# Disease Ontology Semantic and Enrichment analysis

Guangchuang Yu, Li-Gen Wang

October 13, 2014

**Abstract**

Disease Ontology (DO) aims to provide an open source ontology for the integration of biomedical data that is associated with human disease. We developed *DOSE* package to promote the investigation of diseases. *DOSE* provides five methods including Resnik, Lin, Jiang, Rel and Wang for measuring semantic similarities among DO terms and gene products; Hypergeometric model and gene set enrichment analysis were also implemented for extracting disease association insight from genome wide expression profiles.

## Contents

# 1 Introduction

Public health is an important driving force behind biological and medical research. A major challenge of the post-genomic era is bridging the gap between fundamental biological research and its clinical applications. Recent research has increasingly demonstrated that many seemingly dissimilar diseases have common molecular mechanisms. Understanding similarities among disease aids in early diagnosis and new drug development.

Formal knowledge representation of gene-disease association is demanded for this purpose. Ontologies, such as Gene Ontology, have been successfully applied to represent biological knowledge, and many related techniques have been adopted to extract information. Disease Ontology (DO) [1] was developed to create a consistent description of gene products with disease perspectives, and is essential for supporting functional genomics in disease context. Accurate disease descriptions can discover new relationships between genes and disease, and new functions for previous uncharacteried genes and alleles.

Unlike other clinical vocabularies that defined disease related concepts disparately, DO is organized as a directed acyclic graph, laying the foundation for quantitative computation of disease knowledge. The application of disease ontology is in its infancy, lacking programs for mining DO knowledge automatically.

Here, we present an R package *DOSE* for analyzing semantic similarities among DO terms and gene products annotated with DO terms, and extracting disease association insight from genome wide expression profiles.

Four information content (IC)-based methods and one graph structure-based method were implemented for measuring semantic similarity. Hypergeometric test and Gene Set Enrichment Analysis were implemented for extracting biological insight.

To start with *DOSE* package, type following code below:

```
library(DOSE)
help(DOSE)
```

# 2 DO term semantic similarity measurement

Four methods determine the semantic similarity of two terms based on the Information Content of their common ancestor term were proposed by Resnik [2], Jiang [3], Lin [4] and Schlicker [5]. Wang [6] presented a method to measure the similarity based on the graph structure. Each of these methods has its own advantage and weakness. *DOSE* implemented all these methods to compute semantic similarity among DO terms and gene products. We have developed another package *GOSemSim* [7] to explore the functional similarity at GO perspective, including molecular function (MF), biological process (BP) and cellular component (CC).

## 2.1 Information content-based method

Information content (IC) is defined as the negative logarithm of the frequency of each term occurs in the corpus of DO annotation.

The frequency of a term t is defined as:

$$p(t) = \frac{n_{t'}}{N} | t' \in \{t,\ children\ of\ t\}$$

where $n_{t'}$ is the number of term t', and N is the total number of terms in DO corpus.

Thus the information content is defined as:

$$IC(t) = -\log(p(t))$$

IC-based methods calculate similarity of two DO terms based on the information content of their closest common ancestor term, which was also called most informative information ancestor (MICA).

### 2.1.1 Resnik method

The Resnik method is defined as:

$$sim_{Resnik}(t_1, t_2) = IC(MICA)$$

### 2.1.2 Lin method

The Lin method is defined as:

$$sim_{Lin}(t_1, t_2) = \frac{2IC(MICA)}{IC(t_1) + IC(t_2)}$$

### 2.1.3 Rel method

The Relevance method, which was proposed by Schlicker, combine Resnik's and Lin's method and is defined as:

$$sim_{Rel}(t_1, t_2) = \frac{2IC(MICA)(1 - p(MICA))}{IC(t_1) + IC(t_2)}$$

### 2.1.4 Jiang method

The Jiang and Conrath's method is defined as:

$$sim_{Jiang}(t_1, t_2) = 1 - \min(1, IC(t_1) + IC(t_2) - 2IC(MICA))$$

## 2.2 Graph-based method

Graph-based methods using the topology of DO graph structure to compute semantic similarity. Formally, a DO term A can be represented as $DAG_A = (A, T_A, E_A)$ where $T_A$ is the set of DO terms in $DAG_A$, including term A and all of its ancestor terms in the DO graph, and $E_A$ is the set of edges connecting the DO terms in $DAG_A$.

### 2.2.1 Wang method

To encode the semantic of a DO term in a measurable format to enable a quantitative comparison, Wang firstly defined the semantic value of term A as the aggregate contribution of all terms in $DAG_A$ to the semantics of term A, terms closer to term A in $DAG_A$ contribute more to its semantics. Thus, defined the contribution of a DO term *t* to the semantic of DO term A as the S-value of DO term *t* related to term A. For any of term *t* in $DAG_A$, its S-value related to term A, $S_A(t)$ is defined as:

$$\begin{cases} S_A(A) = 1 \\ S_A(t) = \max\{w_e \times S_A(t') | t' \in children\ of(t)\}\ if\ t \neq A \end{cases}$$

where $w_e$ is the semantic contribution factor for edge $e \in E_A$ linking term *t* with its child term *t'*. Term A contributes to its own is defined as one. After obtaining the S-values for all terms in $DAG_A$, the semantic value of DO term A, SV(A), is calculated as:

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

Thus given two DO terms A and B, the semantic similarity between these two terms is defined as:

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

where $S_A(t)$ is the S-value of DO term *t* related to term A and $S_B(t)$ is the S-value of DO term *t* related to term B.

## 2.3 doSim function

In *DOSE*, we implemented all these IC-based and graph-based methods. `doSim` can calculate semantic similarity between two DO terms and two set of DO terms.

```
data(DO2EG)
set.seed(123)
a <- sample(names(DO2EG), 10)
a

##  [1] "DOID:1407"  "DOID:5844"  "DOID:2034"  "DOID:8432"  "DOID:9146"
##  [6] "DOID:10584" "DOID:3209"  "DOID:848"   "DOID:3341"  "DOID:2512"

b <- sample(names(DO2EG), 5)
b

## [1] "DOID:9409"  "DOID:2481"  "DOID:4465"  "DOID:3498"  "DOID:11252"

doSim(a[1], b[1], measure = "Wang")

## Loading required package:  DO.db

## [1] 0.113

doSim(a[1], b[1], measure = "Resnik")

## [1] 0.0763

doSim(a[1], b[1], measure = "Lin")

## [1] 0.1

s <- doSim(a, b, measure = "Wang")
s

##            DOID:9409 DOID:2481 DOID:4465 DOID:3498 DOID:11252
## DOID:1407     0.1133    0.0860    0.0152    0.0152     0.0819
## DOID:5844     0.1490    0.0783    0.0280    0.0280     0.1156
## DOID:2034     0.1735    0.4755    0.0368    0.0368     0.1388
## DOID:8432     0.1735    0.1000    0.0368    0.0368     0.4220
## DOID:9146     0.0714    0.0412    0.0368    0.0368     0.0571
## DOID:10584    0.1211    0.0987    0.0180    0.0180     0.0893
## DOID:3209     0.1490    0.0783    0.0280    0.0280     0.1156
## DOID:848      0.1735    0.1000    0.0368    0.0368     0.1388
## DOID:3341     0.1490    0.0783    0.0280    0.0280     0.1156
## DOID:2512     0.0714    0.0412    0.0368    0.0368     0.0571
```

`doSim` requires three parameter *DOID1*, *DOID2* and *measure*. *DOID1* and *DOID2* should be a vector of DO terms, while *measure* should be one of Resnik, Jiang, Lin, Rel, and Wang.

We also implement a plot function `simplot` to visualize the similarity result.

```
simplot(s, color.low = "white", color.high = "red",
    labs = TRUE, digits = 2, labs.size = 5, font.size = 14,
    xlab = "", ylab = "")

## Using ID as id variables
```

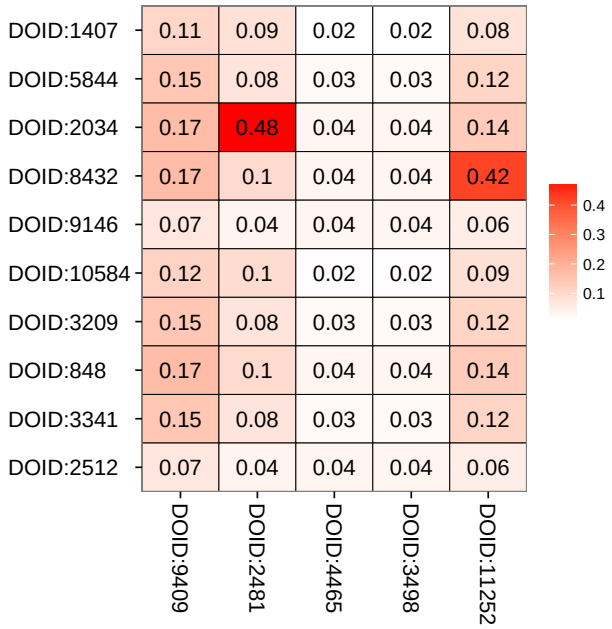


Figure 1: Visualizing similarity matrix.

Parameter *color.low* and *colow.high* are used to setting the color gradient; *labs* is a logical parameter indicating whether to show the similarity values or not, *digits* to indicate the number of decimal places to be used and *labs.size* setting the size of similarity values; *font.size* setting the font size of axis and label of the coordinate system.

# 3 Gene semantic similarity measurement

On the basis of semantic similarity between DO terms, *DOSE* can also compute semantic similarity among gene products.

Suppose we have gene $g_1$ annotated by DO term set $DO_1 = \{do_{11}, do_{12} \cdots do_{1m}\}$

and $g_2$ annotated by $DO_2 = \{do_{21}, do_{22} \cdots do_{2n}\}$, *DOSE* implemented four methods which called `max`, `avg`, `rcmax` and `BMA` to combine semantic similairty scores of multiple DO terms.

## 3.1 Combine method

### 3.1.1 max

The `max` method calculates the maximum semantic similarity score over all pairs of DO terms between these two DO term sets.

$$sim_{max}(g_1, g_2) = \max_{1 \le i \le m, 1 \le j \le n} sim(do_{1i}, do_{2j})$$

### 3.1.2 avg

The `avg` calculates the average semantic similarity score over all pairs of DO terms.

$$sim_{avg}(g_1, g_2) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} sim(do_{1i}, do_{2j})}{m \times n}$$

### 3.1.3 rcmax

Similarities among two sets of DO terms form a matrix, the `rcmax` method uses the maximum of RowScore and ColumnScore as the similarity, where RowScore (or ColumnScore) is the average of maximum similarity on each row (or column).

$$sim_{rcmax}(g_1, g_2) = \max(\frac{\sum_{i=1}^{m} \max_{1 \le j \le n} sim(do_{1i}, do_{2j})}{m}, \frac{\sum_{j=1}^{n} \max_{1 \le i \le m} sim(do_{1i}, do_{2j})}{n})$$

### 3.1.4 BMA

The `BMA` method, used the best-match average strategy, calculates the average of all maximum similarities on each row and column, and is defined as:

$$sim_{BMA}(g_1, g_2) = \frac{\sum_{1=i}^{m} \max_{1 \le j \le n} sim(do_{1i}, do_{2j}) + \sum_{1=j}^{n} \max_{1 \le i \le m} sim(do_{1i}, do_{2j})}{m + n}$$

## 3.2 geneSim function

In *DOSE*, we implemented `geneSim` to measure semantic similarities among genes.

```r
data(EG2DO)
g1 <- sample(names(EG2DO), 5)
g1
```

```
## [1] "84842" "2521"  "10592" "3069"  "91746"
```

```r
g2 <- sample(names(EG2DO), 4)
g2
```

```
## [1] "84289" "6045"  "56999" "9869"
```

```r
geneSim(g1[1], g2[1], measure = "Wang", combine = "BMA")
```

```
## [1] 0.057
```

```r
gs <- geneSim(g1, g2, measure = "Wang", combine = "BMA")
```

```
## Warning in geneID1 == geneID2:  longer object length is not a multiple
of shorter object length
## Warning in geneID1 == geneID2:  longer object length is not a multiple
of shorter object length
## Warning in geneID1 == geneID2:  longer object length is not a multiple
of shorter object length
## Warning in geneID1 == geneID2:  longer object length is not a multiple
of shorter object length
## Warning in geneID1 == geneID2:  longer object length is not a multiple
of shorter object length
```

```r
gs
```

```
##       84289  6045 56999  9869
## 84842 0.057 0.135 0.356 0.107
## 2521  0.573 0.253 0.515 0.575
## 10592 0.057 0.187 0.296 0.139
## 3069  0.573 0.517 1.000 1.000
## 91746 0.573 0.308 0.533 0.545
```

`geneSim` requires four parameter *geneID1*, *geneID2*, *measure* and *combine*. *geneID1* and *geneID2* should be a vector of entrez gene IDs, *measure* should be one of Resnik, Jiang, Lin, Rel, and Wang, while *combine* should be one of max, avg, rcmax and BMA as described previously.

The `simplot` works well with both the output of `doSim` and `geneSim`.

# 4 DO term enrichment analysis

## 4.1 Hypergeometric model

Enrichment analysis [8] is a widely used approach to identify biological themes. Here we implement hypergeometric model to assess whether the number of selected genes associated with disease is larger than expected.

To determine whether any DO terms annotate a specified list of genes at frequency greater than that would be expected by chance, *DOSE* calculates a p-value using the hypergeometric distribution:

$$p = 1 - \sum_{i=0}^{k-1} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}}$$

In this equation, *N* is the total number of genes in the background distribution, *M* is the number of genes within that distribution that are annotated (either directly or indirectly) to the node of interest, *n* is the size of the list of genes of interest and *k* is the number of genes within that list which are annotated to the node. The background distribution by default is all the genes that have DO annotation.

P-values were adjusted for multiple comparison, and q-values were also calculated for FDR control.

## 4.2 enrichDO function

*DOSE* provides an example dataset *geneList* which was derived from R package *breastCancerMAINZ* that contained 200 samples, including 29 samples in grade I, 136 samples in grade II and 35 samples in grade III. We computed the ratios of geometric means of grade III samples versue geometric means of grade I samples. Logarithm of these ratios (base 2) were stored in *geneList* dataset.

In the following example, we selected fold change above 1 as the differential genes and analyzing their disease association.

```
data(geneList)
gene <- names(geneList)[abs(geneList) > 1]
head(gene)

## [1] "4312"  "8318"  "10874" "55143" "55388" "991"

x <- enrichDO(gene, ont = "DOLite", pvalueCutoff = 0.05,
    pAdjustMethod = "BH", universe = names(geneList),
    minGSSize = 5, readable = FALSE)
head(summary(x))

##                   ID                Description GeneRatio  BgRatio   pvalue
```

```
## DOLite:64   DOLite:64             Atherosclerosis    47/493 192/3466 6.92e-05
## DOLite:548 DOLite:548           Vascular disease    12/493  28/3466 2.08e-04
## DOLite:449 DOLite:449 Protein-energy malnutrition     6/493   9/3466 4.61e-04
## DOLite:100 DOLite:100                      Cancer   123/493 668/3466 4.73e-04
## DOLite:450 DOLite:450                  Proteinuria     9/493  20/3466 8.63e-04
## DOLite:38   DOLite:38             Advanced cancer     6/493  10/3466 1.02e-03
##            p.adjust  qvalue
## DOLite:64   0.00734 0.00532
## DOLite:548  0.01103 0.00800
## DOLite:449  0.01254 0.00909
## DOLite:100  0.01254 0.00909
## DOLite:450  0.01794 0.01300
## DOLite:38   0.01794 0.01300
##
## DOLite:64
## DOLite:548
## DOLite:449
## DOLite:100 4312/10874/2305/4605/9833/10403/6241/9787/11065/4751/890/10232/4085/5
## DOLite:450
## DOLite:38
##            Count
## DOLite:64     47
## DOLite:548    12
## DOLite:449     6
## DOLite:100   123
## DOLite:450     9
## DOLite:38      6
```

The `enrichDO` requires an entrezgene ID vector as input, mostly is the differential gene list of gene expression profile studies. The *ont* parameter can be ”DO” or ”DOLite”, DOLite [9] was constructed to aggregate the redundant DO terms; *pvalueCutoff* setting the cutoff value of p value and p value adjust; pAdjustMethod setting the p value correction methods, include the Bonferroni correction (”bonferroni”), Holm (”holm”), Hochberg (”hochberg”), Hommel (”hommel”), Benjamini & Hochberg (”BH”) and Benjamini & Yekutieli (”BY”).

The *universe* setting the background gene universe for testing. If user do not explicitly setting this parameter, `enrichDO` will set the universe to all human genes that have DO annotation.

The *minGSSize* indicates that only those DO terms that have more than *minGSSize* genes annotated will be tested.

The *readable* is a logical parameter, indicates whether the entrezgene IDs will mapping to gene symbols or not.

We also implement `setReadable` function that helps the user to convert entrezgene IDs to gene symbols.

```
x <- setReadable(x)

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

head(summary(x))

##                    ID                 Description GeneRatio  BgRatio   pvalue
## DOLite:64   DOLite:64             Atherosclerosis    47/493 192/3466 6.92e-05
## DOLite:548 DOLite:548           Vascular disease    12/493  28/3466 2.08e-04
## DOLite:449 DOLite:449 Protein-energy malnutrition     6/493   9/3466 4.61e-04
## DOLite:100 DOLite:100                      Cancer   123/493 668/3466 4.73e-04
## DOLite:450 DOLite:450                 Proteinuria     9/493  20/3466 8.63e-04
## DOLite:38   DOLite:38             Advanced cancer     6/493  10/3466 1.02e-03
##            p.adjust  qvalue
## DOLite:64   0.00734 0.00532
## DOLite:548  0.01103 0.00800
## DOLite:449  0.01254 0.00909
## DOLite:100  0.01254 0.00909
## DOLite:450  0.01794 0.01300
## DOLite:38   0.01794 0.01300
##
## DOLite:64
## DOLite:548
## DOLite:449
## DOLite:100 MMP1/NMU/FOXM1/MYBL2/MELK/NDC80/RRM2/DLGAP5/UBE2C/NEK2/CCNA2/MSLN/MAD
## DOLite:450
## DOLite:38
##            Count
## DOLite:64     47
## DOLite:548    12
## DOLite:449     6
## DOLite:100   123
## DOLite:450     9
## DOLite:38      6
```

## 4.3 Visualze enrichment result

We also implement a bar plot and category-gene-network for visualization. It is very common to visualize the enrichment result in bar or pie chart. We believe the pie chart is misleading and only provide bar chart.

```
barplot(x)
```

In order to consider the potentially biological complexities in which a gene may belong to multiple annotation categories, we developed `cnetplot` function to extract the complex association between genes and diseases.

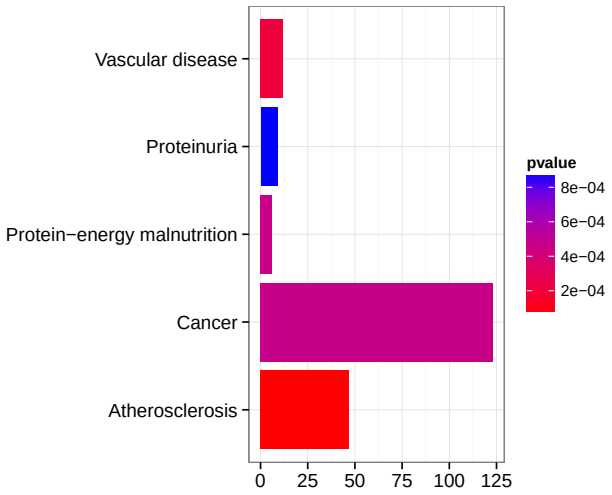


Figure 2: barplot of DO enrichment result.

```
cnetplot(x, categorySize = "pvalue", foldChange = geneList)
```

## 4.4 Disease association comparison

We have developed an R package *clusterProfiler* [10] for comparing biological themes among gene clusters. *DOSE* works fine with *clusterProfiler* and can compare biological themes at disease perspective.

```
require(clusterProfiler)
data(gcSample)
cdo <- compareCluster(gcSample, fun = "enrichDO")
plot(cdo)
```

# 5 Gene set enrichment analysis

## 5.1 GSEA algorithm

A common approach in analyzing gene expression profiles was identifying differential expressed genes that are deemed interesting. The DO term enrichment analysis we demonstrated previous were based on these differential expressed genes. This approach will find genes where the difference is large, but it will not detect a situation where the difference is small, but evidenced in coordinated way in a set of related genes. Gene Set Enrichment Analysis (GSEA) [11] directly addresses this limitation. All genes can be used in GSEA; GSEA aggregates the per gene statistics across genes within a gene set, therefore making it possible

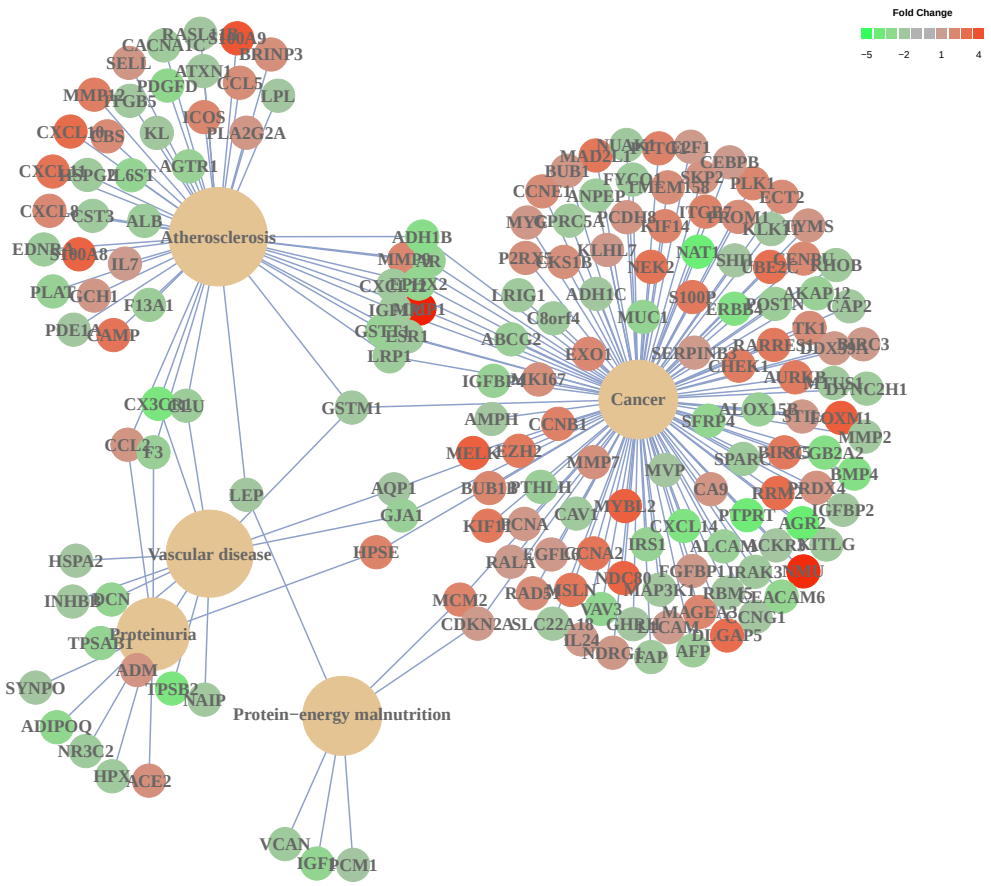


Figure 3: cnetplot of DO enrichment result.

to detect situations where all genes in a predefined set change in a small but coordinated way. Since it is likely that many relevant phenotypic differences are manifested by small but consistent changes in a set of genes.

Genes are ranked based on their phenotypes. Given a priori defined set of gens *S* (e.g., genes shareing the same *DO* or *DOLite* category), the goal of GSEA is to determine whether the members of *S* are randomly distributed throughout the ranked gene list (*L*) or primarily found at the top or bottom.

There are three key elements of the GSEA method:

- Calculation of an Enrichment Score.
  The enrichment score (*ES*) represent the degree to which a set *S* is over-represented at the top or bottom of the ranked list *L*. The score is calculated by walking down the list *L*, increasing a running-sum statistic when we encounter a gene in *S* and decreasing when it is not. The magnitude of the

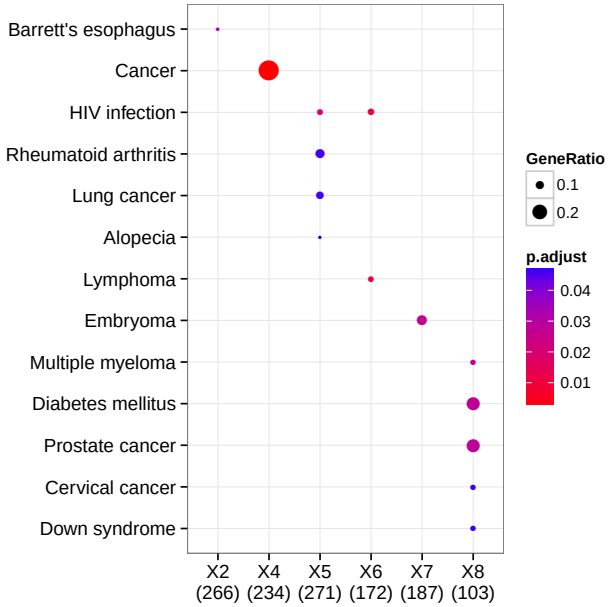


Figure 4: DOSE with clusterProfiler.

increment depends on the gene statistics (e.g., correlation of the gene with phenotype). The *ES* is the maximum deviation from zero encountered in the random walk; it corresponds to a weighted Kolmogorov-Smirnov-like statistic [11].

- Esimation of Significance Level of *ES*.
  The *p-value* of the *ES* is calculated using permutation test. Specifically, we permute the gene labels of the gene list *L* and recompute the *ES* of the gene set for the permutated data, which generate a null distribution for the *ES*. The *p-value* of the observed ES is then calculated relative to this null distribution.

- Adjustment for Multiple Hypothesis Testing.
  When the entire *DO* or *DOLite* gene sets is evaluated, *DOSE* adjust the estimated significance level to account for multiple hypothesis testing and also *q-values* were calculated for FDR control.

## 5.2 gseAnalyzer fuction

In *DOSE*, we implemented GSEA algorithm proposed by Subramanian [11] in `gseAnalyzer` function.

In the following example, in order to speedup the compilation of this document, only gene sets with size above 120 were tested and only 100 permutations were performed.

```
y <- gseAnalyzer(geneList, setType = "DOLite", nPerm = 100,
    minGSSize = 120, pvalueCutoff = 0.05, pAdjustMethod = "BH",
    verbose = FALSE)
res <- summary(y)
head(res)
```

```
##                    ID         Description setSize enrichmentScore pvalue
## DOLite:100 DOLite:100              Cancer     668           0.283      0
## DOLite:165 DOLite:165            Embryoma     231           0.293      0
## DOLite:306 DOLite:306            Leukemia     289           0.342      0
## DOLite:322 DOLite:322 Lupus erythematosus     124           0.366      0
## DOLite:337 DOLite:337            Melanoma     136           0.370      0
## DOLite:64   DOLite:64     Atherosclerosis     192           0.308      0
##            p.adjust qvalues
## DOLite:100        0       0
## DOLite:165        0       0
## DOLite:306        0       0
## DOLite:322        0       0
## DOLite:337        0       0
## DOLite:64         0       0
```

The *setType* should be one of ”DO” or ”DOLite and was required for `gseaAnalyzer` to prepare the corresponding gene sets.

```
topID <- res[1, 1]
topID
```

```
## [1] "DOLite:100"
```

```
plot(y, geneSetID = topID)
```

Parameter *geneSetID* can be numeric, the following command will generate the same figure as illustrated above.

```
plot(y, geneSetID = 1)
```

### 5.2.1 enrichMap

Enrichment Map can be visualized by `enrichMap` function. It supports both enrichment result and GSEA result.

# 6 Session Information

The version number of R and packages loaded for generating the vignette were:

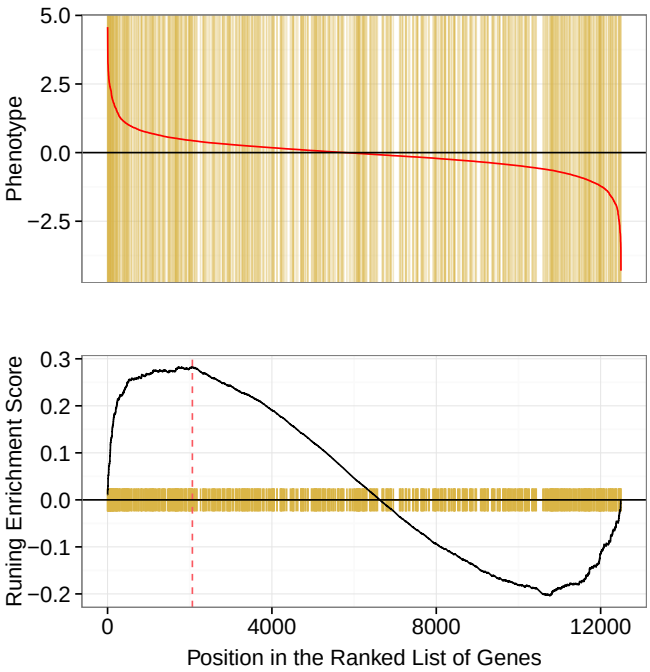


Figure 5: gseaplot example.

- R version 3.1.1 Patched (2014-09-25 r66681), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: AnnotationDbi 1.28.0, Biobase 2.26.0, BiocGenerics 0.12.0, DBI 0.3.1, DO.db 2.8.0, DOSE 2.4.0, GenomeInfoDb 1.2.0, IRanges 2.0.0, RSQLite 0.11.4, S4Vectors 0.4.0, clusterProfiler 2.0.0, knitr 1.7, org.Hs.eg.db 3.0.0
- Loaded via a namespace (and not attached): GO.db 3.0.0, GOSemSim 1.24.0, KEGG.db 3.0.0, MASS 7.3-35, Rcpp 0.11.3, codetools 0.2-9, colorspace 1.2-4, digest 0.6.4, evaluate 0.5.5, formatR 1.0, ggplot2 1.0.0, grid 3.1.1, gtable 0.1.2, highr 0.3, igraph 0.7.1, labeling 0.3, munsell 0.4.2, plyr 1.8.1, proto 0.3-10, qvalue 1.40.0, reshape2 1.4, scales 0.2.4, stringr 0.6.2, tools 3.1.1

# References

[1] L. M. Schriml, C. Arze, S. Nadendla, Y.-W. W. Chang, M. Mazaitis, V. Felix, G. Feng, and W. A. Kibbe. Disease ontology: a backbone for disease se-

mantic integration. *Nucleic Acids Research*, 40(D1):D940–D946, November 2011.

[2] Resnik Philip. Semantic similarity in a taxonomy: An Information-Based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

[3] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *Proceedings of 10th International Conference on Research In Computational Linguistics*, 1997.

[4] Dekang Lin. An Information-Theoretic definition of similarity. *In Proceedings of the 15th International Conference on Machine Learning*, pages 296—304, 1998.

[5] Andreas Schlicker, Francisco S Domingues, Jrg Rahnenfhrer, and Thomas Lengauer. A new measure for functional similarity of gene products based on gene ontology. *BMC Bioinformatics*, 7:302, 2006. PMID: 16776819.

[6] James Z Wang, Zhidian Du, Rapeeporn Payattakool, Philip S Yu, and Chin-Fu Chen. A new method to measure the semantic similarity of go terms. *Bioinformatics (Oxford, England)*, 23:1274–81, May 2007. PMID: 17344234.

[7] 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*, 26:976–978, 2010. PMID: 20179076.

[8] Elizabeth I Boyle, Shuai Weng, Jeremy Gollub, Heng Jin, David Botstein, J Michael Cherry, and Gavin Sherlock. GO::TermFinder–open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics (Oxford, England)*, 20(18):3710–3715, December 2004. PMID: 15297299.

[9] Pan Du, Gang Feng, Jared Flatow, Jie Song, Michelle Holko, Warren A. Kibbe, and Simon M. Lin. From disease ontology to disease-ontology lite: statistical methods to adapt a general-purpose ontology for the test of gene-ontology associations. *Bioinformatics*, 25(12):i63–i68, 2009.

[10] Guangchuang Yu, Li-Gen Wang, Yanyan Han, and Qing-Yu He. clusterProfiler: an r package for comparing biological themes among gene clusters. *OMICS: A Journal of Integrative Biology*, 16(5):284–287, May 2012.

[11] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550, October 2005.