How to compute a semantic similarity threshold

Charles Bettembourg, Christian Diot, Olivier Dameron
Abstract

The analysis of gene annotations related to Gene Ontology plays an important role in the interpretation of high-throughput experiments results. This analysis typically involves semantic similarity and particularity measures that quantify the importance of the Gene Ontology annotations. In our paper entitled “Interpretation of semantic similarity and particularity for functional analysis of gene set”, we propose a new method to compute thresholds supporting the interpretation of the results of a semantic comparison. Indeed, there was no sound method supporting the interpretation of the similarity and particularity values in order to determine whether two genes are similar or whether one gene has some significant particular function. Interpretation is frequently based either on an implicit threshold, or an arbitrary one (typically 0.5). Our method determines the optimal similarity threshold by minimizing the proportions of false-positive and false-negative similarity matches using distributions of the similarity values of pairs of similar genes and pairs of non-similar genes. In this “How to”, we explain the process step by step for determining one of the thresholds computed in our paper: the Biological Process threshold using Wang’s similarity measure.
Similarity threshold determination process

This process is composed of three steps:

1. Define at least two different groups of similar genes for species of interest.

2. (a) In each group, compute the similarities between each pair of genes (i.e. the intra-group similarities). Gather all the similarity results to obtain an S distribution of similar genes.

   (b) Compute the similarities between each combination of a gene from the first group and a gene from a second group (i.e. the inter-group similarities). Gather all the similarity results to obtain an N distribution of non-similar genes.

3. If the S and N distributions have no overlap between the data ranges (min, max), define the threshold $\tau_{\text{sim}}$ using any value between $\tau_S$ (the lowest value of S) and $\tau_N$ (the highest value of N). Else, there are some false negatives (FN) and some false positives (FP):

   (a) Compute the FN proportion of FN in the S distribution for all samples of the similarity threshold between $\tau_N$ to $\tau_S$. In this step, consider every value under the similarity threshold as a FN.

   (b) Compute the FP proportion of FP in the N distribution for all samples of the similarity threshold between $\tau_N$ to $\tau_S$. In this step, consider every value above the similarity threshold as a FP.

   (c) For each possible threshold value, sum the FN and FP proportions obtained in steps 3a and 3b. The similarity threshold $\tau_{\text{sim}}$ is the threshold that minimizes this sum.
Step 1: Define at least two different groups of similar genes for species of interest

Here we need biological knowledge to compose groups of genes that we know similar. As we focus on biological processes, we propose to use PANTHER families. The PANTHER families are composed of genes sharing an evolutionary history, molecular functions and biological processes annotations, and an involvement in the same biological pathways. Consequently, we assumed that genes belonging to a same PANTHER family share enough features to be considered as being involved in similar biological processes. Inversely, we assumed that two genes belonging to two different PANTHER families should not be considered as being involved in similar biological processes.

We need a minimum of two gene groups to compute a threshold but considering more groups increases threshold robustness. In this document, we will use six PANTHER families as groups of similar genes. We select families involved in different biological processes and having at least 10 genes, with no gene common between the selected families. These families are:

- “histone h1/h5 (pthr11467)”
- “g-protein coupled receptor” (pthr12011)
- “neurotransmitter gated ion channel” (pthr18945)
- “tyrosine-protein kinase receptor” (pthr24416)
- “phosphatidylinositol kinase” (pthr10048)
- “sulfate transporter” (pthr11814)

Table 1 gives the group of genes for each family.
Table 1: **Genes of six PANTHER families**

<table>
<thead>
<tr>
<th>PANTHER family</th>
<th>Genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>histone h1/h5 (pthr11467)</td>
<td>H1FOO HIST1H1A HIST1H1B HIST1H1C HIST1H1D</td>
</tr>
<tr>
<td></td>
<td>HIST1H1E HIST1H1T H1FX H1F0 H1FNT</td>
</tr>
<tr>
<td>g-protein coupled receptor (pthr12011)</td>
<td>GLP1R CALCRL GPR64 GPR111 GPR112 GPR113</td>
</tr>
<tr>
<td></td>
<td>GPR114 GPR115 GPR116 GPR125 GPR56 CRHR1</td>
</tr>
<tr>
<td></td>
<td>EVA1C ELTD1 EMR2 CELSR3 BA3 BA2 PTH2R</td>
</tr>
<tr>
<td></td>
<td>CD97 GIPR PTH1R GPR110 LPHN2 GPR128 EMR1</td>
</tr>
<tr>
<td></td>
<td>GPR133 GHRHR SCTR GCCR GPR97 LPHN1 VIPIR1</td>
</tr>
<tr>
<td></td>
<td>GPR144 GPR124 GLP2R VIPIR2 ADCYAP1R1</td>
</tr>
<tr>
<td></td>
<td>LPHN3 EMR3 CRHR2 GPR123 GPR126</td>
</tr>
<tr>
<td>neurotransmitter gated ion channel</td>
<td>CHRNE HTR3A CHRND CHRN4 GABRG1 GLRA4</td>
</tr>
<tr>
<td>(pthr18945)</td>
<td>GABRA6 GABRA3 GABRB2 GABRB3 GABRQ</td>
</tr>
<tr>
<td></td>
<td>GABRD HTR3D GABRA5 GABRA6 GABRA3 GABRB4</td>
</tr>
<tr>
<td></td>
<td>GABRA2 GABRE GLRA3 CHRNA7 GABRA4 GLRB</td>
</tr>
<tr>
<td></td>
<td>CHRNA10 CHRNG GABRG2 GABRB1 HTR3B</td>
</tr>
<tr>
<td></td>
<td>CHRN41 CHRFAM7A CHRN22 GABRB2 GABRA2</td>
</tr>
<tr>
<td></td>
<td>HTR3C GABRG3 CHRNA4 GABRA5 CHRN53</td>
</tr>
<tr>
<td></td>
<td>CHRNA3 ZACN GABRP NAVR4 GABRA1 CHRNA1</td>
</tr>
<tr>
<td></td>
<td>HTR3E GLRA2 GLRA1</td>
</tr>
<tr>
<td>tyrosine-protein kinase receptor</td>
<td>MCT1R DDR2 DCD2 KIT FGFR2 FGFR3 EPHA4</td>
</tr>
<tr>
<td>(pthr24416)</td>
<td>EPHA5 EPHA3 ALK FGFR4 IGF1R AXL PDGFR1</td>
</tr>
<tr>
<td></td>
<td>EPHA6 RET EPHA1 CSF1R MERTK TYRO3 FLT3</td>
</tr>
<tr>
<td></td>
<td>EGF EPRA7 ERBB4 ERBB3 FGFR2 KDR FLT4</td>
</tr>
<tr>
<td></td>
<td>PDGFRB FLT1 LTK EPHA2 EPHA3 EPHA2</td>
</tr>
<tr>
<td></td>
<td>EPHA10 INSR TEK TIE1 INSRR EPHB6 DDR1 MUSK</td>
</tr>
<tr>
<td></td>
<td>RYK FGFR1 ROR2 ROR1</td>
</tr>
<tr>
<td>phosphatidylinositol kinase (pthr10048)</td>
<td>PI4KA PIK3CB PIK3CA PIK3CG PIK3C2G PI4KB</td>
</tr>
<tr>
<td></td>
<td>PIK3C2A PIK3CD PIK3C2B PIK3C3</td>
</tr>
<tr>
<td>sulfate transporter (pthr11814)</td>
<td>SLC26A7 SLC26A5 SLC26A4 SLC26A1 SLC26A2</td>
</tr>
<tr>
<td></td>
<td>SLC26A11 SLC26A8 SLC26A9 SLC26A3 SLC26A10</td>
</tr>
<tr>
<td></td>
<td>SLC26A6</td>
</tr>
</tbody>
</table>

Each PANTHER family of this table is one group of similar genes used to compute a BP threshold.
Step 2: Semantic similarity computation

Intra-group similarity

At this step, we compute similarities between each pair of genes within each group. For example, for the group “phosphatidylinositol kinase" (pthr10048), we have to compute all pairwise similarities for the couples of genes given in Table 2. The result of these similarity computation gives a first S distribution.

Table 2: Couple of genes for which we have to compute pairwise similarity for phosphatidylinositol kinase family

<table>
<thead>
<tr>
<th>phosphatidylinositol kinase intra-group couple of genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(PI4KA, PIK3CB) (PI4KA, PIK3CA) (PI4KA, PIK3CG) (PI4KA, PIK3C2G) (PI4KA, PIK3C2B)</td>
</tr>
<tr>
<td>(PI4KB, PI4KA, PIK3C2A) (PI4KA, PIK3CD) (PI4KA, PIK3C2B)</td>
</tr>
<tr>
<td>(PI4KA, PIK3C3) (PIK3CB, PIK3CA) (PIK3CB, PIK3CG) (PIK3CB, PIK3C2G) (PIK3CB, PI4KB) (PIK3CB, PIK3C2A)</td>
</tr>
<tr>
<td>(PI4KB, PIK3CD) (PIK3CB, PIK3C2B) (PIK3CB, PIK3C3) (PIK3CB, PIK3C2G) (PIK3CB, PIK3C2B)</td>
</tr>
<tr>
<td>(PI4KA, PIK3CA, PIK3CB) (PI4KA, PIK3CA, PIK3CA) (PI4KA, PIK3CA, PIK3CG) (PI4KA, PIK3CA, PIK3C2G) (PI4KA, PIK3CA, PIK3C2B)</td>
</tr>
<tr>
<td>(PI4KA, PIK3CA, PIK3C3) (PI4KA, PIK3CA, PIK3C2G) (PI4KA, PIK3CA, PIK3C2B) (PI4KA, PIK3CA, PIK3C3)</td>
</tr>
<tr>
<td>(PI4KA, PIK3CA, PIK3C2G) (PI4KA, PIK3CA, PIK3C2B) (PI4KA, PIK3CA, PIK3C3)</td>
</tr>
<tr>
<td>(PI4KA, PIK3CA, PIK3C2A) (PI4KA, PIK3CA, PIK3C2G) (PI4KA, PIK3CA, PIK3C2B) (PI4KA, PIK3CA, PIK3C3)</td>
</tr>
<tr>
<td>(PI4KA, PIK3CA, PIK3C2A) (PI4KA, PIK3CA, PIK3C2G) (PI4KA, PIK3CA, PIK3C2B) (PI4KA, PIK3CA, PIK3C3)</td>
</tr>
<tr>
<td>(PI4KA, PIK3CA, PIK3C2A) (PI4KA, PIK3CA, PIK3C2G) (PI4KA, PIK3CA, PIK3C2B) (PI4KA, PIK3CA, PIK3C3)</td>
</tr>
</tbody>
</table>

The result of the computation of similarity for these couples provides one of the six S distribution.

Here we compute a BP threshold for Wang’s similarity measure. We use an in house implementation of this measure, available at http://bettembourg.fr/labo/go2sip/ and illustrated by Figure 1

We repeat this process for each family to obtain six S distributions of similarity values.
Figure 1: Intra-group similarity results for phosphatidylinositol kinase (pthr10048).

Inter-group similarity

At this step, we compute the similarities between each combination of a gene from a first group and a gene from a second group. With our six groups, we will obtain 15 inter-groups N distributions as presented by Table 3.

Table 3: 15 inter-groups N distributions

<table>
<thead>
<tr>
<th>Biological Process</th>
<th>PRCA (9606)</th>
<th>PHI3CB (9606)</th>
<th>PHI3CB (9606)</th>
<th>PHI3C (9606)</th>
<th>PHI3C (9606)</th>
<th>PHI3C (9606)</th>
<th>PHI3C (9606)</th>
<th>PHI3C (9606)</th>
</tr>
</thead>
<tbody>
<tr>
<td># Amnilo</td>
<td>8</td>
<td>11</td>
<td>35</td>
<td>38</td>
<td>8</td>
<td>15</td>
<td>22</td>
<td>10</td>
</tr>
<tr>
<td>PRCA (9606)</td>
<td>8</td>
<td>1.0</td>
<td>0.12</td>
<td>0.516</td>
<td>0.455</td>
<td>0.849</td>
<td>0.658</td>
<td>0.57</td>
</tr>
<tr>
<td>PHI3CB (9606)</td>
<td>11</td>
<td>0.912</td>
<td>1.0</td>
<td>0.545</td>
<td>0.475</td>
<td>0.869</td>
<td>0.688</td>
<td>0.602</td>
</tr>
<tr>
<td>PHI3CB (9606)</td>
<td>35</td>
<td>0.516</td>
<td>0.545</td>
<td>1.0</td>
<td>0.685</td>
<td>0.521</td>
<td>0.562</td>
<td>0.552</td>
</tr>
<tr>
<td>PHI3C (9606)</td>
<td>38</td>
<td>0.455</td>
<td>0.475</td>
<td>0.689</td>
<td>1.0</td>
<td>0.442</td>
<td>0.501</td>
<td>0.503</td>
</tr>
<tr>
<td>PHI3C (9606)</td>
<td>8</td>
<td>0.849</td>
<td>0.889</td>
<td>0.521</td>
<td>0.442</td>
<td>1.0</td>
<td>0.745</td>
<td>0.581</td>
</tr>
<tr>
<td>PHI3C (9606)</td>
<td>15</td>
<td>0.653</td>
<td>0.688</td>
<td>0.580</td>
<td>0.501</td>
<td>0.749</td>
<td>1.0</td>
<td>0.633</td>
</tr>
<tr>
<td>PHI3C (9606)</td>
<td>22</td>
<td>0.57</td>
<td>0.662</td>
<td>0.552</td>
<td>0.503</td>
<td>0.581</td>
<td>0.633</td>
<td>1.0</td>
</tr>
<tr>
<td>PHI3C (9606)</td>
<td>10</td>
<td>0.845</td>
<td>0.895</td>
<td>0.566</td>
<td>0.5</td>
<td>0.818</td>
<td>0.636</td>
<td>0.576</td>
</tr>
<tr>
<td>PHI3C (9606)</td>
<td>35</td>
<td>0.424</td>
<td>0.444</td>
<td>0.658</td>
<td>0.703</td>
<td>0.456</td>
<td>0.493</td>
<td>0.437</td>
</tr>
<tr>
<td>PHI3C (9606)</td>
<td>34</td>
<td>0.498</td>
<td>0.534</td>
<td>0.736</td>
<td>0.676</td>
<td>0.485</td>
<td>0.36</td>
<td>0.521</td>
</tr>
</tbody>
</table>

We have to compute the similarity value of each gene of the first group vs each gene of the second group.

We need to compute the similarity value of each gene of the first group vs each gene of the second group.
For example, to obtain the N distribution comparing sulfate transporter vs phosphatidylinositol kinase, we have to compute all the pairwise similarities given in Table 4.

<table>
<thead>
<tr>
<th>sulfate transporter vs phosphatidylinositol kinase inter-group couple of genes</th>
</tr>
</thead>
</table>

The result of the computation of similarity for these couples provides one of the 15 N distributions.
We use our in-house implementation of Wang’s similarity measure to compute all 15 N distributions.

**S and N distribution study**

We can now compile all the S and N distribution in a file structured like shown in Figure 2 in order to display the distributions as boxplots and to compute Welch’s t-tests to ensure the results are significant.

```
1 "gena1" "gena2" "comparison" "value" "intraInter"
2 "P14KA" "P1K3C2A" "a[1] phosphatidylinositol kinase" 0.912 "a_Intra-group"
3 "P14KA" "P1K3C2B" "a[1] phosphatidylinositol kinase" 0.516 "a_Intra-group"
4 ...
5 "SLC26A1" "SLC26A1" "a[2] sulfate transporter" 0.542 "a_Intra-group"
6 ...
7 "P14KA" "SLC26A1" "b[1] vs [2]" 0.246 "b_Inter-group"
8 ...
```

Figure 2: S and N semantic similarity measure results compilation.

The R script `boxplot.R` displayed next page draws the boxplots for a visual analysis of the distributions and computes the Welch’s t-tests. If Welch’s t-test cannot highlight a significant difference between S and N distributions, we have to choose other groups of genes and restart all the process. With the data used, the boxplot obtained is displayed in the part A of the Figure 8 of our paper “Interpretation of semantic similarity and particularity for functional analysis of gene set” and the result of the Welch’s t-tests are in Supporting Information file S2. These Welch’s t-test results allow to consider the S and N distribution significantly different.
bxthreshold <- read.table("/data/similarity_BP.csv", sep="\t", header=TRUE)
comparison <- bxthreshold$comparison
similarity <- bxthreshold$value
intraInter <- bxthreshold$intraInter
mat=matrix(c(1,1,1,2,2,2), nrow=1)
layout(mat)
boxplot(similarity ~ comparison, data = bxthreshold, 
  subset = intraInter == "a_Intra-group",
  col = "lightblue",
  xlab = "groups of genes",
  ylab = "Distribution of the semantic similarity measured",
  boxwex = 0.75,
  ylim = c(0, 1.1),
  las = 2)
boxplot(similarity ~ comparison, data = bxthreshold, 
  add = TRUE,
  subset = intraInter == "b_Inter-group",
  xaxt = "n",
  yaxt = "n",
  col = "yellow",
  boxwex = 0.75,
  ylim = c(0, 1.1),
  las = 2)
boxplot(similarity ~ intraInter, 
  data = bxthreshold,
  subset = intraInter == "a_Intra-group",
  col = "lightblue",
  ylab = "Distribution of the semantic similarity measured",
  ylim = c(0, 1.1))
boxplot(similarity ~ intraInter, 
  data = bxthreshold,
  add = TRUE,
  subset = intraInter == "b_Inter-group",
  xaxt = "n",
  yaxt = "n",
  col = "yellow",
  ylim = c(0, 1.1))
sink("/tests/welch_BP.txt")
cat("Comparison of the boxes Intra-groups and Inter-groups")
tst <- t.test(similarity ~ intraInter, data = bxthreshold)
print(tst)
for (i in unique(comparison)){
  for (j in unique(comparison)){
    if (i != j){
      t1 <- subset(similarity, (comparison == i))
      t2 <- subset(similarity, (comparison == j))
      cat("Comparison of the boxes [", i, "] and [", j, "]")
      tst <- t.test(t1, t2)
      print(tst)
    }
  }
}
Step 3: Overlap study

If the S and N distributions have no overlap between the ranges (min, max), we define the threshold $\tau_{\text{sim}}$ using any value between $\tau_S$ (the lowest value of S) and $\tau_N$ (the highest value of N). Else, like with the data we used here, there are some false negatives (FN) and some false positives (FP).

The lowest value of all S distributions is $\tau_N$, the threshold value under which the two compared genes are non-similar. There cannot be any FN below $\tau_N$, but there will be some FPs above $\tau_N$ as S and N distributions overlap. The highest value of all N distributions is $\tau_S$, the threshold value above which the two compared genes are similar. There cannot be any FP above $\tau_S$, but there will be some FNs below $\tau_S$ as S and N distributions overlap.

To find the best threshold of similarity $\tau_{\text{sim}}$, we have to study the variation of the proportions of FPs and FNs when varying $\tau_{\text{sim}}$ between $\tau_N$ and $\tau_S$. The Python script thresholdVariation.py computes these FP and FN proportions and allows to find the minimum the sum of these FP and FN proportions as shown in Figure 3.

![Figure 3: Sum of FP and FN proportions when varying $\tau_{\text{sim}}$ between $\tau_N$ and $\tau_S$.](image_url)
#! /usr/bin/env python
#
# coding: utf-8
#

gene2group is a dictionary with genes as keys and a group name as value. Example:
gene2group = 
    {"PI4KA": "phosphatidylinositol kinase",
     "PIK3CB": "phosphatidylinositol kinase",
     "PIK3CB": "phosphatidylinositol kinase",
     "SLC26A8": "sulfate transporter",
     "SLC26A1": "sulfate transporter",
     "SLC26A4": "sulfate transporter"}

similarity is a dictionary with genes as first and second level keys and a similarity value as keys. Example:
similarity = 
    {"PI4KA": {"PIK3CB": 0.912,
                "PIK3CB": 0.516,
                "SLC26A8": 0.246,
                "SLC26A1": 0.35,
                "SLC26A4": 0.244},
     "PIK3CB": {"PIK3CB": 0.545,
                 "SLC26A8": 0.237,
                 "SLC26A1": 0.352,
                 "SLC26A4": 0.228},
     "SLC26A8": {"SLC26A1": 0.255,
                  "SLC26A1": 0.316,
                  "SLC26A4": 0.257},
     "SLC26A1": {"SLC26A4": 0.543}}

intra = []inter = []
tauN = 164 # (tau N ∗ 1000)
tauS = 618 # (tau S ∗ 1000)

for gene1 in similarity.keys():
    for gene2 in similarity[gene1]:
        if gene2group[gene1] == gene2group[gene2]:
            intra.append(similarity[gene1][gene2])
        else:
            inter.append(similarity[gene1][gene2])

for thr in range(tauN, tauS, 5):
    fn = 0
    fp = 0

    for value in intra:
        if value <= thr/1000.0:
            fn+=1

    for value in inter:
        if value >= thr/1000.0:
            fp += 1

    fnProportion = round(fn/float(len(intra))*100, 3)
    fpProportion = round(fp/float(len(inter))*100, 3)

    print thr/1000.0, fnProportion + fpProportion