

# A New Method for Measuring the Semantic Similarity on Gene Ontology

Ying Shen, Shaohong Zhang, Hau-San Wong

Department of Computer Science, City University of Hong Kong, Hong Kong  
 yingshen3,shazhang@student.cityu.edu.hk, cshswong@cityu.edu.hk

## Abstract

*Semantic similarity defined on Gene Ontology (GO) aims to provide the functional relationship between different biological processes, molecular functions, or cellular components. In this paper, a novel method, namely the Shortest Path (SP) algorithm, for measuring the semantic similarity on GO is proposed based on both the GO structure information and the term's property. The proposed algorithm searches for the shortest path that connects two terms and uses the sum of weights on the shortest path to compute the semantic similarity for GO terms. A method for evaluating the nonlinear correlation between two variables is also introduced for validation. Extensive experiments conducted on two public gene expression datasets demonstrate the overall superiority of SP method over the other state-of-the-art methods evaluated.*

## 1. Introduction

Gene Ontology (GO) [16] is a structured and controlled vocabulary, which characterizes the functional properties of gene products using standardized terms. It is organized in the form of a Directed Acyclic Graph (DAG) and consists of three ontologies: biological process (BP), molecular function (MF), and cellular component (CC). GO data provide useful information (e.g. the quantitative relationships between gene products) for many applications. This information has been used to facilitate the existing clustering and classification algorithms [2, 9, 17] in bioinformatics.

Semantic similarity is an important type of information derived from GO, the concept of which is originally used in the field of linguistics. When GO emerged, the previously defined methods in linguistics were used directly to measure the similarity over the terms in GO (e.g. Resnik's method [11] and Lin's method [7]). In 2003, Lord *et al.* [8] found that the semantic similarity calculated from annotations correlates well with the sequence similarity. After that,

many new approaches have been proposed specifically for measuring the semantic similarity on Gene Ontology. Although new methods are proposed from time to time, they all have their own advantages and limitations and there is still a large scope for improvement.

Semantic similarity can be defined for both the GO terms and gene products. The state-of-the-art methods for semantic similarity over the GO terms can be divided into three categories: edge-based, node-based, and a hybrid of the above two. For the edge-based approaches, they mainly consider the length of paths connecting the terms [1, 10]. For the node-based methods, they rely on the property of the terms which is represented using the concept derived from information theory [6, 7, 11, 12]. There are also hybrid methods that consider both the substructure of GO and the property of terms involved [18].

Based on the semantic similarity over terms, the semantic similarity for gene products can be defined as the maximum ('Max') [19], or average ('Ave') [17] value of the semantic similarity between their annotations. In addition to the 'Max' and 'Ave' methods, there are some more complicated methods proposed in [12, 18]. The details of these methods will be described in Section 2.

To evaluate the previously defined methods, manually curated information and experimental data (e.g. protein-protein interaction (PPI) [20], pathway information [18], and gene expression data [20]) have been used.

In this work, we propose a new hybrid method for measuring the semantic similarity over the terms. The new method is proposed based on the observation that, if two terms diverge at the higher levels of GO, the discrepancy between the functions represented by the terms should be larger, and *vice versa*. The new method aims to find a path connecting the terms and uses a metric defined on the path to reflect such change. In addition to the proposed method for semantic similarity measurement, we also introduce another validation approach to replace the previously used Pearson correlation coefficient.

The rest of the paper is organized as follows. Section 2 introduces some representative methods for semantic similarity computation over terms and genes. Section 3 presents the algorithm proposed by us for measuring the semantic similarity over terms. Section 4 reports the experimental results. Finally, Section 5 concludes the paper with a summary.

## 2. Related work

Although there have been a lot of methods to calculate the semantic similarity over terms and genes, a detailed review is beyond the scope of this paper. We will only review several representative ones frequently used in previous research.

The methods for semantic similarity computation over terms generally fall into three categories: edge-based, node-based, and a hybrid of the former two.

Edge-based methods are intuitive and [10] [1] are two representative ones. Suppose  $t_1$  and  $t_2$  are two terms, and  $t$  is their lowest common ancestor. The *distance* method [10] counts the number of edges connecting the root with  $t$ , and edges connecting  $t$  with  $t_1$  and  $t_2$ . The distance between  $t_1$  and  $t_2$  is calculated using Eq. (1) below and can be easily converted to a similarity value:

$$dist(t_1, t_2) = \frac{len(root, t)}{len(root, t) + len(t, t_1) + len(t, t_2)} \quad (1)$$

where  $len(x, y)$  is the length of the path between the node  $x$  and  $y$ , represented by the number of edges on the path. The *distance* method assumes that the weight of each edge is always 1. Another edge-based algorithm [1] uses the average length of all paths that go through the longest partial path shared by two nodes. The edges are weighted using the depth information. The disadvantage of the edge-based methods is that, the weights of the edges on the same level are assumed to be the same. However, the terms on the same level of Gene Ontology do not necessarily have the same specificity, and the edges connecting two terms do not necessarily have the same weights. The edge-based algorithms cannot solve this problem.

Node-based methods focus mainly on the property of the terms, which is represented in the form of *Information Content* (IC). The IC value for a term  $t$  is defined as

$$IC(t) = -\log p(t) \quad (2)$$

where  $p(t)$  is the probability of occurrence of the term  $t$  in a certain corpus (e.g. SGD database). All the node-based methods are defined based on the IC value of the terms involved.

Resnik's method [11] is the first one used to measure the semantic similarity for GO terms. In this

method, the semantic similarity for term  $t_1$  and  $t_2$  is defined as

$$sim_{Resnik}(t_1, t_2) = \max_{t \in ancestor(t_1, t_2)} IC(t) \quad (3)$$

where  $t$  is the common ancestor of  $t_1$  and  $t_2$ . Term  $t$  with the largest IC value is also called the *most informative common ancestor* (MICA). Because Resnik's method only considers the information of the common ancestors, Lin's [7] and Jiang's [6] methods made some improvements by adding the IC values of the terms  $t_1$  and  $t_2$ . The models used in Lin's and Jiang's methods are shown in Eq. (4) and (5) below respectively.

$$sim_{Lin}(t_1, t_2) = \frac{2 \times IC(t)}{IC(t_1) + IC(t_2)}, \quad (4)$$

$$sim_{Jiang} = 1 - (IC(t_1) + IC(t_2) - 2 \times IC(t)), \quad (5)$$

where  $t$  is the MICA of  $t_1$  and  $t_2$ .

From Eqs. (4) and (5), it can be seen that, when  $t_1$  is equal to  $t_2$ , the semantic similarity of the two terms will correspond to the value of 1. However, because the terms on the top levels of GO are less specific than the leaf terms, the similarities between these terms and themselves should be smaller accordingly. To address this issue, Schlicker *et al.* [12] revised Lin's method by incorporating a weight item as shown in Eq. (6) below, and referred to it as the Relevance method:

$$sim_{rel}(t_1, t_2) = sim_{Lin}(t_1, t_2) \times (1 - p(t)) \quad (6)$$

In addition to the edge-based and node-based methods, there are also a number of hybrid methods proposed, e.g. Wang's method [18]. In Wang's method, a term  $t$  can be represented as a DAG structure. The semantic similarity of two terms is computed based on their DAG structures. Due to space limitation, further details will not be described here.

After calculating the semantic similarity over the terms, the next step is to define a measure for the semantic similarity over gene products. The often used methods are the 'Max' [19] and 'Ave' [17] method. Given two gene products  $g_1$  and  $g_2$ , the semantic similarities between their annotations form a semantic similarity matrix. For the 'Max' method, the semantic similarity is the maximum value in the matrix. For 'Ave' method, it is the average value over the whole matrix. They can be computed using Eq. (8) and (9) below respectively.

$$sim_{Max}(g_1, g_2) = \max_{\substack{t_1 \in annotation(g_1) \\ t_2 \in annotation(g_2)}} sim(t_1, t_2) \quad (8)$$

$$sim_{Ave}(g_1, g_2) = \text{average}_{\substack{t_1 \in annotation(g_1) \\ t_2 \in annotation(g_2)}} sim(t_1, t_2) \quad (9)$$

where the terms  $t_1$  and  $t_2$  are annotations for  $g_1$  and  $g_2$ . There are some more complicated methods defined for special structures, e.g. the method proposed by Wang *et al.* [18]. The evaluation of these methods shows that the similarity obtained using the 'Max' method is best

correlated with the gene expression data. However, the ‘Max’ method is more sensitive to outliers, while the ‘Ave’ method is relatively stable [20].

### 3. The Shortest Path algorithm

As mentioned in Section 2, for edge-based methods, the weights of the edges conflict with the property of GO, and for node-based methods, only IC values of the two terms and their MICA are considered regardless of their position in GO. To address these drawbacks, we propose a new hybrid method, namely the *Shortest Path* (SP) algorithm, to measure the semantic similarity over terms in GO.

It is intuitive that, given two terms  $t_A$  and  $t_B$ , if they diverge at a higher level (i.e. their MICA is nearer to the root), the difference between them should be larger; while if they diverge at a lower level, the difference should decrease. Under this assumption, the SP algorithm first assigns the weights to the terms in GO using the reciprocal of their IC values. Then the algorithm finds the path connecting the two terms and their MICA with the smallest sum of weights, and defines the sum of the weights on the path as the semantic distance for the two terms. This path is referred to as the shortest path. The rationale behind the algorithm is that, if MICA is near the root, the weights on the shortest path will increase, and *vice versa*. Therefore, the sum of the weights on the shortest path is consistent with the expected distance and can be used as its estimation.

The *Shortest Path* algorithm can be described as follows. Given two terms  $t_A$  and  $t_B$ , the normalized distance between them is defined as:

$$dist_{SP}(t_A, t_B) = \frac{\arctan\left(\sum_{t_1 \in path_A} \frac{1}{IC[t_1]} + \sum_{t_2 \in path_B} \frac{1}{IC[t_2]}\right)}{\pi/2} \quad (10)$$

where  $path_A$  ( $path_B$ ) is the shortest path that connects the term  $t_A$  ( $t_B$ ) with MICA;  $t_1$  and  $t_2$  are the terms located on  $path_A$  and  $path_B$ . Because MICA appears in both  $path_A$  and  $path_B$ , it is considered only once in Eq. (10). The function of  $\arctan$  is to normalize the distance obtained by summing the weights of the terms on the shortest path to  $[0, 1]$ . After the normalization, the semantic similarity can be defined as

$$sim_{SP}(t_A, t_B) = 1 - dist_{SP}(t_A, t_B) \quad (11)$$

Before using Dijkstra algorithm to find the shortest path between MICA and  $t_A$  ( $t_B$ ), we first assign the weight of each edge as the weight of the more specific term among the two that it connects to. Dijkstra algorithm is then used to find the shortest path on the new edge-weighted graph. In Eq. (10), when the weights on  $path_A$  or  $path_B$  increase, i.e. MICA and its

descendants on the shortest path become more general, the distance increases and the semantic similarity decreases.

We show an example of computing the semantic similarity for the terms GO:0006793 and GO:0008219 in Fig. 1. In the first step, the SP algorithm calculates the IC values for the terms on the graph using Eq. (2). In practice, though, several R packages (e.g. [4]) provide IC information for GO terms. Therefore, the IC values can be retrieved from these packages when needed. In this example,  $IC_{GO:0006793}=3.5402$  and  $IC_{GO:0008219}=3.5285$ . In the second step, SP algorithm weights each term using the value of  $1/IC$ , i.e.  $weight_{GO:0006793}=1/3.5402=0.2825$ , and  $weight_{GO:0008219}=1/3.5285=0.2834$ . The weights of the other terms are computed in a similar way. In the third step, SP algorithm finds the MICA (GO:0009987) and the shortest path connecting MICA and the two terms. In this example, there is only one path starting from GO:0006793 (GO:0008219) to MICA. Therefore, it is marked as the shortest path and shown in Fig. 1 in red. The distance between GO:0006793 and GO:0008219 is the normalized sum of weights on the shortest path, i.e.  $dist=\arctan(0.2825+0.7242+2.1179+0.2834)/(\pi/2)=0.8183$ . The semantic similarity  $sim$  is calculated as  $1-0.8183=0.1817$ .

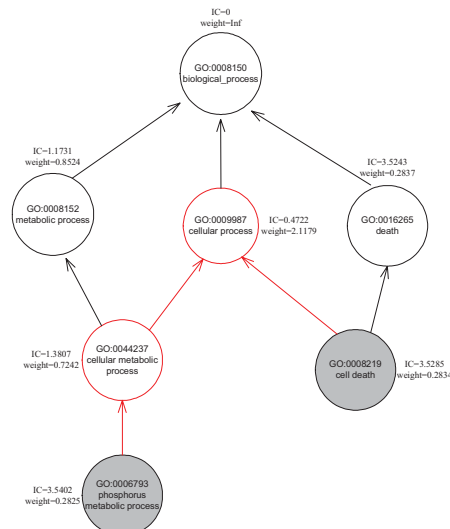


Fig. 1. An example of semantic similarity computation for GO:0008219 and GO:0006793.

SP algorithm integrates the information from two sources. They are the structure information contained in the paths connecting the terms, and the IC information of the terms represented by the weights on the graph. When searching for the shortest path, both structure and IC information will be considered, unlike the existing edge-based (node-based) methods that use only structure (IC) information.

## 4. An approach for nonlinear correlation

The Pearson correlation coefficient is often used to evaluate the linear dependency between two variables. When dealing with the nonlinear dependency problems, evaluation using Pearson correlation will not be suitable. Here we introduce an approach for characterizing nonlinear correlation [14].

Given a set of points  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ , the first step of the algorithm is to apply regression analysis to find a fitting curve  $f(x)$ . The curve corresponds to a nonlinear mapping between  $\mathbf{x}=\{x_1, \dots, x_n\}$  and  $\mathbf{y}=\{y_1, \dots, y_n\}$ . In the next step, the Pearson correlation coefficient is calculated between  $\mathbf{y}$  and  $f(\mathbf{x})$  after nonlinear regression. In addition to the Pearson correlation coefficient, another metric referred to as *root mean square error* (RMSE) is also calculated between  $\mathbf{y}$  and  $f(\mathbf{x})$  using Eq. (12) below. RMSE is used to measure the difference between values predicted by the model (i.e.  $f(\mathbf{x})$ ) and the observed values (i.e. the value of  $\mathbf{y}$ ). A smaller RMSE corresponds to a better prediction model.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - f(x_i))^2}{n}} \quad (12)$$

## 5. Experiments and results

In our experiments, we used the gene expression data to validate our algorithm following the way described in [20]. The goal of the validation is to find out whether the estimated semantic similarity is in line with the similarity based on the expression data. In general, a higher correlation indicates a better performance. We also evaluated other four state-of-the-art methods for measuring the semantic similarity over the terms. In Section 5.1, we will present the experimental details, including the description of the datasets and the experimental setup. Then, in Section 5.2, the experimental results together with some explanations will be given.

### 5.1. Data description and experimental setup

We used two datasets in our experiments for validation. The first one is the Eisen dataset [3], which consists of 2467 genes. The second one is the Spellman dataset [15], containing 6178 genes. The details of the datasets are described in Table 1. The missing values

in the two datasets were filled in using the *impute* package from the bioconductor project [5].

The annotations for genes were retrieved from the *Saccharomyces* Genome Database (SGD) [13]. In our experiments, we used the annotations from BP ontology. According to [20], terms at the top levels will create noise. Therefore, in our experiments, annotations at the first two levels were removed.

The similarity based on the gene expression data is calculated using the Pearson correlation and is referred to as the expression similarity.

We tested other four state-of-the-art methods including Resnik’s [11], Jiang’s [6], and the Relevance [12] method from the node-based category, and Wang’s [18] method from the hybrid category, as a comparison to the Shortest Path algorithm proposed by us. We used the *GOSim* package [4] to calculate the semantic similarity for Resnik’s, Jiang’s and the Relevance methods, and the *GOSemSim* package [21] for Wang’s method.

To compute the semantic similarity over gene products, we used the ‘Max’ operation for all the five methods, because it consistently results in the best correlation scores for all the methods measuring the semantic similarity over the terms [20].

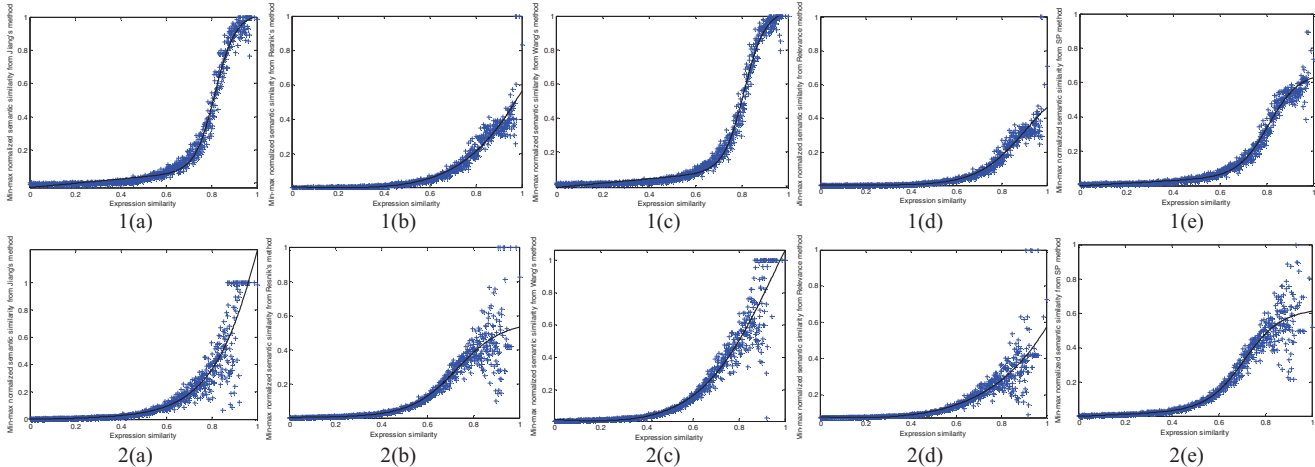
**Table 1: Datasets used in the experiments**

	# of genes	# of experiments	species
Eisen	2467	79	yeast
Spellman	6178	77	yeast

### 5.2. Experimental results

In the previous works, Pearson correlation was used to evaluate the consistency between the semantic similarity and the expression similarity. We first equally divided the interval  $[0, 1]$  into 1000 sub-intervals. Then we calculated the semantic similarity and the expression similarity for each gene pair. All the gene pairs were assigned to the sub-intervals according to their absolute gene expression similarity. After that, we calculated the average semantic similarity for each interval. Finally, the Pearson correlation coefficient value was calculated between the average semantic similarity and the expression similarity. The correlation coefficients for the different methods are shown in Table 2. From this table, it can be seen that the SP method proposed by us performs the best among the five methods. Specifically, our method achieves the correlation coefficient values of 82.8% and 86.9% on the Eisen and Spellman datasets respectively, which are about 3% higher than the second best methods (i.e. Wang’s method on the Eisen





**Fig. 2. Scatter plots of the min-max normalized semantic similarity versus the gene expression similarity on the Eisen and Spellman dataset. 1(a)-1(e) for Eisen dataset; 2(a)-2(e) for Spellman dataset. Five methods are compared: (a) Jiang’s method; (b) Resnik’s method; (c) Wang’s method; (d) Relevance method; (e) the Shortest Path method.**

dataset and Resnik’s method on the Spellman dataset), and a more significant improvement over the others.

Next, we calculated the nonlinear correlation coefficients and RMSE values for the average semantic similarity and the expression similarity using the method introduced in Section 4. The following mapping is used for the nonlinear regression analysis [14, 22]:

$$f(x) = a_1 \left( \frac{1}{2} - \frac{1}{1 + e^{a_2(x - a_3)}} \right) + a_4 x + a_5 \quad (13)$$

**Table 2: Pearson correlation coefficient**

	Jiang	Resnik	Wang	Relevance	SP
Eisen	0.7901	0.8031	0.7955	0.7817	<b>0.8278</b>
Spellman	0.7436	0.8383	0.8368	0.7718	<b>0.8685</b>

**Table 3:**

**Nonlinear correlation and RMSE on Eisen dataset**

	Jiang	Resnik	Wang	Relevance	SP
corr	0.9939	0.9602	<b>0.9950</b>	0.9581	<b>0.9892</b>
RMSE	0.0249	0.0299	0.0190	0.0321	<b>0.0166</b>

**Table 4:**

**Nonlinear correlation and RMSE on Spellman dataset**

	Jiang	Resnik	Wang	Relevance	SP
corr	0.9133	0.9295	0.9636	0.8869	<b>0.9656</b>
RMSE	0.0738	0.0518	0.0435	0.0553	<b>0.0354</b>

**Table 5:**

**Time consumption (sec) for example shown in Fig. 1**

	Jiang	Resnik	Wang	Relevance	SP
Time	0.02	0.01	0.42	0.01	0.15

The correlation coefficients and RMSE values on the Eisen dataset and the Spellman dataset are listed in Table 3 and Table 4 respectively. Fig. 2 shows the scatter plots of the min-max normalized gene semantic similarity versus the expression similarity for the five methods on the two datasets. Min-max normalization means that the minimum value is mapped to 0, the maximum value is mapped to 1, and the other values are linearly rescaled accordingly. In this way, the various scatter plots, which correspond to methods with different dynamic ranges for the semantic similarity values, can be more easily compared with each other. The curves are obtained by a nonlinear fitting using the model in Eq. (13).

Wang’s method and our SP method have quite similar performance with respect to the nonlinear correlation coefficients and RMSE scores, and they are much better than the other three methods. However, Wang’s method is heavily time-consuming. First, the IC value used in other methods is used to compute the “semantic value” in Wang’s method based on the contribution of the ancestors. Therefore, to obtain a term’s “semantic value”, we need to compute the contribution from all its ancestors in advance. In contrast, the IC value used by our SP method only relies on the number of genes annotated with the term, which can be quickly retrieved from the corpus. Second, the contribution from the same ancestor to different terms may not be the same. Therefore, the contribution values cannot be stored and reused for computing different term’s “semantic value”, resulting in computational redundancy. Third, when computing the semantic similarity, Wang’s method considers not only the semantic values of the two terms but also the

contribution from their common ancestors to the terms. As a comparison, our SP method only computes the sum of the reciprocal of the IC values of the terms on the shortest path, which can be performed more efficiently. To support these claims, we report the time consumption (in sec) in Table 5 for the five methods on the example shown in Fig. 1. It can be seen that the time consumption of Wang's method is about 3 times that of SP method.

## 5. Conclusion

In this paper, a novel method for measuring the semantic similarity, namely the *Shortest Path* (SP) algorithm, is proposed. The SP algorithm depends on the substructure of GO containing two terms and their MICA. The substructure contains more information than the single IC values used in the node-based algorithm. In addition, the weights assigned to the substructure are more consistent with common interpretation than the previous edge-based methods. In general, the semantic similarity obtained by SP algorithm better correlates with the expression similarity than other node-based methods. Moreover, compared with another state-of-the-art hybrid method, Wang's method, the SP algorithm has the advantage of less computation time due to fewer variables.

## 6. References

- [1] J. Cheng, M. Cline, J. Martin, D. Finkelstein, T. Awad, D. Kulp, and M. A. Siani-Rose, "A knowledge-based clustering algorithm driven by gene ontology", *Journal of Biopharmaceutical Statistics*, 14, pp. 687-700, 2004.
- [2] D. Dotan-Cohen, S. Kasif, and A. Melkman, "Seeing the forest for the trees: using the Gene Ontology to restructure hierarchical clustering", *Bioinformatics*, 25, pp. 1789-1795, 2009.
- [3] M. Eisen, P. Spellman, P. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns", *PNAS*, 95, pp. 14863-14868, 1998.
- [4] H. Froehlich, N. Speer, A. Poustka, T. Beissbarth, "GOSim - An R-Package for Computation of Information Theoretic GO Similarities Between Terms and Gene Products", *BMC Bioinformatics*, 8, pp. 166, 2007.
- [5] R. Gentleman, V. Carey, D. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge and J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry and F. Leisch, C. Li, M. Maechler, A. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Yang, and J. Zhang, "Bioconductor: Open software development for computational biology and bioinformatics", *Genome Biology*, 5, pp. R80, 2004.
- [6] J. Jiang and D. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy", *Proc. Int. Conf. on Research in Computational Linguistics*, pp. 19-33, 1997.
- [7] D. Lin, "An information-theoretic definition of similarity", *ICML*, pp. 296-304, 1998.
- [8] P. W. Lord, R. D. Stevens, A. Brass, and C. Goble, "Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation", *Bioinformatics*, 19, pp. 1275-1283, 2003.
- [9] G. Pandey, C. Myers, and V. Kumar, "Incorporating functional inter-relationships into protein function prediction algorithms", *BMC Bioinformatics*, 10, pp. 1471-2105, 2009.
- [10] V. Pekar and S. Staab, "Taxonomy learning: factoring the structure of a taxonomy into a semantic classification decision", *Proc. Int. Conf. on Computational Linguistics*, 2002.
- [11] P. Resnik, "Semantic similarity in taxonomy: An information-based measure and its application to problems of ambiguity in natural language", *Journal of Artificial Intelligence Research*, 11, pp. 95-130, 1999.
- [12] A. Schlicker, F. Domingues, J. Rahnenführer, and T. Lengauer, "A new measure for functional similarity of gene products based on gene ontology", *BMC Bioinformatics*, 7, pp. 302, 2006.
- [13] SGD project, <http://www.yeastgenome.org/>.
- [14] H. Sheikh, M. Sabir, and A. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms", *IEEE Trans. IP*, 15, pp. 3440-3451, 2006.
- [15] P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher, "Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomyces cerevisiae* by Microarray Hybridization", *Molecular Biology of the Cell*, 9, pp. 3273-3297, 1998.
- [16] The Gene Ontology Consortium, "Gene ontology: tool for the unification of biology", *Nature Genetics*, 25, pp. 25-29, 2000.
- [17] H. Wang, F. Azuaje, and O. Bodenreider, "An ontology-driven clustering method for supporting gene expression analysis, Computer-based Medical Systems", *Proc. IEEE Symposium on Computer-based Medical Systems*, pp. 23-24, 2005.
- [18] J. Wang, Z. Du, R. Payattakool, P. S. Yu, and C. Chen, "A new method to measure the semantic similarity of GO terms", *Bioinformatics*, 23, pp. 1274-1281, 2007.
- [19] H. Wu, Z. Su, F. Mao, V. Olman, and Y. Xu, "Prediction of functional modules based on Comparative genome analysis and Gene Ontology application", *Nucleic Acid Research*, 33, pp. 2822-2837, 2005.
- [20] T. Xu, L. Du, and Y. Zhou, "Evaluation of GO-based functional similarity measures using *S. cerevisiae* protein interaction and expression profile data", *BMC Bioinformatics*, 9, pp. 472, 2008.
- [21] G. Yu, F. Li, Y. Qin, X. Bo, Y. Wu, and S. Wang, "GOSemSim: an R package for measuring semantic similarity among GO terms and gene products", *Bioinformatics*, 26, pp. 976-978, 2010.
- [22] L. Zhang, L. Zhang, and X. Mou, "RFSIM: a feature based image quality assessment metric using Riesz transforms", *ICIP*, 2010.