

GO-terms Semantic Similarity Measures

Guangchuang Yu
College of Life Science and Technology
Jinan University, Guangzhou, China
email: guangchuangyu@gmail.com

February 28, 2012

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 (?), Jiang (?), Lin (?) and Schlicker (?)). Wang (?) 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 (?) is developed to compute semantic similarity among GO terms, sets of GO terms, gene products, and gene clusters, providing both five methods mentioned above. To start with *GOSemSim* package, type following code below:

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

2 Citation

Please cite the following articles when using *GOSemSim*.

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.

3 Semantic Similarity Measurement Based on GO

Four methods proposed by Resnik (?), Jiang (?), Lin (?) and Schlicker (?) are information content 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. Information content is defined as frequency of each term occurs in the corpus. 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.

As GO allow multiple parents for each concept, two terms can share parents by multiple paths. We take the minimum $p(t)$, where there is more than one shared parents. The p_{ms} is defined as:

$$p_{ms}(t1, t2) = \min_{t \in S(t1, t2)} \{p(t)\}$$

Where $S(t1, t2)$ is the set of parent terms shared by $t1$ and $t2$.

- Resnik's method is defined as:

$$sim(t1, t2) = -\ln p_{ms}(t1, t2)$$

- Lin's method is defined as:

$$sim(t1, t2) = \frac{2 \times \ln(p_{ms}(t1, t2))}{\ln p(t1) + \ln p(t2)}$$

- Schlicker's method, which combine Resnik's and Lin's method, is defined as:

$$sim(t1, t2) = \frac{2 \times \ln p_{ms}(t1, t2)}{\ln p(t1) + \ln p(t2)} \times (1 - p_{ms}(t1, t2))$$

- Jiang and Conrath's method is defined as:

$$sim(t1, t2) = 1 - \min(1, d(t1, t2))$$

where

$$d(t1, t2) = \ln p(t1) + \ln p(t2) - 2 \times \ln p_{ms}(t1, t2)$$

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 .

- Wang's method

To encode the semantics of a GO term in a measurable format to enable a quantitative comparison between two term's semantics, Wang firstly define 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, define the contribution of a GO term t to the semantics of GO term A as the S-value of GO term t related to term A. For any of term t in $DAG_A = (A, T_A, E_A)$, its S-value related to term A. $S_A(t)$ is defined as:

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

where w_e is the semantic contribution factor for edge $e \in E_A$ linking term t with its child term t' . Wang defined term A contributes to its own as one. After obtaining the S-values for all terms in DAG_A , the semantic value of GO 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, $GO_{A,B}$, is defined as:

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

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 (?) 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.

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*, *average*, *rcmax*, and *rcmax.avg* 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.

Given two GO terms sets $GO_1 = \{go_{11}, go_{12} \cdots go_{1m}\}$ and $GO_2 = \{go_{21}, go_{22} \cdots go_{2n}\}$, method *max* calculate the maximum semantic similarity score over all pairs of GO terms between these two sets, method *average* calculate the average semantic similarity score over all pairs of GO terms.

Similarities between GO terms form a matrix, and method *rcmax* use the maximum of RowScore and ColumnScore as the similarity, where RowScore (or ColumnScore) is the average of maximum similarities on each row (or column).

And method *rcmax.avg* calculate the average of all maximum similarities on each row and column, and defined as:

$$Sim(GO1, GO2) = \frac{\sum_{1 \leq i \leq m} \max(Sim((go_{1i}), (GO_2))) + \sum_{1 \leq j \leq n} \max(Sim((go_{2j}), (GO_1)))}{m+n}$$

4 Examples

GOSemSim implemented multiple functions for calculate semantic similarities:

- *goSim* for calculate semantic similarity between two GO terms.
- *mgoSim* for calculate semantic similarity among multiple GO terms.
- *geneSim* for calculate semantic similarity between two gene products.
- *mgeneSim* for calculate semantic similarity among multiple gene products.
- *clusterSim* for calculate semantic similarity between two gene clusters.
- *mclusterSim* for calculate semantic similarity among multiple gene clusters.

The following example demonstrated the function calls of these function, details about the arguments can refer to the manuals (eg *?geneSim*).

```
> goSim("GO:0004022", "GO:0005515", ont="MF", measure="Wang")
[1] 0.252

> go1 = c("GO:0004022", "GO:0004024", "GO:0004174")
> go2 = c("GO:0009055", "GO:0005515")
> mgoSim(go1, go2, ont="MF", measure="Wang", combine="rcmax.avg")
[1] 0.299

> geneSim("241", "251", ont="MF", organism="human", measure="Wang", combine="rcmax.avg")
[1] "loading GOMap..."
[1] "Done..."
$geneSim
[1] 0.228

$GO1
[1] "GO:0005515" "GO:0047485" "GO:0050544"

$GO2
[1] "GO:0004035"

> mgeneSim(genes=c("835", "5261", "241", "994"), ont="MF", organism="human", measure="Wang", combine="rcmax.avg")
      835  5261   241   994
835  1.000 0.246 0.726 0.525
5261 0.246 1.000 0.242 0.168
241   0.726 0.242 1.000 0.447
994   0.525 0.168 0.447 1.000

> 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="rcmax.avg")
```

```
[1] 0.641
```

```
> 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="rcmax.avg")

      a      b      c
a 1.000 0.578 0.652
b 0.578 1.000 0.660
c 0.652 0.660 1.000
```

GOSemSim was internally designed using S4 Object Oriented paradigm. The functions above are wrapper functions of the S4 method *sim*. For advance users, we recommend using *sim* method directly for calculate semantic similarities.

Firstly a *Params* class was defined to store a set of parameters for measuring semantic similarity. *Params* containing parameters are ontology, organism, method, combine, and dropCodes. Parameter ontology specify which ontology were used in measurement, organism specify which GO Map were loaded for mapping Gene IDs to GO terms, dropCodes restrict evident codes when mapping Gene IDs to GO Terms, method specify which method to be used to measure the similarity and combine specify which combine method was used to combining semantic similarity scores.

```
> params <- new("Params", ontology="MF", organism="human", method="Wang")
```

A *GOSet* class stores two set of GO IDs.

```
> go1 <- c("GO:0004022", "GO:0004024", "GO:0004023")
> go2 <- c("GO:0009055", "GO:0020037")
> gos <- new("GOSet", GOSet1=go1, GOSet2=go2)
```

A *GeneSet* class containing two set of Gene IDs.

```
> gs <- new("GeneSet", GeneSet1=gs1, GeneSet2=gs2)
```

A *GeneClusterSet* class containing a list of gene clusters.

```
> geneClusters <- new("GeneClusterSet", GeneClusters=clusters)
```

S4 method *sim* was designed to measuring semantic similarity for *GOSet*, *GeneSet* and *GeneClusterSet*.

```
> sim(gos, params)

      GO:0009055 GO:0020037
GO:0004022      0.318      0.151
GO:0004024      0.290      0.136
GO:0004023      0.290      0.136

> setCombineMethod(params) <- "rcmax.avg"
> sim(gos, params)

[1] 0.273

> sim(gs, params)
```

```
[1] "loading GOMap..."
[1] "Done..."
      578    582    400    409    411
835  0.736 1.000 0.471 0.726 0.576
5261 0.591 0.842 0.527 0.606 0.376
241  0.744 1.000 0.533 0.700 0.382
994  0.467 0.647 0.377 0.555 0.665
514  0.292 0.390 0.421 0.277 0.603
533  0.382 0.481 0.354 0.359 0.406
```

```
> sim(geneClusters, params)
```

```
      a      b      c
a 1.000 0.763 0.730
b 0.763 1.000 0.804
c 0.730 0.804 1.000
```

5 Case Study

In ?, we proposed a method for measuring functional similarity of microRNAs. 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 using *clusterProfiler* (?).

The semantic similarities of human viral microRNAs which was calculated by *GOSemSim* was illustrated in Figure 1.

6 Session Information

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

```
R Under development (unstable) (2012-02-25 r58492)
Platform: i686-pc-linux-gnu (32-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8      LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=C               LC_NAME=C
 [9] LC_ADDRESS=C            LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats      graphics  grDevices  utils      datasets
[6] methods    base

other attached packages:
[1] GO.db_2.6.1          org.Hs.eg.db_2.6.4
[3] GOSemSim_1.13.4      AnnotationDbi_1.17.22
[5] Biobase_2.15.3       BiocGenerics_0.1.6
[7] RSQLite_0.11.1       DBI_0.2-5
```

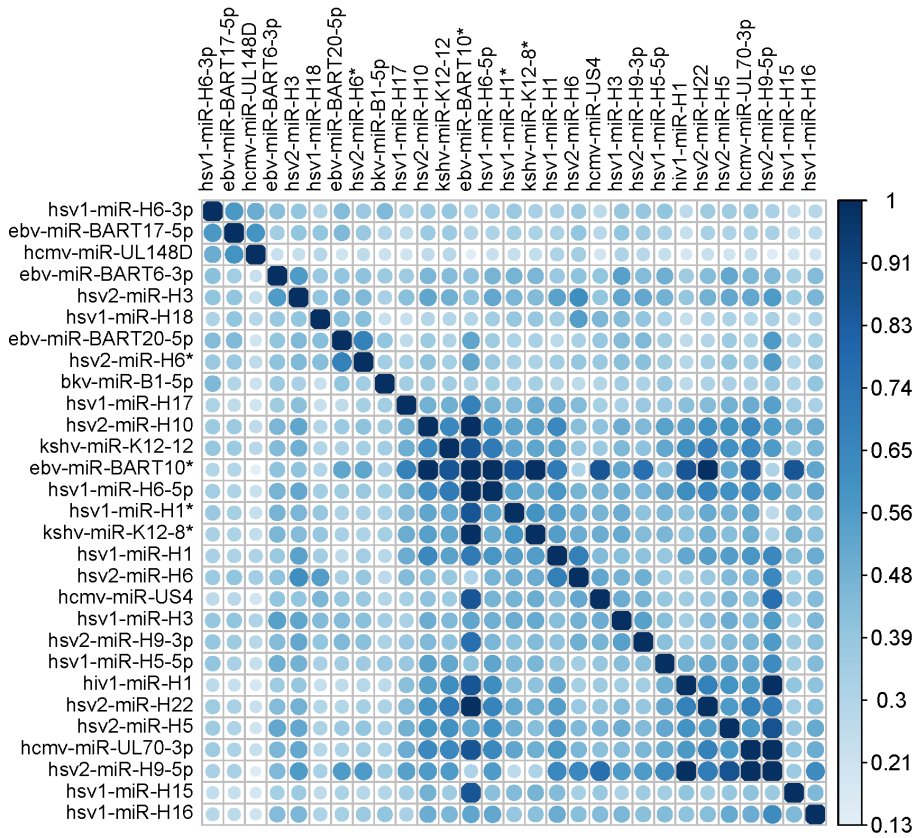


Figure 1: Semantic Similarities among Viral microRNAs

loaded via a namespace (and not attached):

```
[1] IRanges_1.13.25      org.Ag.eg.db_2.6.4
[3] org.At.tair.db_2.6.4  org.Bt.eg.db_2.6.4
[5] org.Ce.eg.db_2.6.4    org.Cf.eg.db_2.6.4
[7] org.Dm.eg.db_2.6.4    org.Dr.eg.db_2.6.4
[9] org.EcK12.eg.db_2.6.4 org.EcSakai.eg.db_2.6.4
[11] org.Gg.eg.db_2.6.4    org.Mm.eg.db_2.6.4
[13] org.Mmu.eg.db_2.6.4   org.Pf.plasmo.db_2.6.4
[15] org.Pt.eg.db_2.6.4    org.Rn.eg.db_2.6.4
[17] org.Sc.sgd.db_2.6.4   org.Ss.eg.db_2.6.4
[19] org.Xl.eg.db_2.6.4    tools_2.15.0
```