How To Use DOSim

Jiang Li

April 27, 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, information content(IC) based is the most popular one. In our package, we provide 13 different methods to get DO terms similarity,including IC based and graph 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  0.9766585 0.8042373 0.9610522 0.4452106
DOID:1117 0.8042373 0.9398034 0.4252460 0.5429404
DOID:2313 0.9610522 0.4252460 0.9742015 0.4538407
DOID:2040 0.4452106 0.5429404 0.4538407 0.7781327
```

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

2.1 Resnik

Method *Resnik* [2] is IC based,the similarity between *term1* and *term2* is the maximum IC of their common ancestors. Defined as

$$IC(term1, term2) = \max_{t \in S(term1,term2)} [IC(t)] = IC_{ms}$$

where $S(term1, term2)$ is the set of terms that subsume both $term1$ and $term2$.

2.2 JiangConrath

In 1997, Jay J. Jiang and David W. Conrath[3] proposed a new method and the formular is below:

$$IC(term1, term2) = 1 - \min(1, IC(term1) - 2IC_{ms} + IC(term2))$$

where IC_{ms} is the similarity defiend by Resnik.

2.3 Lin

The formula for Lin[4] is below:

$$IC(term1, term2) = \frac{2IC_{ms}}{IC(term1) + IC(term2)}$$

where IC_{ms} is the similarity defiend by Resnik.

2.4 CoutoEnriched

These method is proposed by Couto in 2003[5],please see the original paper for detail.

2.5 CoutoResnik

It is similar to *Resnik* ,but instead of using common ancestor, the similarity of $term1$ and $term2$ is the maximun IC of all the commom **disjunctive ancestors** of $term1$ and $term2$[6]. It is defined as:

$$IC(term1, term2) = \max_{t \in CommonDisjAnc(term1,term2)} [IC(t)] = IC_{share}$$

where $CommonDisjAnc(term1, term2)$ is the set of common disjunctive ancestors of $term1$ and $term2$.

2.6 CoutoJiangConrath

Similar to JiangConrath,use the Couto's[6] concept and defined as :

$$IC(term1, term2) = 1 - \min(1, IC(term1) - 2IC_{share} + IC(term2))$$

where IC_{share} is the similarity defiend by CoutoResnik.

2.7 CoutoLin

Similar to Lin,use the Couto's[6] concept and defined as :

$$IC(term1, term2) = \frac{2IC_{share}}{IC(term1) + IC(term2)}$$

where IC_{share} is the similarity defiend by CoutoResnik.

2.8 relevance

Proposed by Schlicker[7] in 2006.

$$IC(term1, term2) = Sim_{Lin} * (1 - e^{-IC_{ms}})$$

where Sim_{Lin} is the similarity defined by Lin and IC_{ms} for Resnik.

2.9 GIC

Proposed by Pesquita[8] in 2007.

$$IC(term1, term2) = \frac{\sum\limits_{t \in (Ancestor(term1) \cap Ancestor(term2))} IC(t)}{\sum\limits_{t \in (Ancestor(term1) \cup Ancestor(term2))} IC(t)}$$

where $Ancestor(t)$ is the set of all ancestor terms of term t

2.10 simIC

Proposed by Li[9] in 2009.

$$IC(term1, term2) = Sim_{Lin} * (1 - \frac{1}{1 + IC_{ms}})$$

where Sim_{Lin} is the similarity defined by Lin and IC_{ms} for Resnik.

2.11 path

This method is not IC based and first proposed by Wu Z[10] in 1994 and mentiond in Pedersen's[11] article in 2007.

$$IC(term1, term2) = \frac{1}{p}$$

where p is the number of nodes on the shortest path between $term1$ and $term2$.

2.12 lch

This method is also not IC based and first proposed by Leacock C[12] in 1998 and mentiond in Pedersen's[11] article in 2007.

$$IC(term1, term2) = -\log(\frac{p}{2 * depth})$$

where p is the number of nodes on the shortest path between $term1$ and $term2$ and $depth$ is the maximum depth of the hierarchy.

2.13 Wang

Proposed by Wang[13] in 2007 and see the original paper for detail.

$$Sim(term1, term2) = \frac{\sum\limits_{t \in T_{term1} \cap T_{term2}} (S_{term1}(t) + S_{term2}(t))}{SV(term1) + SV(term2)}$$

where $S_{term1}(t)$ is the $S - value$ of term t related to term $term1$. In DO, term $term1$ can be represented as $DAG_{term1} = (term1, T_{term1}, E_{term1})$ where T_{term1} is the set of DO terms in DAG_{term1},including term $term1$ and all of its ancestor terms in the DO graph,and E_{term1} is the set of edeges connecting the DO terms in DAG_{term1}.And for any term t in $DAG_{term1} = (term1, T_{term1}, E_{term1})$,its S-value is defined as:

$$\begin{cases} S_{term1}(term1)=1 \\ S_{term1}(t)=\max\{w_e*S_{term1}(t^{'})|t^{'}\in childrenof(t)\} \quad if \ t\neq A \end{cases}$$

where w_e is the semantic contribution factor for edge $e \in E_{term1}$ linking term t with its child term $t^{'}$.After obtaining the S-values for all terms in DAG_{term1}, we calculate the semantic value of DO term $term1$,$SV(term1)$,as:

$$SV(term1) = \sum_{t \in T_{term1}} S_{term1}(t)$$

3 Calculate Genes Similarity

Genes similarity is calculate based on their annotated DO terms similarity.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 = "funSimMax", similarityTerm = "Lin")

> gsim
```

```
            10003 10008       10015       10042       10036
10003 1.000000000     0 0.003439812 0.002969545 0.281067587
10008 0.000000000     1 0.000000000 0.000000000 0.000000000
10015 0.003439812     0 1.000000000 0.001945925 0.002137409
10042 0.002969545     0 0.001945925 1.000000000 0.001945925
10036 0.281067587     0 0.002137409 0.001945925 1.000000000
```

Here we define some formula and detail information for each method is described below. Assume $gene1$ have m DO annoated($DO_1 = \{do_{11}, do_{12}, \ldots, do_{1m}\}$) and $gene2$ have n DO annotated($DO_2 = \{do_{21}, do_{22}, \ldots, do_{2n}\}$). We define Sim_{matrix} is an $m \times n$ matrix of any pairwise DO terms similarity from DO_1 to DO_2.

$$Sim_{matrix} = \left\{ \begin{array}{cccc} sim_{11} & sim_{12} & \cdots & sim_{1n} \\ sim_{21} & sim_{22} & \cdots & sim_{2n} \\ \multicolumn{4}{c}{\dots\dots\dots\dots\dots\dots\dots\dots} \\ sim_{m1} & sim_{m2} & \cdots & sim_{mn} \end{array} \right\}$$

3.1 max

The maximum similarity between any two DO terms.

$$Sim(gene1, gene2) = \max(Sim_{matrix})$$

3.2 mean

The average similarity between any two DO terms

$$Sim(gene1, gene2) = mean(Sim_{matrix})$$

3.3 funSimMax

The average of best matching DO term similarities. Take the maximum of the scores achieved by assignments of DO terms from gene 1 to gene 2 and vice versa.[14]

$$Sim(gene1, gene2) = \max(rowMax, colMax)$$

where $rowMax$ is the average score of each row's maximum score, and same for $colMax$.

3.4 funSimAvg

The average of best matching DO term similarities. Take the average of the scores achieved by assignments of DO terms from gene 1 to gene 2 and vice versa. [14]

$$Sim(gene1, gene2) = \frac{rowMax + colMax}{2}$$

where $rowMax$ is the average score of each row's maximum score, and same for $colMax$.

3.5 OA

The optimal assignment (maximally weighted bipartite matching) of DO terms associated to the gene having fewer annotation to the DO terms of the other gene.[15]. See the original paper for details.

3.6 hausdorff

The translation of the Hausdorff distance between two sets:[16] Let A and B be the two sets of DO terms associated to two genes($gene1$ and $gene2$). Then

$$Sim(gene1, gene2) = \min\left(\min\left(\max_{x \in A}(x, y)\right), \min\left(\max_{y \in B}(x, y)\right)\right)$$

3.7 dot

The dot product between feature vectors describing the absence/presence of each DO term. The absence of each DO term is weighted by its information content. Depending on the type of later normalization one can arrive at the cosine similarity (method=”sqrt”) or at the Tanimoto coefficient (method=”Tanimoto”).[17].See the original paper for details.

3.8 Wang

Propose by Wang in 2007.[13]. Give two genes $gene1$ and $gene2$ annotated by DO term sets $DO_1 = \{do_{11}, do_{12}, \ldots, do_{1m}\}$ and $DO_2 = \{do_{21}, do_{22}, \ldots, do_{2n}\}$ respectively, we define their similarity as:

$$Sim(gene1, gene2) = \frac{\sum\limits_{1 \leq i \leq m} Sim(do_{1i}, DO_2) + \sum\limits_{1 \leq j \leq n} Sim(do_{2j}, DO_1)}{m + n}$$

where $Sim(do_{1i}, DO_2) = \max\limits_{do_j \in DO_2}(sim(do_{1i}, do_j))$

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[18], 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 example 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:95"

$`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:2040"    "DOID:4"       "DOID:95"

$`DOID:1579`
[1] "DOID:8"  "DOID:2"  "DOID:5"  "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:5460"  "DOID:874"   "DOID:12017" "DOID:14474" "DOID:10531"
 [6] "DOID:13277" "DOID:10510" "DOID:13275" "DOID:13815" "DOID:12607"
[11] "DOID:5461"  "DOID:12888" "DOID:14473" "DOID:10508" "DOID:10509"
[16] "DOID:14338" "DOID:14475" "DOID:13272" "DOID:12608" "DOID:14319"
[21] "DOID:13278" "DOID:13273" "DOID:10457" "DOID:13164" "DOID:12019"
[26] "DOID:11742" "DOID:14472" "DOID:14477" "DOID:11741" "DOID:10532"
[31] "DOID:13167" "DOID:10527" "DOID:14476" "DOID:13276" "DOID:13274"
[36] "DOID:13165" "DOID:873"   "DOID:12375"

$`DOID:550`
[1] "DOID:549" "DOID:551" "DOID:554" "DOID:553"
```

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:623"     "DOID:1329"    "DOID:5064"    "DOID:1301"    "DOID:2950"
 [6] "DOID:937"     "DOID:2295"    "DOID:0050079" "DOID:13801"   "DOID:1274"
[11] "DOID:3294"    "DOID:2940"    "DOID:2941"    "DOID:1304"    "DOID:4121"
[16] "DOID:6297"    "DOID:1310"    "DOID:4146"    "DOID:2931"    "DOID:2932"
[21] "DOID:2947"    "DOID:1885"    "DOID:1331"    "DOID:2937"    "DOID:1385"
[26] "DOID:10533"
```

```
$`DOID:1579`
 [1] "DOID:11023" "DOID:1585"  "DOID:12118" "DOID:12117" "DOID:3224" 
 [6] "DOID:974"   "DOID:11091" "DOID:13016" "DOID:6144"  "DOID:766"  
[11] "DOID:550"  
```

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] "Virus diseases"

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

4.6 getDoAnno

Get gene list associated to each DO term

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

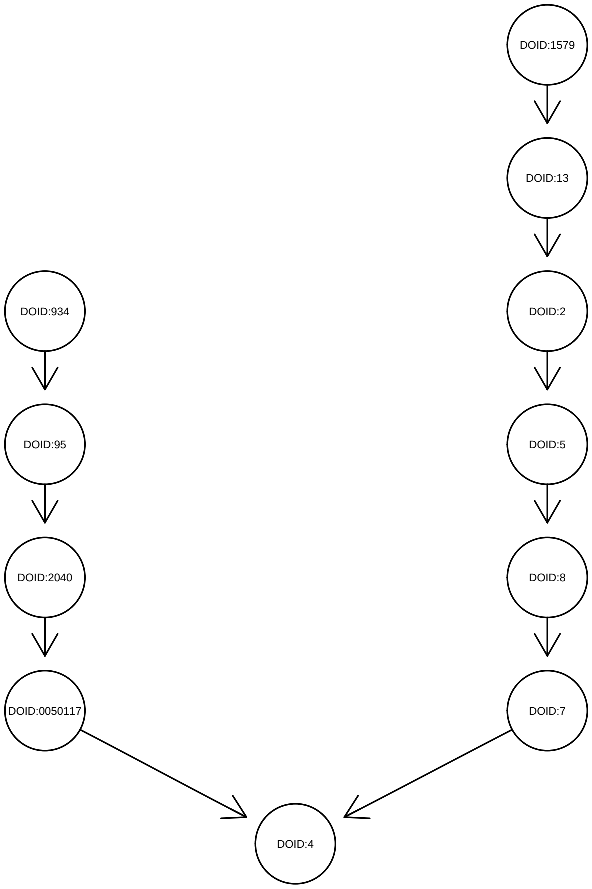
```
$`DOID:934`
 [1] "3596"   "943"    "3802"   "941"    "2159"   "7098"   "5806"   "3837"  
 [9] "348"    "3659"   "3665"   "3566"   "29110"  "60489"  "939"    "282618"
[17] "3105"   "10859"  "4599"   "5133"   "3439"   "3824"   "8797"   "3491"  
[25] "1231"   "3821"   "5322"   "57062"  "3661"   "1487"   "3567"   "796"   
[33] "708"    "2022"   "103"    "3565"   "4000"   "3576"   "4277"   "5058"  
[41] "3553"   "6504"   "325"    "942"    "3627"   "64135"  "3554"   "3456"  
[49] "332"    "3998"   "3586"   "3106"   "3265"   "282616" "929"    "59067" 
[57] "5932"   "3676"   "3620"   "5371"   "10010"  "842"    "4153"   "1616"  
[65] "5366"   "3438"   "1234"   "10344"  "4001"   "3609"   "1147"   "57506" 
[73] "10219"  "3838"   "7293"   "6041"   "10673" 

$`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] 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.

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

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

[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] Pedersen T, Pakhomov SV, Patwardhan S, Chute CG: **Measures of semantic similarity and relatedness in the biomedical domain**. *Journal of Biomedical Informatics* 2007, **40**(3):288 – 299.

[12] 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.

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

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

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

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

[17] M Mistry PP: **Gene Ontology term overlap as a measure of gene functional similarity**. *BMC Bioinformatics* 2008, :9:327.

[18] 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.