GO-terms Semantic Similarity Measures

Guangchuang Yu

School of Public Health
The University of Hong Kong
guangchuangyu@gmail.com

April 16, 2015

GOSemSim version: 1.26.0

If you use *GOSemSim* in published research, please cite:

G Yu, F Li, Y Qin, X Bo, Y Wu, S Wang. **GOSemSim: an R package for measuring semantic similarity among GO terms and gene products**. *Bioinformatics* 2010, 26(7):976-978.
http://dx.doi.org/10.1093/bioinformatics/btq064

Contents

1 Introduction

Functional similarity of gene products can be estimated by controlled biological vocabularies, such as Gene Ontology (GO). GO comprises of three orthogonal ontologies, i.e. molecular function (MF), biological process (BP), and cellular component (CC).

Four methods have been presented to determine the semantic similarity of two GO terms based on the annotation statistics of their common ancestor terms (Resnik [1], Jiang [2], Lin [3] and Schlicker [4]). Wang [5] proposed a new method to measure the similarity based on the graph structure of GO. Each of these methods has its own advantages and weaknesses. *GOSemSim* package [6] is developed to compute semantic similarity among GO terms, sets of GO terms, gene products, and gene clusters, providing both five methods mentioned above. I have developed another package, *DOSE* [7], for measuring semantic similarity among DO terms and gene products at disease perspective.

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

```
library(GOSemSim)
help(GOSemSim)
```

2 Semantic Similarity Measurement Based on GO

2.1 Information content-based methods

Four methods proposed by Resnik [1], Jiang [2], Lin [3] and Schlicker [4] are information content (IC) based, which depend on the frequencies of two GO terms involved and that of their closest common ancestor term in a specific corpus of GO annotations. The information content of a GO term is computed by the negative log probability of the term occurring in GO corpus. A rarely used term contains a greater amount of information.

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 GO corpus.

Thus the information content is defined as:

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

As GO allow multiple parents for each concept, two terms can share parents by multiple paths. IC-based methods calculate similarity of two GO 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:

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

At present, *GOSemSim* supports analysis on many species. We used the following Bioconductor packages to calculate the information content.

- *org.At.tair.db* for *Arabidopsis*
- *org.Ag.eg.db* for *Anopheles*
- *org.Bt.eg.db* for *Bovine*
- *org.Cf.eg.db* for *Canine*
- *org.Gg.eg.db* for *Chicken*
- *org.Pt.eg.db* for *Chimp*
- *org.Sco.eg.db* for *Coelicolor*
- *org.EcK12.eg.db* for *E coli strain K12*
- *org.EcSakai.eg.db* for *E coli strain Sakai*
- *org.Dm.eg.db* for *Fly*
- *org.Hs.eg.db* for *Human*
- *org.Pf.plasmo.db* for *Malaria*
- *org.Mm.eg.db* for *Mouse*
- *org.Ss.eg.db* for *Pig*
- *org.Rn.eg.db* for *Rat*

- *org.Mmu.eg.db* for *Rhesus*
- *org.Ce.eg.db* for *Worm*
- *org.Xl.eg.db* for *Xenopus*
- *org.Sc.sgd.db* for *Yeast*
- *org.Dr.eg.db* for *Zebrafish*

The information content will update regularly.

2.2 Graph-based method

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

2.2.1 Wang method

To encode the semantic of a GO 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 GO term *t* to the semantic of GO term A as the S-value of GO 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 GO 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 GO term *t* related to term A and $S_B(t)$ is the S-value of GO term *t* related to term B.

This method proposed by Wang [5] determines the semantic similarity of two GO terms based on both the locations of these terms in the GO graph and their relations with their ancestor terms.

2.3 goSim and mgoSim function

In *GOSemSim*, we implemented all these IC-based and graph-based methods. `goSim` calculate semantic similarity between two GO terms, while `mgoSim` calculate semantic similarity between two sets of GO terms.

```
goSim("GO:0004022", "GO:0005515", ont="MF", measure="Wang")
```

```
## [1] 0.158
```

```
go1 = c("GO:0004022","GO:0004024","GO:0004174")
go2 = c("GO:0009055","GO:0005515")
mgoSim(go1, go2, ont="MF", measure="Wang", combine=NULL)
```

```
##            GO:0009055 GO:0005515
## GO:0004022      0.205      0.158
## GO:0004024      0.185      0.141
## GO:0004174      0.205      0.158
```

```
mgoSim(go1, go2, ont="MF", measure="Wang", combine="BMA")
```

```
## [1] 0.192
```

3 Gene Semantic Similarity Measurement

On the basis of semantic similarity between GO terms, *GOSemSim* can also compute semantic similarity among sets of GO terms, gene products, and gene clusters.

We implemented four methods which called *max*, *avg*, *rcmax*, and *BMA* to combine semantic similarity scores of multiple GO terms. The similarities among gene products and gene clusters which annotated by multiple GO terms were also calculated by the same combine methods mentioned above.

Suppose we have gene g_1 annotated by GO terms sets $GO_1 = \{go_{11}, go_{12} \cdots go_{1m}\}$ and g_2 annotated by $GO_2 = \{go_{21}, go_{22} \cdots go_{2n}\}$, *GOSemSim* implemented four methods which called `max`, `avg`, `rcmax` and `BMA` to combine semantic similairty scores of multiple GO terms.

3.1 Combine methods

3.1.1 max

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

$$sim_{max}(g\_1, g\_2) = \max_{1 \le i \le m, 1 \le j \le n} sim(go_{1i}, go_{2j})$$

3.1.2 avg

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

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

3.1.3 rcmax

Similarities among two sets of GO 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).

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

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:

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

3.2 geneSim and mgeneSim

In *GOSemSim*, we implemented `geneSim` to calculate semantic similarity between two gene products, and `mgeneSim` to calculate semantic similarity among multiple gene products.

```
geneSim("241", "251", ont="MF", organism="human", measure="Wang", combine="BMA")
```

```
## $geneSim
## [1] 0.141
##
## $GO1
## [1] "GO:0050544" "GO:0005515" "GO:0047485"
##
## $GO2
## [1] "GO:0004035"
```

```
mgeneSim(genes=c("835", "5261","241", "994"),
                        ont="MF", organism="human", measure="Wang",
                        verbose=FALSE)
```

```
##         835  5261   241   994
## 835   1.000 0.116 0.612 0.615
## 5261  0.116 1.000 0.108 0.115
## 241   0.612 0.108 1.000 0.626
## 994   0.615 0.115 0.626 1.000
```

3.3 clusterSim and mclusterSim

We also implemented `clusterSim` for calculating semantic similarity between two gene clusters, and `mclusterSim` for calculating semantic similarities among multiple gene clusters.

```
gs1 <- c("835", "5261","241", "994", "514", "533")
gs2 <- c("578","582", "400", "409", "411")
clusterSim(gs1, gs2, ont="MF", organism="human", measure="Wang", combine="BMA")

x <- org.Hs.egGO
hsEG <- mappedkeys(x)
set.seed <- 123
clusters <- list(a=sample(hsEG, 20), b=sample(hsEG, 20), c=sample(hsEG, 20))
mclusterSim(clusters, ont="MF", organism="human", measure="Wang", combine="BMA")
```

4 Case Study

We proposed a method for measuring functional similarity of microRNAs [8]. This method was based on semantic similarity of microRNAs' target genes, and was calculated by *GOSemSim*. We further analyzed viral microRNAs using this method and compared significant KEGG pathways regulated by different viruses' microRNAs [9] using *clusterProfiler* [10].

5 GO enrichment analysis

GO enrichment analysis can be supported by our package *clusterProfiler* [10], which supports hypergeometric test and Gene Set Enrichment Analysis (GSEA). Enrichment results across different gene clusters can be compared using `compareCluster` function.

6 Disease Ontology Semantic and Enrichment analysis

Disease Ontology (DO) annotates human genes in the context of disease. DO is important annotation in translating molecular findings from high-throughput data to clinical relevance. *DOSE* [7] supports semantic similarity computation among DO terms and genes. Enrichment analysis including hypergeometric model and GSEA are also implemented to support discovering disease associations of high-throughput biological data.

7 External documents

- proper use of GOSemSim

8 Session Information

9 Session Information

Here is the output of `sessionInfo()` on the system on which this document was compiled:

- R version 3.2.0 (2015-04-16), `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 1.30.0, Biobase 2.28.0, BiocGenerics 0.14.0, DBI 0.3.1, GO.db 3.1.2, GOSemSim 1.26.0, GenomeInfoDb 1.4.0, IRanges 2.2.0, RSQLite 1.0.0, S4Vectors 0.6.0, knitr 1.9, org.Hs.eg.db 3.1.2
- Loaded via a namespace (and not attached): BiocStyle 1.6.0, Rcpp 0.11.5, evaluate 0.6, formatR 1.1, highr 0.4.1, stringr 0.6.2, tools 3.2.0

References

[1] 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.

[2] 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. URL: http://www.citebase.org/abstract?id=oai:arXiv.org:cmp-lg/9709008.

[3] Dekang Lin. An Information-Theoretic definition of similarity. *In Proceedings of the 15th International Conference on Machine Learning*, pages 296—304, 1998. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.1832, doi:10.1.1.55.1832.

[4] 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. doi:1471-2105-7-302.

[5] 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. URL: http://www.ncbi.nlm.nih.gov/pubmed/17344234, doi:btm087.

[6] 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. URL: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/26/7/976, doi:10.1093/bioinformatics/btq064.

[7] Guangchuang Yu, Li-Gen Wang, Guang-Rong Yan, and Qing-Yu He. DOSE: an r/bioconductor package for disease ontology semantic and enrichment analysis. 31(4):608–609. URL: http://bioinformatics.oxfordjournals.org.eproxy2.lib.hku.hk/content/31/4/608, doi:10.1093/bioinformatics/btu684.

[8] Guangchuang Yu, Chuan-Le Xiao, Xiaochen Bo, Chun-Hua Lu, Yide Qin, Sheng Zhan, and Qing-Yu He. A new method for measuring functional similarity of micrornas. *Journal of Integrated OMICS*, 1(1):49–54, February 2011. URL: http://www.jiomics.com/index.php/jio/article/view/21, doi:10.5584/jiomics.v1i1.21.

[9] Guangchuang Yu and Qing-Yu He. Functional similarity analysis of human virus-encoded miRNAs. *Journal of Clinical Bioinformatics*, 1(1):15, May 2011. URL: http://www.jclinbioinformatics.com/content/1/1/15, doi:10.1186/2043-9113-1-15.

[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. URL: http://online.liebertpub.com/doi/abs/10.1089/omi.2011.0118, doi:10.1089/omi.2011.0118.