How To Use DOSim

Jiang Li

July 15, 2010

Contents

1 Overview

This vignette demonstrates how to use the DOSim package easily. DOSim is used to calculate DO terms similarity and genes similarity, and meanwhile it provides functions to extract information about Disease Ontology (DO) (e.g. visualizing terms hierachies) and conduct DO Enrichment analysis.

To use DOSim package, start with the following codes below:

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

2 Measuring the Similarity between DO Terms

Terms in DO are organized in Directed Acyclic Graph (DAG). Previous studies have developed many methods to calculate terms similarities. DOSim implements two types of approaches for the calculation of similarity between terms in DO. One is node-based, in which the calculation of the similarity between DO terms is based on the attributes of nodes (node properties); the other is edge-based, which uses the attributes of edges between nodes (edge types) as the measure. In node-based approaches, information content (IC) is used to quantify the specificity and information of a DO term. In our work, the IC of a DO term t is defined as follows:

$$IC(t) = -\log p(t) \tag{1}$$

In total, DOSim implementes thirteen different methods to calculate the similarity between DO terms, of which ten are node-based, and three are edge-based. The function *getTermSim* is the interface for users to calculate DO terms similarity.

An example of how to calculate DO terms similarity is shown as 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
```

In the following text, we will introduce all these thirteen methods in an individual subsection, which is named after the parameter name for "method" in function *getTermSim*.

2.1 Resnik

The measure proposed by Resnik is one of the most widely used semantic similarity measures [1]. It measures the similarity between two terms t_1 and t_2 as simply the IC of their most informative common ancestor (MICA), which possesses the larget IC among the common ancestor terms of these two terms. The formula is as follows:

$$Sim_{Resnik}(t_1, t_2) = IC(t_{MICA}) \tag{2}$$

2.2 JiangConrath

Compared with Resnik's method, Jiang and Conrath's method scales the information content of the MICA by the information content of the individual terms [2]. It is defined as follows:

$$Sim_{JC}(t_1, t_2) = 1 - \min(1, IC(t_1) + IC(t_2) - 2 \times IC(t_{MICA})) \tag{3}$$

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[3]. It is defined as follows:

$$Sim_{Lin}(t_1, t_2) = \frac{2 \times IC(t_{MICA})}{IC(t_1) + IC(t_2)} \tag{4}$$

2.4 CoutoEnriched

This method is proposed by Couto in 2003 [4]. In the matter of our work, if DO terms t_1 and t_2 are with an ascendant-descendant relationship, the semantic distance (inverse of similarity) between t_1 and t_2 is defined as follows:

$$\Delta(t_1, t_2) = IC(t_2) - IC(t_1) \tag{5}$$

If there isn't an ascendant-descendant relationship between t_2 and t_3, the semantic distance between t_2 and t_3 is measured as the sum of their semantic distance to MICA (e.g. t_1). Thus, the semantic distance between t_2 and t_3 is quantified as follows:

$$\Delta(t_2, t_3) = \Delta(t_1, t_2) + \Delta(t_1, t_3) \tag{6}$$

Then we integrate the node depth and density factors in DO as conceptual distance factors to quantify the distance defined in equations 5 and 6. Considering there is an anscendant-descendant relationship between t_0 and t_n, and a path from ascendant node t_0 to descendant t_n is $t_0, t_1, \ldots, t_n$, where n is the path length, the semantic distance between t_0 and t_n is redefined as follows:

$$\Delta(t_0, t_n) = \sum_{i=0}^{n-1} D(t_i) \times E(t_i) \times (IC(t_{i+1}) - IC(t_i)) \tag{7}$$

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

$D(t)$ is defined as follows:

$$D(t) = \left(\frac{d(t)+1}{d(t)}\right)^{\alpha} \tag{8}$$

where $d(t)$ denotes the depth of term t in DO, and parameter α controls the contribution of depth factor in equation 7 (In DOSim, we set it to 0.5). In the extreme case, like α is equal to 0, the contribution of depth factor becomes less significant, for $D(t)$ would be equal to 1.

$E(t)$ is defined as follows:

$$E(t) = (1-\beta) \times \frac{\overline{E}}{e(t)} + \beta \tag{9}$$

where $e(t)$ denotes the number of edges that start from t. $\overline{E}$ represents the average density of the DO system, which is equal to the ratio of the total number of edges in DO to the total number of terms in DO(it is equal to 2.85 in DOSim). Parameter β controls the contribution of density factor in equation 7 (In DOSim, we set it to 0.5).

By normalizing the distance defined in equataion 7, we finally get the semantic similarity between term t_1 and t_2 as follows:

$$Sim_{CoutoEnriched}(t_1, t_2) = 1 - \min\left(1, \frac{\Delta(t_1, t_2)}{IC(t_0)}\right) \tag{10}$$

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

2.5 CoutoResnik

In 2005, Couto et.al [5] proposed the GraSM (Graph-based Similarity Measure) approach in their study of correlation between GO and family similarity by introducing the concept of disjunctive common ancestor (DCA).

First of all, we will introduce the concept of disjunctive ancestors of term t, $DisjAnc(t)$. Two ancestors a_1 and a_2 are disjunctive ancestors of term 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 can be formulated as follows:

$$\begin{aligned} DisjAnc\left(t\right) = \{(a_1, a_2) \mid \\ (\exists p : (p \in Paths\left(a_1, t\right)) \wedge (a_2 \notin p)) \wedge \\ (\exists p : (p \in Paths\left(a_2, t\right)) \wedge (a_1 \notin p))\} \end{aligned} \tag{11}$$

Now, we will introduce the concept of common disjunctive ancestors (DCA) of term t_1 and t_2, $DistCommonAnc(t_1, t_2)$. The common disjunctive ancestors (DCA) of term t_1 and t_2 are the most informative common ancestor of disjunctive ancestors of t_1 and t_2. It can be formulated as follows:

$$\begin{aligned} DisjCommonAnc\left(t_1, t_2\right) = \{a_1 \mid \\ a_1 \in CommonAnc\left(t_1, t_2\right) \wedge \\ \forall a_2 : [(a_2 \in CommonAnc\left(t_1, t_2\right)) \wedge (IC\left(a_1\right) \leq IC\left(a_2\right))] \Rightarrow \\ [(a_1, a_2) \in (DisjAnc\left(t_1\right) \cup DisjAnc\left(t_2\right))]\} \end{aligned} \tag{12}$$

GraSM defines the shared information of two terms t_1 and t_2 as the average of the information content of the DCAs which is formulated as follows:

$$Share_{GraSM}\left(t_1, t_2\right) = \overline{\{IC\left(a\right) \mid a \in DisjCommonAnc\left(t_1, t_2\right)\}} \tag{13}$$

The advantage of GraSM is that it can capture more interpretations for the terms t_1 and t_2 in a Directed Acyclic Graph (DAG) ontology, compared with the approach taking MICA as the measure. Also, GraSM could be combined into 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:

$$Sim_{CoutoResnik}\left(t_1, t_2\right) = Share_{GraSM}\left(t_1, t_2\right) \tag{14}$$

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:

$$Sim_{CoutoJC}\left(t_1, t_2\right) = 1 - \min\left(1, IC\left(t_1\right) + IC\left(t_2\right) - 2 \times Share_{GraSM}\left(t_1, t_2\right)\right) \tag{15}$$

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:

$$Sim_{CoutuLin}\left(t_1, t_2\right) = \frac{2 \times Share_{GraSM}\left(t_1, t_2\right)}{IC\left(t_1\right) + IC\left(t_2\right)} \tag{16}$$

2.8 relevance

In 2006, Schlicker et al. [6] developed the relevance similarity measure based on the measure proposed by Lin et al. mentioned above. The characteristic of this measure is that it used the probability of annotation of the MICA as a weighting factor to provide graph placement [7]. It can be formulated as follows:

$$Sim_{relevance}\left(t_1, t_2\right) = Sim_{Lin}\left(t_1, t_2\right) \times \left(1 - p\left(t_{MICA}\right)\right) \tag{17}$$

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

2.9 GIC

GIC (Graph Information Content) is proposed by Presquita et al. [7] and it is an extension of graph-based similarity measures. The formula is shown as follows:

$$Sim_{GIC}\left(t_1, t_2\right) = \frac{\sum\limits_{t \in \left(Ancestor(t_1) \cap Ancestor(t_2)\right)} IC\left(t\right)}{\sum\limits_{t \in \left(Ancestor(t_1) \cup Ancestor(t_2)\right)} IC\left(t\right)} \tag{18}$$

2.10 simIC

The measure of simIC (information coefficient similarity) is similar to the measure of relevance mentioned above. It could effectively integrate the information content and the structural information of terms in a DAG ontology [8]. It is defined as follows:

$$Sim_{simIC}\left(t_1, t_2\right) = Sim_{Lin} \times \left(1 - \frac{1}{1 + IC\left(t_{MICA}\right)}\right) \tag{19}$$

2.11 path

This path-length measure is proposed by Wu et al.[9] in 1994, and it is defined as follows:

$$Sim_{path}\left(t_1, t_2\right) = \frac{1}{p} \tag{20}$$

where p is the number of nodes on the shortest path between two terms in DO.

2.12 lch

The Leacock and Chodorow measure (lch) [10] is computed as follows:

$$Sim_{lch}\left(t_1, t_2\right) = -\log\left(\frac{p}{2 * depth}\right) \tag{21}$$

where p is the number of nodes on the shortest path between two terms t_1 and t_2 in DO and *depth* is the maximum depth of the hierarchy (In DOSim, the maximum *depth* of DO is 90).

2.13 Wang

In 2007, Wang et al. [11] developed a method to measure the semantic similarity of GO terms. In their method, each edge is given a weight according to the relationship type of edges. For a term A, a sub-DAG comprised of term A and all its ancestor terms can be represented as $DAG_A = (A, T_A, E_A)$, where T_A is the ancestor term set 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. defined the semantic contribution of t to A, $D_A(t)$, as the product of all the edge weights in the "best" path from term t to A, where the "best" path is the one that maximizes the product (the semantic contribution of term A to itself is set to 1). It could be represented as follows:

$$\begin{cases} S_A\left(A\right) = 1 \\ S_A\left(t\right) = \max\left\{w_e \times S_A\left(t'\right) \mid t' \in childrenof\left(t\right)\right\} \quad if\ t \neq A \end{cases} \tag{22}$$

where w_e is the semantic contribution factor of edge e ($e \in E_A$). It is set between 0 and 1 according to the type of relationship, e.g., "is-a" or "part-of". In DO, there is only one type of relationship, defined as "is-a", and we set w_e to 0.7 in DOSim.

Then the senmantic similarity between two terms A and B is calculated as follows:

$$Sim_{Wang}\left(A, B\right) = \frac{\sum\limits_{t \in T_A \cap T_B}\left(S_A\left(t\right) + S_B\left(t\right)\right)}{SV\left(A\right) + SV\left(B\right)} \tag{23}$$

where $SV\left(A\right)$ (or $SV\left(B\right)$) is the total semantic contribution to term A (or B) in DAG_A (or DAG_B), which could be calculated as follows:

$$SV\left(A\right) = \sum_{t \in T_A} S_A\left(t\right) \tag{24}$$

3 Measuring the Similarity between Human Genes

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
```

In the following text, we will describe each of the eight methods in an individual sub-section which is named after the parameter name for "similarity" in function *getGeneSim*.

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 disease similarity between two genes g and $g^{'}$ is defined as follows:

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

3.2 mean

It is similar to max method except taking averaged paired DO term similarity as the measure and it is defined as follows:

$$Sim_{mean}\left(g, g^{'}\right) = \frac{\sum_{\substack{i=1,\ldots,n \\ j=1,\ldots,m}} sim\left(t_i, t_j^{'}\right)}{m \times n} \tag{26}$$

3.3 funSimMax

This method is proposed by Schlicker et.al[6] 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:

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

$$columnscore = \frac{1}{m} \sum_{j=1}^{m} \max_{1 \leq i \leq n} s_{ij} \tag{28}$$

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:

$$Sim_{funSimMax}\left(g, g^{'}\right) = \max\{rowScore, columnScore\} \tag{29}$$

3.4 funSimAvg

This method is similar to *funSimMax* except taking the average value between $rowScore$ and $columnScore$ as the measure, which can be formulateds as follows:

$$Sim_{funSimAvg}\left(g, g^{'}\right) = \frac{rowScore + columnScore}{2} \tag{30}$$

3.5 OA

OA(optimal assignment) is proposed by Frohlich et al.[12, 13]. Supposing that two gene g and $g^{'}$ have DO terms annotations $t_1, \ldots, t_n$ and $t_1^{'}, \ldots, t_m^{'}$ respectively, then a way of calculating the similarity of g and $g^{'}$ is to assign each term in the smaller of both lists to exactly one term in the longer one so that the sum of terms similarity is maximized. 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:

$$Sim_{OA}\left(g, g^{'}\right) = \begin{cases} \max_{\pi} \sum_{i=1}^{n} sim\left(t_i, t_{\pi(i)}^{'}\right) & if\ m \leq n \\ \max_{\pi} \sum_{j=1}^{m} sim\left(t_{\pi(j)}, t_j^{'}\right) & otherwise \end{cases} \tag{31}$$

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:

$$Dist_{hausdorff}^{a \to b} = \max_{a \in A} \left\{ \min_{b \in B} \left(Dist\left(a, b\right) \right) \right\} \tag{32}$$

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:

$$Dist_{hausdorff} = \max \left(Dist_{hausdorff}^{a \to b}, Dist_{hausdorff}^{b \to a} \right) \tag{33}$$

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:

$$Sim_{hausdorff}\left(g, g^{'}\right) = \min \left(Sim_{hausdorff}^{a \to b}\left(g, g^{'}\right), Sim_{hausdorff}^{b \to a}\left(g, g^{'}\right) \right) \tag{34}$$

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

$$Sim_{hausdorff}^{a \to b}\left(g, g^{'}\right) = \min_{a \in A} \left\{ \max_{b \in B} \left(Sim\left(a, b\right) \right) \right\} \tag{35}$$

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:

$$\phi\left(g\right) = \left(sim_{max}\left(g, p_1\right), \ldots, sim_{max}\left(g, p_n\right)\right)^T \tag{36}$$

Then the similarity between genes g and $g^{'}$ is the dot product of their feature vector defined by equation 36 after normalizing which can be formulated as:

$$Sim_{dot}\left(g, g^{'}\right) = \frac{< \phi\left(g\right), \phi(g^{'}) >}{\sqrt{< \phi\left(g\right), \phi(g) > \times < \phi\left(g^{'}\right), \phi(g^{'}) >}} \tag{37}$$

3.8 Wang

It is proposed by Wang et.al [11] using 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:

$$Sim_{wang}\left(g, g^{'}\right) = \frac{\sum\limits_{1 \le i \le n} Sim\left(t_i, DO_2\right) + \sum\limits_{1 \le j \le m} Sim\left(t_j^{'}, DO_1\right)}{n + m} \tag{38}$$

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 Geting Information of DO

The DO 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 shown 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 offsprings 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

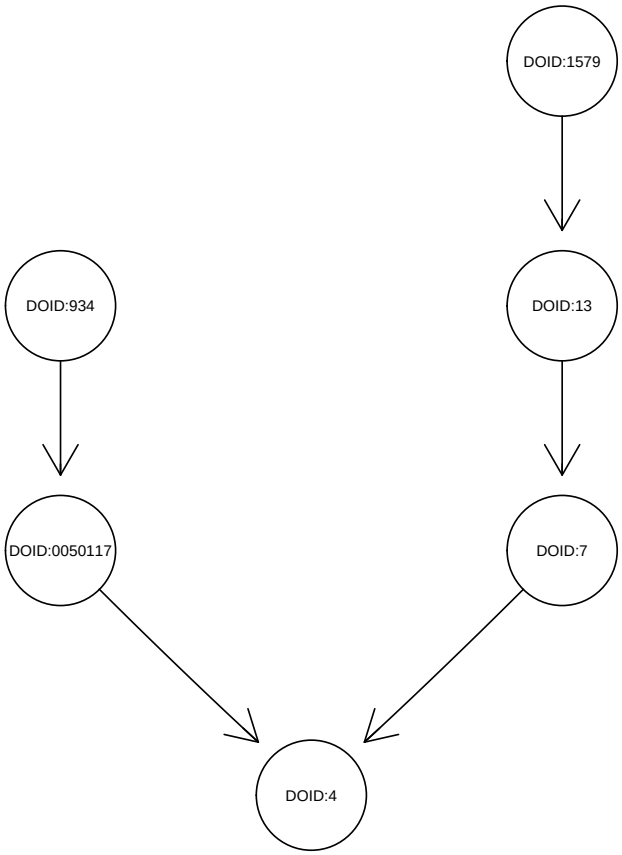
```
> terms <- c("DOID:1579")
> getDoAnno(terms)

$`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 **hypergeometric 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 = "fisher", filter = 50, cutoff = 0.01,
+     adjustp = "fdr")
```

```
                DOID              Term genenum1 genenum2     odds       pvalue
DOID:14330 DOID:14330 Parkinson disease      101        5 18.06832 1.402219e-05
             adjustedP
DOID:14330 0.0009394866
```

References

[1] 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.

[2] 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.

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

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

[5] 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.

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

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

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

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

[10] 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.

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

[12] 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.

[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.