DOID:1579")
> getAncestors(terms)

[1] "Start to fetch the ancestors"
$`DOID:934`
[1] "DOID:0050117" "DOID:4"

$`DOID:1579`
[1] "DOID:4"  "DOID:13" "DOID:7"
```

4.3 getOffsprings

Returns the list of all offspring associated to each DO term.

```
> terms <- c("DOID:10533", "DOID:550")
> getOffsprings(terms)

[1] "Start to fetch the offsprings"
$`DOID:10533`
[1] "DOID:14473" "DOID:14476" "DOID:14475" "DOID:10510" "DOID:14474"
[6] "DOID:14472" "DOID:14477"

$`DOID:550`
[1] NA
```

4.4 getChildren

Returns the list of all direct children associated to each DO term.

```
> terms <- c("DOID:934", "DOID:1579")
> getChildren(terms)

[1] "Start to fetch the children"
$`DOID:934`
 [1] "DOID:0050079" "DOID:10533"   "DOID:1301"    "DOID:1329"    "DOID:13801"
 [6] "DOID:1385"    "DOID:1884"    "DOID:2295"    "DOID:2931"    "DOID:2932"
[11] "DOID:2937"    "DOID:2940"    "DOID:2947"    "DOID:2950"    "DOID:3294"
[16] "DOID:4121"    "DOID:623"     "DOID:6297"    "DOID:8568"    "DOID:8672"
[21] "DOID:8867"    "DOID:937"

$`DOID:1579`
 [1] "DOID:0050161" "DOID:10458"   "DOID:11091"   "DOID:1116"    "DOID:11565"
 [6] "DOID:1273"    "DOID:2945"    "DOID:4298"    "DOID:4493"    "DOID:9395"
[11] "DOID:974"
```

4.5 getDoTerm

Returns the list of DO term's name associated to each DO ID.

```
> terms <- c("DOID:934", "DOID:1579")
> getDoTerm(terms)

$`DOID:934`
[1] "viral infectious disease"

$`DOID:1579`
[1] "respiratory system disease"
```

4.6 getDoAnno

Get gene list associated to each DO term

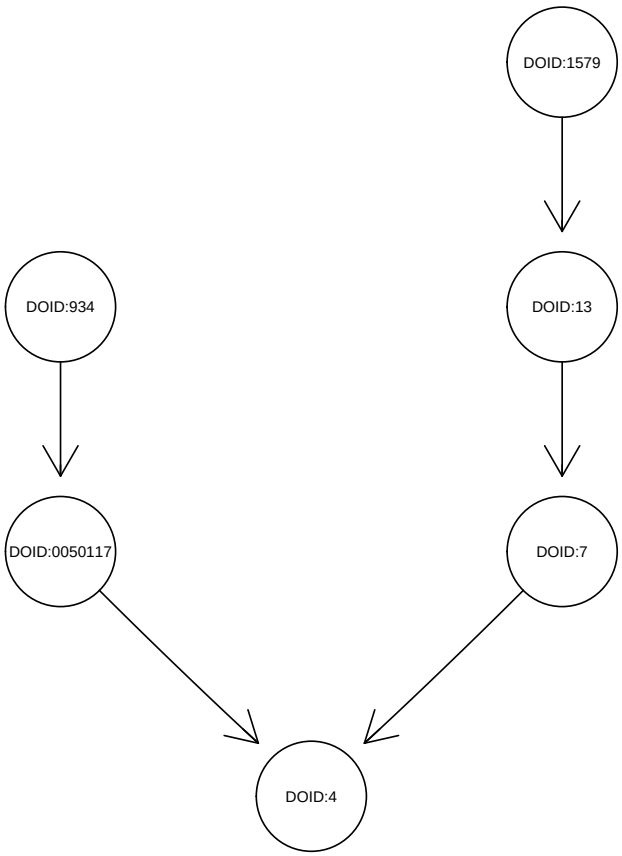
```
> terms <- c("DOID:1579")
> getDoAnno(terms)

$`DOID:1579`
[1] "1636"
```

4.7 getDOGraph

Get DO graph with specified DO terms at its leave.

```
> terms <- c("DOID:934", "DOID:1579")
> if (require(graph)) {
+     g <- getDOGraph(terms)
+     if (require(Rgraphviz)) {
+         plot(g)
+     }
+ }
```



5 DO Enrichment Analysis

DOSim can do DO enrichment analysis for a list of Entrez gene ids by using **hyper geometric test** or **fisher test**. To do it, you can simply invoke the function *DOEnrichment*. Here is an example.

```
> genelist = as.character(1:100)
> DOEnrichment(genelist, method = "hypertest", filter = 50, cutoff = 0.001)

                DOID       pvalue      odds genenum1 genenum2
DOID:14330 DOID:14330 3.732400e-07 18.068317      101        5
DOID:10652 DOID:10652 2.854039e-05  8.527570      214        5
DOID:759     DOID:759 3.416859e-05  8.257466      221        5
DOID:10591 DOID:10591 2.537661e-04  9.864324      111        3
DOID:12603 DOID:12603 3.777357e-04 14.312941       51        2
DOID:3683   DOID:3683 4.000677e-04 14.037692       52        2
DOID:722     DOID:722 4.232310e-04 13.772830       53        2
DOID:9074   DOID:9074 4.761334e-04  8.358321      131        3
DOID:10825 DOID:10825 5.519650e-04 12.585517       58        2
DOID:10283 DOID:10283 5.594589e-04  4.243953      516        6
DOID:3300   DOID:3300 8.417936e-04 10.894925       67        2
DOID:2370   DOID:2370 8.417936e-04 10.894925       67        2
DOID:12849 DOID:12849 8.789028e-04 10.734706       68        2
```

References

[1] Frohlich H, Speer N, Poustka A, BeiSZbarth T: **GOSim - an R-package for computation of information theoretic GO similarities between terms and gene products**. *BMC Bioinformatics* 2007, **8**:166.

[2] Resnik P: **Using Information Content to Evaluate Semantic Similarity in a Taxonomy**. *Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal* 1995, **1**:448–453.

[3] Jiang J, Conrath D: **Semantic similarity based on corpus statistics and lexical taxonomy**. *Proceedings of the International Conference on Research in Computational Linguistics, Taiwan* 1998.

[4] Lin D: **An Information-Theoretic Definition of Similarity** 1998, :296–304.

[5] Couto F, Silva M, Coutinho P: **Implementation of a Functional Semantic Similarity Measure between Gene-Products**. *Tech Rep DI/FCUL TR 03-29* 2003.

[6] Couto F, Silva M, Coutinho P: **Semantic Similarity over the Gene Ontology: Family Correlation and Selecting Disjunctive Ancestors**. *Conference in Information and Knowledge Management* 2005.

[7] C Pesquita DF: **Evaluating GO-based Semantic Similarity Measures**. *In: Proc. 10th Annual Bio-Ontologies Meeting* 2007, :37–40.

[8] A Schlicker FD: **A new measure for functional similarity of gene products based on Gene Ontology**. *BMC Bioinformatics* 2006.

[9] B Li AF J Wang: **Effectively Integrating Information Content and Structural Relationship to Improve the GO-based Similarity Measure Between Proteins**. *BMC Bioinformatics* 2009.

[10] Wu Z PM: **Verb semantics and lexical selection**. *In:Proceedings of the 32nd annual meeting of the association for computational linguistics* 1994, :133–8.

[11] Leacock C CM: **Combining local context and WordNet similarity for word sense identification**. *In: Fellbaum C, editor. WordNet: An electronic lexical database. Cambridge, MA: MIT Press* 1998, :265–83.

[12] James ZWang ZD: **A new method to measure the semantic similarity of GO terms**. *Bioinformatics* 2007, :1274–1281.

[13] H Froehlich NS: **Kernel Based Functional Gene Grouping**. *Proc. Int. Joint Conf. on Neural Networks (IJCNN)* 2006, :6886 – 6891.

[14] A del Pozo AV F Pazos: **Defining functional distances over Gene Ontology**. *BMC Bioinformatics* 2008, :9:50.

[15] Osborne J, Flatow J, Holko M, Lin S, Kibbe W, Zhu L, Danila M, Feng G, Chisholm R: **Annotating the human genome with Disease Ontology**. *BMC Genomics* 2009, **10**(Suppl 1):S6.