The *GOSim* package

Holger Fröhlich

December 11, 2006

1 Introduction

The Gene Ontology (GO) has become one of the most widespread systems for systematically annotating gene products within the bioinformatics community and is developed by the Gene Ontology Consortium (14). It is specifically intended for describing gene products with a controlled and structured vocabulary. GO terms are part of a Directed Acyclic Graph (DAG), covering three orthogonal taxonomies or "aspects": *molecular function, biological process* and *cellular component.* Two different kinds of relationship between GO terms exist: the "is-a" relationship and the "part-of" relationship. Providing a standard vocabulary across any biological resources, the GO enables researchers to use this information for automated data analysis.

The *GOSim* package provides the researcher with various information theoretic similarity concepts for GO terms (9; 10; 6; 5; 7; 2; 3). It additionally implements different methods for computing functional similarities between gene products based on the similarties between the associated GO terms. This can, for instances, be used for clustering genes according to their biological function (13; 4) and thus may help to get a better understanding of the biological aspects covered by a set of genes.

2 Usage of *GOSim*

To elucidate the usage of *GOSim* we show an example workflow and explain the employed similarity concepts. We create a character vector of Entrez gene IDs:

```
> library(GOSim)
> genes = c("207", "208", "596", "901", "780", "3169", "9518",
+     "2852", "26353", "8614", "7494")
```

Next we investigate the GO annotation within the current ontology (which is *biological process* by default):

```
> getGOInfo(genes)
```

2.1 Term Similarities

Let us examine the similarity of the GO terms for genes ”8614” and ”2852” in greater detail:

```
> getTermSim(c("GO:0007166", "GO:0007267", "GO:0007584", "GO:0007165",
+     "GO:0007186"), method = "Resnik", verbose = FALSE)
```

```
           GO:0007166 GO:0007267 GO:0007584 GO:0007165 GO:0007186
GO:0007166  1.0000000  0.3225697  0.3225697  0.3317104  0.3787826
GO:0007267  0.3225697  1.0000000  0.3225697  0.3225697  0.3225697
GO:0007584  0.3225697  0.3225697  1.0000000  0.3225697  0.3225697
GO:0007165  0.3317104  0.3225697  0.3225697  1.0000000  0.3317104
GO:0007186  0.3787826  0.3225697  0.3225697  0.3317104  1.0000000
```

This calculates Resnik’s pairwise similarity between GO terms (9; 10):

$$sim(t,t') = IC_{ms}(t,t') := \max_{\hat{t} \in Pa(t,t')} IC(\hat{t}) \tag{1}$$

Here $Pa(t,t')$ denotes the set of all common ancestors of GO terms t and t', while $IC(t)$ denotes the information content of term t. It is defined as (e.g. (7))

$$IC(\hat{t}) = -\log P(\hat{t}) \tag{2}$$

i.e. as the negative logarithm of the probability of observing t. The information content of each GO term is already precomputed for each ontology based on the empirical observation, how many times a specific GO term or any of its direct or indirect offsprings appear in the annotation of the GO with gene products. The association between gene products and GO identifiers is reported regularily by the NCBI.

```
> data("ICsBPall")
> IC[c("GO:0007166", "GO:0007267", "GO:0007584", "GO:0007165",
+     "GO:0007186")]
```

```
GO:0007166 GO:0007267 GO:0007584 GO:0007165 GO:0007186
  5.540203   6.900127  10.776192   4.851709   5.945838
```

This loads the information contents of all GO terms within ”biological process”. Likewise, the data files `ICsMFall` and `ICsCCall` contain the information contents of all GO terms within ”molecular function” and ”cellular component”. If only GO terms having evidence codes ”IMP” (inferred from mutant phenotype), ”IGI”, (inferred from genetic interaction), ”IDA” (inferred from direct assay), ”IEP” (inferred from expression pattern) or ”IPI” (inferred from physical interaction) are wanted, one can use the data files `ICsBPIMP_IGI_IDA_IEP_IPI`, `ICsMFIMP_IGI_IDA_IEP_IPI` and `ICsCCIMP_IGI_IDA_IEP_IPI`, respectively. The information contents for GO terms filtered with respect to different evidence codes must be calculated explicitely using the function `calcICs`. Please refer to the manual pages for details.

For the similarity computation in (Eq.: 1) normalized information contents are used, which are obtained by dividing the raw information contents by its maximal value:

```
> IC[c("GO:0007166", "GO:0007267", "GO:0007584", "GO:0007165",
+     "GO:0007186")]/max(IC[IC != Inf])
```

```
GO:0007166 GO:0007267 GO:0007584 GO:0007165 GO:0007186
 0.3787826  0.4717604  0.7367662  0.3317104  0.4065158
```

To continue our example from above, let us also calculate Jiang and Conrath's pairwise similarity between GO terms, which is the default, for compairson reasons (5):

```
> getTermSim(c("GO:0007166", "GO:0007267", "GO:0007584", "GO:0007165",
+     "GO:0007186"), verbose = FALSE)
```

```
           GO:0007166 GO:0007267 GO:0007584 GO:0007165 GO:0007186
GO:0007166  1.0000000  0.7945964  0.5295906  0.9529278  0.9722668
GO:0007267  0.7945964  1.0000000  0.4366129  0.8416687  0.7668633
GO:0007584  0.5295906  0.4366129  1.0000000  0.5766628  0.5018574
GO:0007165  0.9529278  0.8416687  0.5766628  1.0000000  0.9251946
GO:0007186  0.9722668  0.7668633  0.5018574  0.9251946  1.0000000
```

Jiang and Conrath's similarity measure is defined as

$$sim(t,t') = 1 - \min(1, IC(t) - 2IC_{ms}(t,t') + IC(t')) \tag{3}$$

i.e. the similarity between t and t' is 0, if their normalized distance is at least 1.

Likewise, we can also compute Lin's pairwise similarity between GO terms (6):

```
> getTermSim(c("GO:0007166", "GO:0007267", "GO:0007584", "GO:0007165",
+     "GO:0007186"), method = "Lin", verbose = FALSE)
```

```
           GO:0007166 GO:0007267 GO:0007584 GO:0007165 GO:0007186
GO:0007166  1.0000000  0.7585030  0.5783157  0.9337471  0.9646845
GO:0007267  0.7585030  1.0000000  0.5338231  0.8029407  0.7345519
GO:0007584  0.5783157  0.5338231  1.0000000  0.6037937  0.5642872
GO:0007165  0.9337471  0.8029407  0.6037937  1.0000000  0.8986687
GO:0007186  0.9646845  0.7345519  0.5642872  0.8986687  1.0000000
```

It is defined as:

$$sim(t,t') = \frac{2IC_{ms}(t,t')}{IC(t) + IC(t')} \tag{4}$$

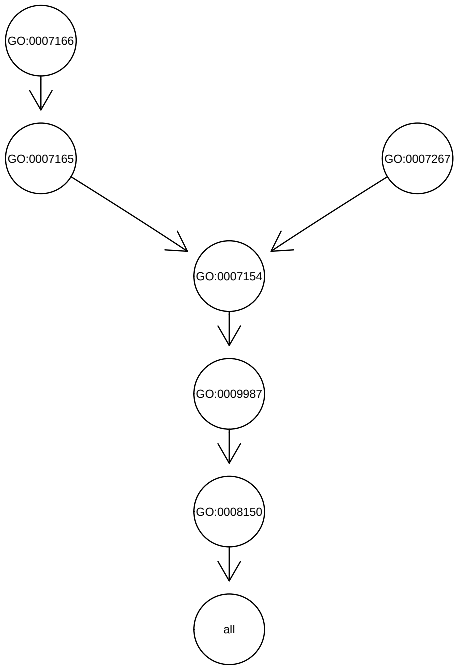

Figure 1: Example of a GO graph starting with leaves GO:0007166 and GO:0007267.

Resnik's, Jiang-Conraths's and Lin's term similarities all refer to $IC_{ms}(t, t')$, the information content of the minimum subsumer of t and t', i.e. of the lowest common ancestor in the hierarchy. For illustration let us plot the GO graph with leaves GO:0007166 and GO:0007267 and let us compute their minimum subsumer (see Fig. 1):

```
> library(Rgraphviz)
> G = getGOGraph(c("GO:0007166", "GO:0007267"))
> plot(G)

> getMinimumSubsumer("GO:0007166", "GO:0007267")

[1] "GO:0007154"
```

In contrast to the above defined similarity measures Couto et al. (3) introduced a concept, which is not based on the minimum subsumer, but on the set of all disjunctive common ancestors. Roughly speaking, the idea is not to consider the common ancestor having the highest information content only, but also others, if they are somehow "separate" from each other, i.e. there is a path to t and t' not passing any other of the disjunctive common ancestors.

```
> getDisjCommAnc("GO:0007166", "GO:0007267")
```

```
[1] "GO:0007154"
```

In this case the set of disjunctive common ancestors only consists of the minimum subsumer, because any path from the other ancestors to GO:0007166 and GO:0007267 would have to pass the minimum subsumer (see Fig. 1).

Based on the notion of disjunctive common ancestors Resnik's similarity concept can be extended by defining:

$$sim(t,t') = IC_{share}(t,t') = \frac{1}{|DisjCommAnc|} \sum_{t \in DisjCommAnc} IC(t) \qquad (5)$$

Likewise, Jiang-Conraths's and Lin's measures can be extended as well by replacing $IC_{ms}(t,t')$ by $IC_{share}(t,t')$.

```
> getTermSim(c("GO:0007166", "GO:0007267", "GO:0007584", "GO:0007165",
+     "GO:0007186"), method = "CoutoResnik", verbose = FALSE)

           GO:0007166 GO:0007267 GO:0007584 GO:0007165 GO:0007186
GO:0007166  1.0000000  0.3225697  0.2237777  0.3225697  0.3317104
GO:0007267  0.3225697  1.0000000  0.2237777  0.3225697  0.3225697
GO:0007584  0.2237777  0.2237777  1.0000000  0.2237777  0.2237777
GO:0007165  0.3225697  0.3225697  0.2237777  1.0000000  0.3225697
GO:0007186  0.3317104  0.3225697  0.2237777  0.3225697  1.0000000
```

Finally, it should be mentioned that also the depth and density enriched term similarity by Couto et al. (2) has been integrated into *GOSim*:

```
> setEnrichmentFactors(alpha = 0.5, beta = 0.3)
> getTermSim(c("GO:0007166", "GO:0007267", "GO:0007584", "GO:0007165",
+     "GO:0007186"), method = "CoutoEnriched", verbose = FALSE)

           GO:0007166 GO:0007267 GO:0007584 GO:0007165 GO:0007186
GO:0007166  1.0000000  0.1477308  0.1666937  0.1216959  0.1517828
GO:0007267  0.1477308  1.0000000  0.1749312  0.1342918  0.1558092
GO:0007584  0.1666937  0.1749312  1.0000000  0.1536618  0.1746225
GO:0007165  0.1216959  0.1342918  0.1536618  1.0000000  0.1286809
GO:0007186  0.1517828  0.1558092  0.1746225  0.1286809  1.0000000
```

2.2 Functional Gene Similarities

The special strength of *GOSim* lies in the possibility not only to calculate similarities for individual GO terms, but also for genes based on their complete GO anntation. For this purpose three basic ideas have been implemented:

1. Maximum and average pairwise GO term similarity

2. Computation of a so-called *optimal assignment* of terms from one gene to those of another one (4).

3. Embedding of each gene into a feature space defined by the gene's similarity to certain prototype genes (13; 4). Within this feature space similarities naturally arise as dot products between the feature vectors. These dot products can be understood as so-called *kernel functions* (12), as used in e.g. Support Vector Machines (1).

2.3 Maximum and Average Pairwise GO Term Similarity

The idea of the maximum pairwise GO term similarity is straight forward. Given two genes g and g' annotated with GO terms $t_1, ..., t_n$ and $t'_1, ..., t'_m$ we define the functional similarity between between g and g' as

$$sim_{gene}(g, g') \quad = \quad \max_{\substack{i=1,,...,n \\ j=1,...,m}} \quad sim(t_i, t'_j) \tag{6}$$

where sim is some similarity measure to compare GO terms t_i and t'_j. The resulting value is then further normalized to account for an unequal number of GO terms for both genes:

$$sim_{gene}(g, g') \leftarrow \frac{sim_{gene}(g, g')}{\sqrt{sim_{gene}(g, g) sim_{gene}(g', g')}} \tag{7}$$

Instead of computing the maximum pairwise GO term similarity one may also take the average here.

2.3.1 Optimal Assignment Gene Similarities

To elucidate the idea of the optimal assignment, consider the GO terms associated with gene "8614" on one hand and gene "2852" on the other hand:

```
> getGOInfo(genes)[c("8614", "2852")]

$`8614`
$`8614`$`GO:0007166`
GOID = GO:0007166

Term = cell surface receptor linked signal transduction

Definition = Any series of molecular signals initiated by the binding
     of an extracellular ligand to a receptor on the surface of the
     target cell.
```

```
Ontology = BP


$`8614`$`GO:0007267`
GOID = GO:0007267

Term = cell-cell signaling

Synonym = cell-cell signalling

Definition = Any process that mediates the transfer of information from
     one cell to another.

Ontology = BP


$`8614`$`GO:0007584`
GOID = GO:0007584

Term = response to nutrient

Synonym = nutritional response pathway

Definition = A change in state or activity of a cell or an organism (in
     terms of movement, secretion, enzyme production, gene expression,
     etc.) as a result of a nutrient stimulus.

Ontology = BP



$`2852`
$`2852`$`GO:0007165`
GOID = GO:0007165

Term = signal transduction

Synonym = MAPKKK cascade during sporulation (sensu Saccharomyces)

Definition = The cascade of processes by which a signal interacts with
     a receptor, causing a change in the level or activity of a second
     messenger or other downstream target, and ultimately effecting a
     change in the functioning of the cell.

Ontology = BP


$`2852`$`GO:0007186`
GOID = GO:0007186

Term = G-protein coupled receptor protein signaling pathway

Synonym = GPCR protein signalling pathway

Definition = The series of molecular signals generated as a consequence
     of a G-protein coupled receptor binding to its physiological
     ligand.

Ontology = BP
```

Given a similarity concept *sim* to compare individual GO terms, the idea is now to assign each term of the gene having fewer annotation to exactly one term of the other gene such that the overall similarity is maximized. More formally this can be stated as follows: Let π be some permutation of either an n-subset of natural numbers $\{1, ..., m\}$ or an m-subset of natural numbers $\{1, ..., n\}$ (this will be clear from context). Then we are looking for the quantity

$$sim_{gene}(g, g') = \begin{cases} \max_\pi \sum_{i=1}^{n} sim(t_i, t'_{\pi(i)}) & \text{if } m > n \\ \max_\pi \sum_{j=1}^{m} sim(t_{\pi(j)}, t'_j) & \text{otherwise} \end{cases} \tag{8}$$

The computation of (8) corresponds to the solution of the classical maximum weighted bipartite matching (optimal assignment) problem in graph theory and can be carried out in $O(\max(n, m)^3)$ time (8). To prevent that larger lists of terms automatically achieve a higher similarity we should further normalize sim_{gene} according to (Eq. 7)

In our example, using Lin's GO term similarity measure the following assignments are found:

$$\mathtt{GO:0007165} \rightarrow \mathtt{GO:0007267} \tag{9}$$

$$\mathtt{GO:0007186} \rightarrow \mathtt{GO:0007166} \tag{10}$$

The resulting similarity matrix is:

```
> getGeneSim(c("8614", "2852"), similarity = "OA", similarityTerm = "Lin",
+     verbose = FALSE)

        8614    2852
8614 1.00000 0.72163
2852 0.72163 1.00000
```

Note the difference to a gene similarity that is just based on the maximum GO term similarity:

```
> getGeneSim(c("8614", "2852"), similarity = "max", similarityTerm = "Lin",
+     verbose = FALSE)

          8614      2852
8614 1.0000000 0.9646845
2852 0.9646845 1.0000000
```

2.3.2 Feature Space Embedding of Gene Products

To calculate the feature vectors for each gene we can either define certain prototype genes a priori or we use one of the heuristics implemented in the function `selectPrototypes`. The default behavior is to select the 250 best annotated genes, i.e. which have been annotated with GO terms most often:

```
> proto = selectPrototypes(verbose = FALSE)
```

We now calculate for each gene g feature vectors $\phi(g)$ by using their similarity to all prototypes $p_1, ..., p_n$:

$$\phi(g) = (sim'(g, p_1), ..., sim'(g, p_n))^T \tag{11}$$

Here sim' by default is the maximum pairwise GO term similarity. Alternatively, one can use the optimal assignment similarity for sim' as well. Both similarity measures can by itself again be combined with arbitrary GO term similarity concepts. The default is the Jiang-Conrath term similarity.

Because the feature vectors are very high-dimensional we usually perform a principal component analysis (PCA) to project the data into a lower dimensional subspace:

```
> PHI = getGeneFeaturesPrototypes(genes, prototypes = proto, verbose = FALSE)
```

This uses the above define prototypes to calculate feature vectors and performs a PCA afterwards. The number of principal components is chosen such that at least 95% of the total variance in feature space can be explained (this is a relatively conservatve criterion).

We can now plot our genes in the space spanned by the first 2 principal components to get an impression of the relative ”position” of the genes to each other in the feature space (see Fig. 2). The feature vectors are normalized to Euclidian norm 1 by default:

```
> x = seq(min(PHI$features[, 1]), max(PHI$features[, 1]), length.out = 100)
> y = seq(min(PHI$features[, 2]), max(PHI$features[, 2]), length.out = 100)
> plot(x, y, xlab = "principal component 1", ylab = "principal component 2",
+     type = "n")
> text(PHI$features[, 1], PHI$features[, 2], labels = genes)
```

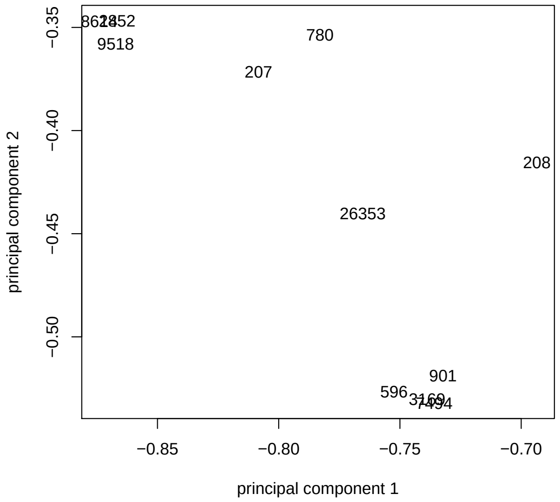

Figure 2: Embedding of the genes into the feature space spanned by the first 2 principal components

Finally, we can directly calculate the similarities of the genes to each other, this time using the Resnik's GO term similarity concept. These similarities may then be used to cluster genes with respect to their function:

```
> sim = getGeneSimPrototypes(genes, prototypes = proto, similarityTerm = "Resnik",
+     verbose = FALSE)
> h = hclust(as.dist(1 - sim$similarity), "ward")
> plot(h, xlab = "")
```

This produces a hierarchical clustering of all genes using Ward's method (see Fig. 3).

It should be mentioned that up to now all similarity computations were performed within the ontology ”biological process”. One could imagine to combine functional similarities between gene products with regard to different taxonomies. An obvious way for doing so would be to consider the sum of the respective similarities:

$$sim_{total}(g,g') = sim_{Ontology1}(g,g') + sim_{Ontology2}(g,g') \tag{12}$$

Of course, one could also use a weighted averaging scheme here, if desired.

2.4 Cluster Evaluations

GOSim has the possibility to evaluate a given clustering of genes or terms by means of their GO similarities. Supposed, based on other experiments (e.g. microarry), we have

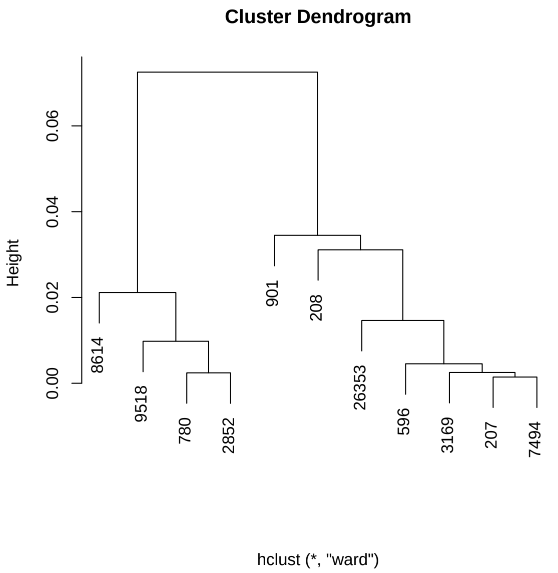

Figure 3: Possible functional clustering of the genes using Ward's method.

decided to put genes ”8614”, ”9518”, ”780”, ”2852” in one group, genes ”3169”, ”207”, ”7494”, ”596” in a second and the rest in a third group. Then we can ask ourselves, how similar these groups are with respect to their GO annotations:

```
> ev = evaluateClustering(c(2, 3, 2, 3, 1, 2, 1, 1, 3, 1, 2), sim$similarity)
> plot(ev$clustersil, main = "")
```

A good indiciation of the clustering qualitiy can be obtained by looking at the cluster silhouettes (11) (see Fig. 4). This shows that clusters 1 and 2 are relatively homogenous with respect to the functional similarity of the genes contained in it, while the genes in cluster 3 are more dissimilar.

References

[1] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273 – 297, 1995.

[2] F. Couto, M. Silva, and P. Coutinho. Implementation of a Functional Semantic Similarity Measure between Gene-Products. Technical Report DI/FCUL TR 03–29, Department of Informatics, University of Lisbon, 2003.

[3] F. Couto, M. Silva, and P. Coutinho. Semantic Similarity over the Gene Ontology: Family Correlation and Selecting Disjunctive Ancestors. In *Conference in Information and Knowledge Management*, 2005.

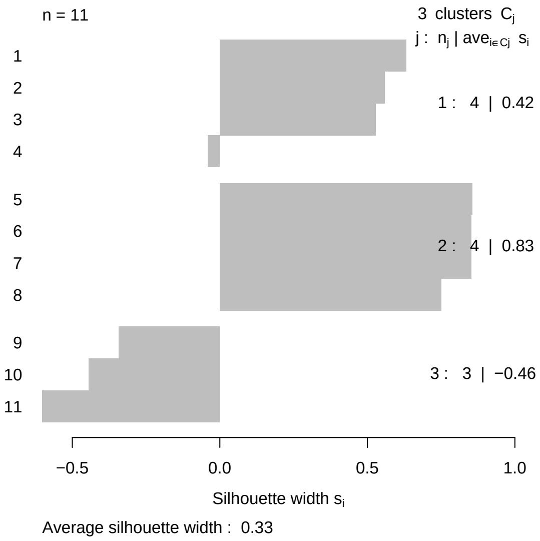

Figure 4: Silhouette plot of a possible given grouping of genes.

[4] H. Fröhlich, N. Speer, and A. Zell. Kernel based functional gene grouping. In *Proc. Int. Joint Conf. Neural Networks*, pages 6886 – 6891, 2006.

[5] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, Taiwan, 1998.

[6] D. Lin. An information-theoretic definition of similarity. In Morgan Kaufmann, editor, *Proceedings of the 15th International Conference on Machine Learning*, volume 1, pages 296–304, San Francisco, CA, 1998.

[7] P.W. Lord, R.D. Stevens, A. Brass, and C.A. Goble. Semantic similarity measures as tools for exploring the gene ontology. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 601–612, 2003.

[8] K. Mehlhorn and S. Näher. *The LEDA Platform of Combinatorial and Geometric Computing.* Cambridge University Press, 1999.

[9] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, volume 1, pages 448–453, Montreal, 1995.

[10] P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambigiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

[11] P.J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comp. and Applied Mathematics*, 20:53–65, 1987.

[12] B. Schölkopf and A. J. Smola. *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.

[13] N. Speer, H. Fröhlich, C. Spieth, and A. Zell. Functional grouping of genes using spectral clustering and gene ontology. In *Proc. Int. Joint Conf. Neural Networks*, pages 298 – 303, 2005.

[14] The Gene Ontology Consortium. The gene ontology (GO) database and informatics resource. *Nucleic Acids Research*, 32:D258–D261, 2004.