GOtoGene: A Method for Determining the Functional Similarity among Gene Products

Kamal Taha
Department of Electrical and Computer Engineering
Khalifa University of Science, Technology, and Research
Box 127788, Abu Dhabi, UAE
E-mail: kamal.taha@kustar.ac.ae

Abstract

We present in this paper novel techniques that determine the semantic relationships among genes and gene products. We implemented these techniques in a middleware system called GOtoGene, which resides between user application and Gene Ontology database. Given a set *S* of genes, GOtoGene would return another set S' of genes, where *each* gene in S' is semantically related to *each* gene in S. The framework of GOtoGene refines the concept of Lowest Common Ancestor by defining the concept of Semantically Relevant Lowest Common Ancestor using the concept of existence dependency. We evaluated GOtoGene experimentally and compared it with three other methods. Results showed marked improvement.

Keywords: middleware; Gene Ontology; semantic similarity

1 Introduction

Life science ontologies are used in different types of applications. One of these applications is the annotation of biological objects such as gene products and proteins. Biological objects are annotated with ontology concepts to semantically describe their properties. The Gene Ontology (GO) (Gene Ontology 2011) has emerged as one of the most important ontology concept and the most widely used bio-ontology. Many genomic databases use GO annotations, which assign genes to term nodes to describe these genes. GO ontology is structured as a Directed Acyclic Graphs (DAG). In this graph, GO terms are represented by nodes and the different hierarchical relations between the terms *(mostly "is-a" and "part-of" relations)* are represented by edges. The "is-a" relation represents the fact that a given child term is a subtype of a parent term, and the "part-of" relation represents part-whole relationships. The lower in the DAG a term is located, the more specific it is. When a gene product is annotated using GO, the DAG displays the term node(s) describing this gene product in such a way that reflects how this gene product is related to other gene products. Thus, annotation of a gene with a GO term is an indicative that this gene is closely related to all other genes annotated with the same GO term and the genes annotated with ancestors and descendants of this GO term.

Biologists often need to determine the semantic similarities and relationships between genes. Semantic similarity measures in GO is widely used to identify the relationships between genes and gene products. That is because genes whose GO terms are semantically related tend to have common properties. The correlation between protein/gene expression and GO semantic similarities have been demonstrated in several studies such as (Sevilla et al. 2005, Wang et al. 2007). Functional similarity describes the similarity between genes/gene products based on the similarity between the GO terms annotating these genes/gene products. The similarity between two genes is the maximal semantic similarity of two GO terms, where one of the terms annotates one of the genes and the other annotates the other gene. That is, determining the relationships between GO terms enables the quantification of the semantic similarity of the gene products annotated with these terms. Thus, functional similarity between genes can be determined using a semantic similarity measure, since GO terms are organized in DAG.

The semantic relationships between a set of genes corresponds to that between the GO terms describing these genes, if each gene is annotated by only one GO term. But most genes have several annotation GO terms. Therefore, we need a strategy and mechanism to determine the relationships between *all the occurrences* of genes under consideration. In this paper, we propose a middleware system called GOtoGene that determines the semantic relationships between gene products and considers all the annotation terms of each gene product.

Given a set *S* of genes, GOtoGene would return a another set S' of genes, where *each* gene in S' is semantically related to *each* gene in S. Towards this, GOtoGene would identify the GO terms that have the closest semantic relationships with *all* GO terms annotating the genes in set *S*. It would first determine the most *significant* Lowest Common Ancestor (LCA) term of the terms annotating the genes in *S*. Towards this, the framework of GOtoGene refines the concept of LCA by defining the concept of Relevant Lowest Common

Ancestor (RLCA) and the concept of Semantically Relevant Lowest Common Ancestor (SRLCA). We observe that the terms annotating a certain set of genes have *existence dependency relationships* with the SRLCA *t* of these terms. That is, their existence in the GO graph is dependent on the existence of *t*. We developed this observation into formal sets of rules and techniques that compute the semantic relationships among GO terms.

2 Relates WORK

Pesquita (2008, 2009) defines a semantic similarity measure as a function that returns a numerical value reflecting the closeness in meaning between two ontology terms (or two sets of terms) annotating two biological entities (Pesquita et al. 2009). The authors distinguish between the comparison of two ontology terms and two sets of ontology terms. GOAT (Bada et al. 2004) proposes the mining of the *Gene Ontology Annotation* (GOA) of a database for co-occurrence of GO terms in order to acquire associations between the terms. Using this method, 600,000 associations were identified, excluding unreliable associations as well as the hierarchical relations that are explicitly represented in GO.

Node-based measures are the most cited semantic similarity measures. This approach exploits the information content (IC) of two terms being compared and of their Lowest Common Ancestor (LCA) (Coute et al. 2003, Lee et al. 2004, Lin 1998, Resnik 1999). The information content of a term is based on its frequency or probability of occurring in a corpus. Resnik (1999) uses the negative logarithm of the probability of a term to quantify its information content. Thus, a term with a high probability of occurring has a low IC. Very specific terms that are less frequent have a high IC. Resnik's similarity measure consists of determining the IC of all common ancestors of two terms and selecting the one with maximal value, since it is the most specific common ancestor of the two terms. That is, if two terms have an ancestor with high information content, they are considered to be semantically related. Since the maximum of this IC value can be greater than one, Lin (1998) introduced a normalization term into Resnik's measure. Schlicker (2006) improved Lin's measure by using a correction factor based on the probability of occurrence of the LCA. A general ancestor of terms should not have high contribution to the similarity of the terms (Schlicker et al. 2006). GOSim (Frohlich 2007) extended Resnik's similarity concept by considering *all* terms having the highest information content, based on the notion of disjunctive common ancestors. Lord (2003) computes the *information content* for each GO term as a measure of the degree of its specificity. A term that describes many genes (i.e., frequently used) is not specific and vice versa. Therefore, (Lord 2003) uses the negative logarithm of the frequency of each term to quantify its *information content.*

However, node-based measures have limitations such as: (1) they do not take into account the *distance* separating term nodes from their LCA (Frohlich 2007), (2) they use IC as the major factor for determining the semantic similarity of term nodes, *which is inappropriate for some types/scenarios of biological ontologies*, (3) some of them rely only on the *number* of common ancestor nodes, while overlook their semantic contributions to the two nodes under consideration, and (4) many of these methods overlook the information contained in the structure of the ontology and concentrate only on the information content of a node. We take (Benabderrahmane et al. 2010, Frohlich 2007, Wang et al. 2007) as sample of current semantic similarity measures and overview them below.

*Similarity method proposed by (*Wang et al. 2007*):* The semantic similarity between terms *A* and *B*, $S_{GO}(A, B)$, is defined as: $S_{GO}(A,B)=\frac{\sum_{t\in T_A\cap T_B}(S_A(t)+S_B(t))}{SV(A)+SV(B)}$

$S_A(t)$ is the contribution of term t to the semantics of *A*:

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

w_e is the semantic contribution factor for edge *e* linking term *t* with its child term t' ($0< w_e <1$).

IntelliGO (Benabderrahmane et al. 2010): Given two terms t_i and t_j represented by their vectors $\vec{e}_i$ and $\vec{e}_j$ respectively, the dot product between the base vectors is: $\vec{e}_i * \vec{e}_j \frac{2*Depth(LCA)}{MinSPL(t_i,t_j)+2*Depth(LCA)}$

Depth(LCA) is function associating the LCA with its maximal depth. *MinSPL(t_i, t_j)* is the minimal shortest path length between t_i and t_j.

GOSim (Frohlich 2007): It extended Resnik's similarity concept by considering *all* terms having the highest information content, based on the notion of disjunctive common ancestors:

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

3 Outline of the Approach

In the framework of GOtoGene, the structure of GO is described in terms of a graph, which we call GO Graph. In this graph, GO terms are nodes and the relationships between the terms are edges. For example, Fig. 1 presents a fragment of a GO Graph showing the ontological relationships of 29 GO terms. GOtoGene accepts Keyword-based queries with the form *Q* ("g_1", "g_2", .., "g_n"), where g_i denotes a gene (or a gene product) keyword.

User selects an input *(the query, which is composed of genes that are annotated to GO).* GOtoGene would then map these genes to a set of GO terms. Let S_T denote these GO terms. GOtoGene would then determine the *Relevant Lowest Common Ancestors* (RLCA) of the set S_T. It would then find the *Semantically Relevant Lowest Common Ancestors* (SRLCA) of the set S_T. Let S_1 be a set of GO terms annotating a gene g_1 and let S_2 be another set of GO terms annotating a gene g_2. For each term T_i from set S_1, the concept of RLCA determines the most relevant term T_j from set S_2 to T_i and then identifies their

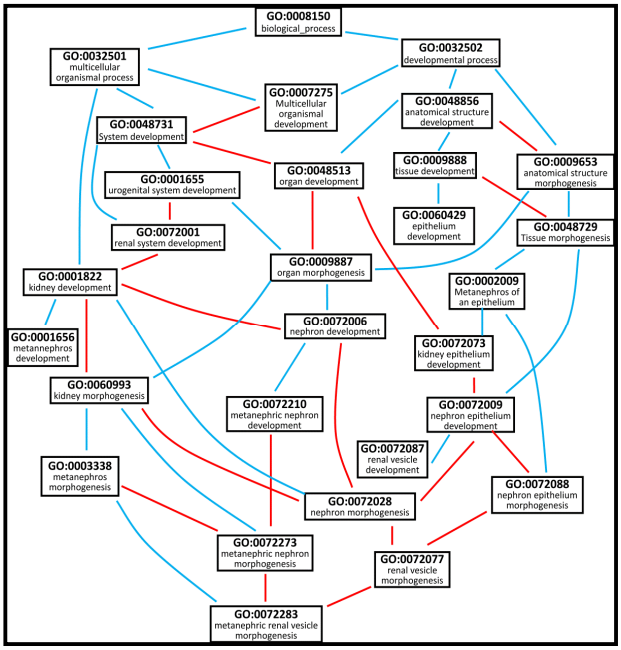

Fig. 1: A fragment of GO Graph showing the ontological relationships of 29 GO terms. *Blue edges denote "is-a" relations and red edges denote "part-of" relations.*

RLCA. If more than one RLCA have been identifies, the concept of SRLCA would identify the most significant one using the concept of *existence dependency*. That is, the SRLCA is a LCA, on which the *existence* of T_j and T_i depends in GO graph. GOtoGene would then convert back to genes based on annotations and retrieved back to the user. The genes annotated to the SRLCA are the most semantically related to the user's input genes.

> *Notation 3.1, Keyword Context (KC): A KC is a GO term that is annotated to a query gene product. For example, consider Fig. 1 and the query Q("JAG1"). The term organ morphogenesis (GO:0009887) is a KC because the gene "JAG1" is annotated to it.*

Let S_{KC} be a set of KCs annotating user's input genes (i.e., query). To construct the answer for this query, GOtoGene needs to identify the SRLCA of the set S_{KC} based on the concept of existence dependency. Towards this, GOtoGene will need to check all *"part-of"* relations in GO graph, because: "part of has a specific meaning in GO and a part of relation would only be added between A and B if B is necessarily part of A: wherever B exists, it is as part of A, and the presence of the B implies the presence of A" (Gene Ontology 2011). "part-of relation embodies some aspects of existence dependency. A part-of relation with existence dependent parts can simply be replaced by existence dependency: In case of existence dependent components, the existence dependency relation is identical to the part of relation" (Snoeck and Dedene 1998). Fig. 2 is an overview of our approach.

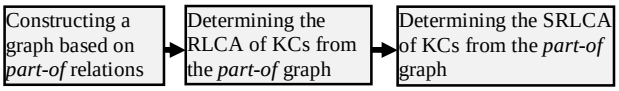

Fig 2: The sequential processing steps for answering a query

4 Constructing part-of Graph

Since not all "part-of" relations are explicitly expressed in a GO Graph (*some can be inferred from the graph*), GOtoGene converts the GO Graph into a graph called Part-Of Graph (POG), which contains only the explicit and inferred "part-of" relations. The LCA of KCs will be determined from the POG and not from the GO Graph. A POG is a GO Graph after: (1) removing all its relations except for the "part-of" ones, and (2) adding the inferred "part-of" relations. The terms *A* and *B* are connected by a "part-of" relation in the POG, if the GO Graph either states this relation expressly or it can be inferred from the graph using the following two *inference rules* described in (Gene Ontology 2011): (1) if *A* "is-a" *B* and *B* is "part-of" *C*, *A* is "part-of" *C*, and (2) if *A* is "part-of" *B* and *B* "is-a" *C*, *A* is "part-of" *C*. Fig. 3 shows a fragment of a POG derived from the GO Graph in Fig. 1. For example, since in Fig. 1: (1) the term multicellular organismal process (GO:0032501) "is-a" the term biological process (GO:0008150), (2) the term multicellular organismal development (GO:0007275) "is-a" the term multicellular organismal process (GO:0032501), and (3) the term system development (GO:0048731) is "part-of" the term multicellular organismal development (GO:0007275), then in Fig. 3 the term system development (GO:0048731) is "part-of" the term biological process (GO:0008150). In Fig. 3, each term node shows the genes that the term annotates.

We observe that the specificity of a term *(with regard to its "is-a" relations)* influences its semantic relationships with other terms. This specificity differentiates "general" functions that are close to the root from specific detailed ones. Therefore, we determine the specificity of each term. "*is-a*" is a simple type-subtype relation between two GO terms (Gene Ontology 2012). Consider that: (1) A' "is-a" *A*, (2) *A* "is-a" *C*, (3) B' "is-a" *B*, and (4) *B* "is-a" *C*. Both of the terms *A* and *B* inherit the characteristics and properties of their supertype *C*. Therefore, intuitively, *A* and *B* have the *same* specificity. Since A' and B' inherit from the characteristics and properties of terms that have the same specificity (*the terms A and B*), A' and B' have the *same* specificity also. *Thus, the* specificity *of a term node is the number of "is-a" relations that connect it with the root term node (its "is-a" distance to the root).* For example, recall Fig. 1. The root term biological process (GO:0008150) has its own specificity. Since both of the terms multicellular organismal process (GO:0032501) and developmental process (GO:0032502) inherit the same characteristics from their supertype GO:0008150, they both have the same specificity[1]. As another example, the terms kidney development (GO:0001822), system development (GO:0048731), multicellular organismal development (GO:0007275), anatomical structure morphogenesis (GO:0009653), and anatomical structure development (GO:0048856) have the same specificity[1]. *If a term has multiple inheritances, only its shortest distance to the root is*

[1] Alternatively, we can determine that these terms have the same specificity, because they have the same *distance to the root* based on their *is-a* relations.

considered. In the POG in Fig. 3, each set of terms that have the same specificity are colored with the same color for easy reference. For example, the terms kidney development (GO:0001822), system development (GO:0048731), and anatomical structure morphogenesis (GO:0009653) are colored with the same color as an indicative that they have the same specificity.

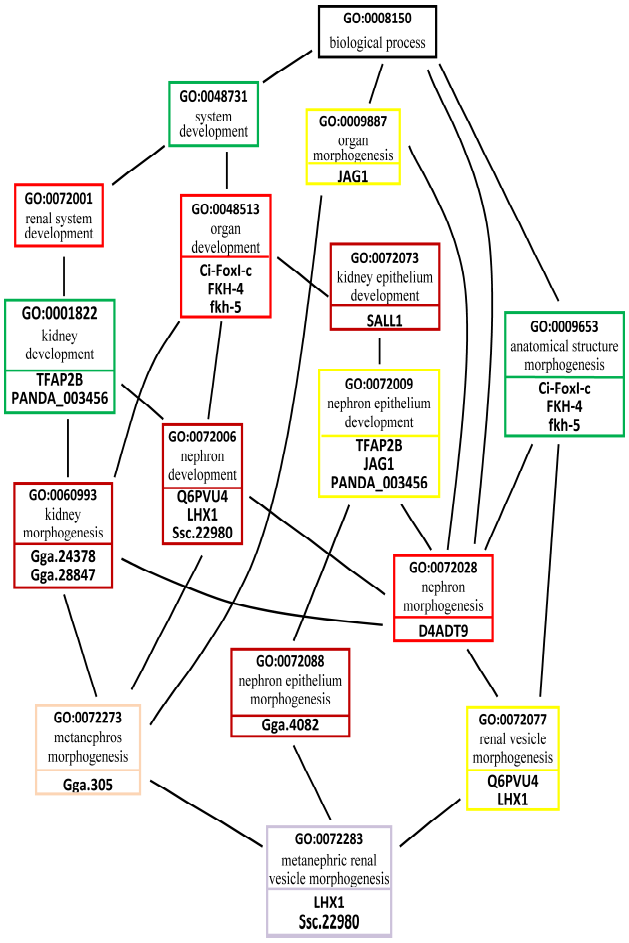

Fig. 3: POG constructed from the GO Graph in Fig. 1 after coloring each set of terms that have the same specificity with the same color. Each term node includes the genes annotated to the term.

5 Determining RLCA

In this section, we describe how GOtoGene determines the most *significant* Lowest Common Ancestor of terms annotating input genes (i.e., keywords of a query). We first formalize the notion of LCA in Definition 5.1.

Definition 5.1, Lowest Common Ancestor (LCA): Let: (1) N be the set of GO terms in a GO Graph, (2) T_i, T_j, $T_x \in N$, (3) T_x is the LCA of T_i and T_j (denoted as LCA(T_i, T_j)), and (4) descendant-or-self(T_i, T_x) denotes that T_i is a descendant of T_x or is equal to T_x. If:

- *descendant-or-self(T_i, T_x) = true, and*
- *descendant-or-self(T_j, T_x) = true, and*
- *$\forall\, T' \in N$, if descendant-or-self(T_i, T') = true and descendant-or-self(T_j, T') = true, then descendant-or-self(T_x, T') = true.*

We now introduce a notion for two or more terms that annotate at least one same gene product.

Notation 5.1, $ANTN_{T_x} = ANTN_{T_y}$: Denotes GO terms T_x and T_y annotate at least one same-gene. That is, there is at least one gene annotated to both T_x and T_y. For example, consider Fig. 3. The gene "LUX1" is annotated to GO terms GO:0072006, GO:0072077, and GO:0072283. Therefore, $ANTN_{\text{GO:0072006}} = ANTN_{\text{GO:0072077}} = ANTN_{\text{GO:0072283}}$

We now refine the concept of LCA and introduce the concept of Relevant Lowest Common Ancestor (RLCA). Let $g_1, g_2, \ldots g_n$ be a set of input genes selected by the user (i.e., keywords of a query). Let S be the set of terms annotating the input $g_1, g_2, \ldots g_n$. A RLCA is a LCA of a subset $S' \subseteq S$ where the terms of S' are *meaningfully* related to each other and contain at least one occurrence of each of the genes. We present below two scenarios that describe what we mean by *meaningfully* related terms.

Scenario 1: Consider the situation where two GO term nodes have no hierarchical relationship with each other. Suppose that the LCA of T_1 and T_2 is T_x as shown in Fig. 4. We can regard both T_1 and T_2 as *meaningfully* related to each other *by belonging* to T_x. The RLCA of T_1 and T_2 is T_x. Given two sets of GO terms, where the terms in each set fall under the same annotation cluster, Definition 4.3 describes how to determine the RLCA of each pair from the two sets.

Definition 5.2, RLCA of two nodes: Let the set of GO terms in a GO Graph be N. Given A, B $\subseteq$ N, where A is comprised of nodes having the same annotation A, and B is comprised of nodes having the same annotation B, the RLCA Set C $\subseteq$ N of A and B satisfies the following conditions:

- *$\forall\, c_k \in C$, $\exists\, a_i \in A$, $b_j \in B$, such that c_k = LCA(a_i,b_j). c_k is denoted as RLCA(a_i, b_j).*
- *$\forall\, a_i \in A$, $b_j \in B$, if d_{ij} = LCA(a_i, b_j) and $d_{ij} \notin C$, then $\exists\, c_k \in C$, descendant(c_k, d_{ij}) = true.*

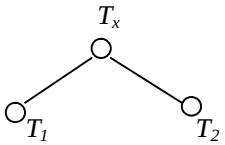

Fig. 4: Relationships between nodes located in adjacent hierarchical levels

Scenario 2: As demonstrated by Fig. 5, let there be two terms T_2' and T_2, where $ANTN_{T_2'} = ANTN_{T_2}$. Let the LCA of T_1 and T_2' be T'. If T_x is an ancestor of T', we should then conclude that nodes T_1 and T_2 are *not* meaningfully related to each other, because: (1) T_1 is more related to T_2' than to T_2 (recall that

$ANTN_{T_2'} = ANTN_{T_2}$), and (2) the LCA of T_1 and T_2 is an ancestor of the LCA of T_1 and T_2'. Term T' is the RLCA of T_1 and T_2'. We formalize this concept in Definition 5.3.

Definition 5.3 : *Let* $ANTN_A = ANTN_B \neq ANTN_C$. *Node C is relevantly related to node A and not to node B if the LCA of nodes A and C is a descendant of the LCA of nodes C and B. The LCA of nodes C and A is a RLCA.*

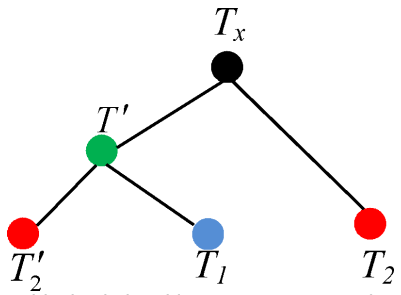

Fig. 5: Hierarchical relationships among term nodes of different specificities. *A node's color represents its specificity*

Example 1: Consider Fig. 3 and the query Q("LHX1", "Gga,4082"). As shown in Fig. 3: (1) the KCs annotating the gene "LHX1" are *nephron development* (GO:0072006), *renal vesicle* morphogenesis (GO:0072077), and *metanephric renal vesicle* morphogenesis (GO:0072283), and (2) the KC annotating the gene "Gga.4082" is nephron epithelium morphogenesis (GO:0072088). The term GO:0072088 is more related to GO:0072077 than to GO:0072006, because: (1) $ANTN_{GO:0072077} = ANTN_{GO:0072006}$, and (2) the LCA of GO:0072006 and GO:0072088 *(which is GO:0048513)* is an ancestor of the LCA of GO:0072088 and GO:0072077 *(which is GO:0072009)*. Therefore, GO:0072009 is the RLCA of GO:0072088 and GO:0072077 and the genes it annotates *(i.e., the genes "TFAP2B", "JAG1", and "PANDA_003456")* are related to both of the input gene keywords "Gga.4082" and "LHX1". Fig. 6 shows the subtree rooted at the RLCA node GO:0072009.

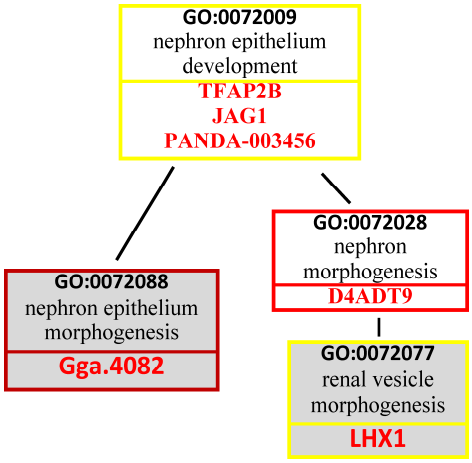

Fig. 6: GO:0072009 is a RLCA of GO:0072088 and GO:0072077

Example 2: Consider Fig. 3 and the keyword query Q_2("JAG1", "LHX1"). The GO terms annotating the gene "JAG1" are organ morphogenesis (GO:0009887) and nephron epithelium development (GO:0072009). Recall example 1 for the GO terms annotating the gene "LHX1". The term GO:0072006 is more related to GO:0072009 than to GO:0009887, because: (1) $ANTN_{GO:0072009} = ANTN_{GO:0009887}$, and (2) the LCA of GO:0072006 and GO:0009887 *(which is GO:0008150)* is an ancestor of the LCA of GO:0072006 and GO:0072009 *(which is GO:0048513)*. Therefore, GO:0048513 is the RLCA of GO:0072006 and GO:0072009 and the genes it annotates *(i.e., the genes Ci-FoxI-c, FKH-4, and fkh-5)* are related to both of the input genes "JAG1" and "LHX1". Fig. 7 shows the subtree rooted at the RLCA GO:0048513.

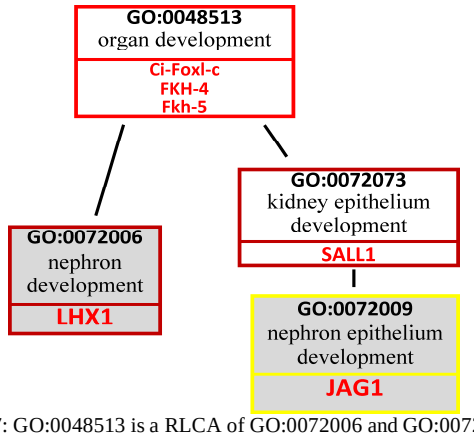

Fig. 7: GO:0048513 is a RLCA of GO:0072006 and GO:0072009

We constructed an algorithm called *GetRLCA* (see Fig. 8) that determines the RLCA for an input set of genes. Function *getAnnotations* (see line 1) returns the terms annotating the input genes and stores the results in set S. Line 2 stores in set S' the duplicate annotations in set S. In line 6, function *GetLCA* (see Fig. 9) returns the LCA of an input set of annotations, and function *getAncestors* returns all ancestors of the annotations. In lines 6 and 7, if the LCA of nodes n_i and $S'-S''$ is an ancestor of the LCA of nodes n_j and $S'-S''$, line 7 will return the later LCA as the RLCA.

GetRLCA $(g_1, g_2, \ldots, g_3)$ {
1. $S \leftarrow getAnnotations(g_1, g_2, \ldots, g_n)$
2. $S' \leftarrow S - distinct(S)$
3. *for each term node* $n \in S'$
4. $\quad S'' \leftarrow distinct(S) - annotation(n)$
5. $\quad$ *for each term nodes* $n_i, n_j \in S''$
6. *if* $GetLCA(n_i, (S'-S'')) \in getAncestors(GetLCA(n_j, (S'-S'')))$
7. $\quad RLCA \leftarrow GetLCA(n_j, (S'-S''))$
}

Fig. 8: Algorithm *GetRLCA*

GetLCA $(T_1, T_2, \ldots, T_n)$ {
1. $S \leftarrow getAncestors(T_1) \cap getAncestors(T_2) \ldots \cap getAncestors(T_n)$
2. *if* $(S \neq null)$
3. $\quad LCA(T_1, T_2, \ldots, T_n) \leftarrow getDescendantAll(S)$
}

Fig. 9: Subroutine *GetLCA*

6 Determining SRLCA

From the set of RLCAs, GOtoGene needs to determine the ones that are *semantically related* to the KCs. We call each one the Semantic Relevant Lowest Common Ancestor (SRLCA) of the KCs. A SRLCA should be semantically related to *each* of the KCs, otherwise the answer subtree is considered invalid.

We use the notation SR_{KC} to denote the set of terms that are *semantically related* to the KC. The notation $T \in SR_{KC}$ denotes that term T is semantically related to the KC. In order for a RLCA $\in SR_{KC}$ *(i.e., to be SRLCA)*, we observe that: (1) the RLCA should have a different specificity than the *KC*, and (2) the path from the RLCA to the KC in the GO Graph should not include two or more terms with the same specificity. For example, recall the query in example 1. As described in the example, the term GO:0072009 is a RLCA of the KCs GO:0072088 and GO:0072077. However, GO:0072009 $\notin SR_{GO:0072077}$, because its specificity is the same as the specificity of GO:0072077. Therefore, GO:0072009 is not a SRLCA for GO:0072077. Based on these observations, we introduce proposition 6.1 below.

> *Proposition 6.1: SRLCA: In order for a RLCA to be a SRLCA: (1) its* specificity *should be different than the* specificities *of the KCs, and (2) the path in the GO Graph from the RLCA to each of the KCs should not include two or more terms with the same specificity.*

> *Notation 6.1, $SPEC_x$: $SPEC_x$ denotes the* specificity *of GO term x.*

We prove observation/proposition 6.1 heuristically as follows. First, we prove: *if a RLCA* $\in SR_{KC}$, *then* $SPEC_{RLCA} \neq SPEC_{KC}$. That is, in order for a RLCA to be SRLCA, its specificity should be different than the specificity of the KC. We are going to validate this observation by checking whether it conforms to the structural characteristics of *existence dependency*. The concept of existence dependency was first proposed for Entity-Relationship modeling (Elmasri, and Navathe 2011). An object *x* is *existence-dependent* on an object *y* if the existence of *x* is dependent on the existence of *y* (Widjaya et al. 2003). The existence dependency concept and the SR_{KC} concept have correspondences: both denote that *an object(s) has a strong association with another object*. SR_{KC} is a set of GO terms, whose *existence* in a POG is *dependent* on the existence of the KC *(or conversely, the existence of the KC in the graph is dependent on the existence of the set of terms)*. Snoeck et al. (1998) argue that the existence dependency relation is a partial ordering of *object types (i.e., specificities)*. The authors transform an OO schema into a graph consisting of the *object types* found in the schema and their relations. The object types in the graph are related only through associations that express existence dependency. The authors demonstrated through the graph that *an object type is never existence-dependent on itself*. That is, if the two objects O_i and O_j belong to the same type, O_i cannot be dependent on O_j and vice versa. This finding is in agreement with our proposed rule, when we view: (1) a GO term in a GO Graph as an object, and (2) a GO term's specificity as an object's type. Thus, if a RLCA has the same specificity as the KC, the RLCA can never be existence-dependent on the KC (and vice versa); therefore, this RLCA is NOT SRLCA and the genes annotated to it may not be semantically related to the genes annotated to the KCs.

Second, we prove: *If a RLCA is semantically related to the KC, then* $SPEC_{Tx} \neq SPEC_{Ty}$ *where* T_x *and* T_y *are term nodes located between the RLCA and the KC in the POG*. We can verify this rule as follows. In order for *RLCA* $\in SR_{KC}$, all term nodes located between the *RLCA* and the KC in the POG have to be related to the KC. Let: (1) term $T_y \in SR_{KC}$, (2) T_y be a descendant of the KC, and (3) term T_x be a descendant of T_y. In order for T_x to be semantically related to the KC, intuitively T_x has to be semantically related T_y, because T_y relates (connects) T_x with the KC. If T_x and T_y have the same specificity, then $T_x \notin SR_{T_y}$ (according to the first rule). Therefore, in order for T_x to be semantically related to the KC, $SPEC_{Tx} \neq SPEC_{Ty}$.

Example 3: Recall example 1. By applying proposition 6.1, GO:0072009 is *NOT* a SRLCA for GO:0072088 and GO:0072077, because GO:0072009 and GO:0072077 have the same specificity. Therefore, the genes annotated to GO:0072009 *(i.e., the genes TFAP2B, JAG1, and PANDA_003456)* may not be *semantically* related to the input gene keywords "Gga.4082" and "LHX1".

Example 4: Recall example 2. By applying proposition 6.1, the RLCA of GO:0072088 and GO:0072077 (i.e., the term GO:0048513) is a SRLCA. Therefore, the genes annotated to GO:0048513 (*i.e., the genes Ci-FoxI-c, FKH-4, and fkh-5)* are *semantically* related to both of the input gene keywords "JAG1"and "LHX".

7 Experimental Results

We experimentally evaluated the quality of GOtoGene and compared it with (Benabderrahmane et al. 2010, Frohlich 2007, Wang et al. 2007) *(recall their descriptions in section 2)*. We implemented GOtoGene in Java, run on Intel(R) Core(TM)2 Dup CPU processor, with a CPU of 2.1 GHz and 3 GB of RAM, under Windows Vista. The implementation of (Frohlich 2007) was released as part of GOSim package (Cran 2012), which we used for the evaluation of (Frohlich 2007). We implemented the methods of (Benabderrahmane et al. 2010, Frohlich 2007, Wang et al. 2007) from scratch.

7.1 Benchmarking Datasets

Pathways are sets of genes shown to have high functional similarity and can be used to validate similarity measures (Guo 2006, Nagar and Al-Mubaid 2008, Wang 2004). A fully describe a pathway represent the dynamics and dependencies among a set of gene/gene products. Therefore, we used in our experiments pathways as a

reference for evaluating and comparing the similarity (and semantic relationships) measures of GOtoGene and (Benabderrahmane et al. 2010, Frohlich 2007, Wang et al. 2007). Given a set S of genes, a system/method should return a set S' of other genes that are semantically related to S. In order for sets S and S' to be related, S and S' should be part of a *same pathway*.

We used for the evaluation two different benchmarks: KEGG and Pfam benchmarks. We selected 15 groups of highly related Pfam entries *(see Table 1)* from the Sanger Pfam database. We selected a set of 15 human and 15 yeast diverse KEGG pathways *(see Tables 2 and 3)* containing between 10 and 30 genes, which were retrieved using the *DBGET* database. For each group, we retrieved the corresponding human and yeast gene identifiers from the Uniprot database. Assuming that genes belonging to the same KEGG pathway are often related to a similar biological process, the similarity values calculated for this dataset should be related to the BP GO aspect. And, assuming that genes which share common domains in a Pfam clan often have a similar molecular function, the similarity values calculated for this second dataset should be related to the MF GO aspect.

7.2 Evaluating Recall and Precision

We measured the *recall (or true positive rate)* and *precision* of GOtoGene and of (Benabderrahmane et al. 2010, Frohlich 2007, Wang et al. 2007). Recall *(or true positive rate)* is the *fraction* of correct genes determined by a similarity measure relevant to all genes determined by the measure. Precision is the *fraction* of correct genes determined by a similarity measure relevant to all genes in the pathway. Let: (1) G_P be all genes in the pathway and n be the number of these genes, and (2) G_M be the top n genes determined by a similarity measure, which are semantically related to the input gene keywords. recall = ($|G_M \cap G_P| / |G_M|$). Precision = ($|G_M \cap G_P| / |G_P|$). We computed the recall and precision for the four methods as follows. We first measured the semantic similarity/relationship between each term in the GO Graph and the set of terms annotating the input gene keywords using each of the four methods. We then clustered the genes based on the obtained similarity values. G_M are the top n genes in each cluster with the highest similarity values. As for GOtoGene, G_M are n genes annotated by the GO terms located in the paths from a SRLCA to the terms annotating the input gene keywords.

Fig. 10 shows the recall and precision results obtained with the KEGG pathways. Fig. 11 shows the recall and precision results obtained with the pfam pathways. For each KEGG and pfam pathway (x-axis), the recall and precision values are represented as histograms (y-axis). As the figures show, recall and precision values vary based on: (1) pathways, and (2) the accuracy of each of the four methods to capture the semantic similarities and relationships among gene annotations within pathways.

Table 1. The15 Pfam Human Pathways and the 15 Pfam Yeast Pathways used in the experiments

Pfam Accession	Pfam ID	Number of genes (human)	Number of genes (yeast)
CL0406	vWA-like	11	6
CL0344	4Fe-4S	7	4
CL0461	Metallothionein	18	11
CL0020	TPR	13	6
CL0418	GIY-YIG	8	19
CL0417	BIR-like	10	6
CL0233	SufE_NifU	9	10
CL0167	Zn_Beta_Ribbon	7	5
CL0099	ALDH-like	18	11
CL0042	Flavoprotein	10	7
CL0040	tRNA_synt_II	12	2
CL0179	ATP-grasp	7	6
CL0417	BIR-like	11	9
CL0445	SNARE-fusion	8	6
CL0444	YNI	9	5
Total number of genes		**158**	**113**

Table 2. The 15 KEGG Human Pathways used in the experiments.

Pathway	Name	# of genes
sce00562	Inositol phosphate metabolism	15
sce00920	Sulfur metabolism	15
sce00600	Sphingolipid metabolism	13
sce00410	beta-Alanine metabolism	12
sce00514	Saccharomyces cerevisiae	13
sce00670	One carbon pool by folate	15
sce00903	Limonene and pinene degradation	20
sce03022	Basal transcription factors	32
sce04130	SNARE interactions in vesicular transport	23
sce03450	Non-homologous end-joining	10
sce04070	Phosphatidylinositol signaling system	15
sce04140	Regulation of autophagy	17
sce04111	Saccharomyces cerevisiae	25
sce04011	MAPK signaling pathway	57
sce03010	Ribosome	12
Total number of genes		**294**

Table 3. The 15 KEGG Yeast Pathways used in the experiments.

Pathway	Name	# of genes
hsa00040	Pentose and glucuronate interconversions	34
hsa00920	Sulfur metabolism	14
hsa00140	Steroid hormone biosynthesis	26
hsa00290	Valine, leucine and isoleucine biosynthesis	5
hsa00563	Glycosylphosphatidylinositol	25
hsa00670	One carbon pool by folate	19
hsa00232	Caffeine metabolism	7
hsa03022	Basal transcription factors	23
hsa04130	SNARE interactions in vesicular transport	36
hsa03450	Non-homologous end-joining	13
hsa03430	Mismatch repair	23
Hsa00085	Fatty acid biosynthesis	12
hsa04950	Maturity onset diabetes of the young	25
hsa04803	Homo sapiens	16
hsa00120	Primary bile acid biosynthesis	14
Total number of genes		**292**

In summary, the recall and precision values for the two benchmarking datasets showed that GOtoGene outperforms the other three methods. The results reveal the robustness of the GOtoGene's method and its ability to reflect the semantic relationships among gene annotations.

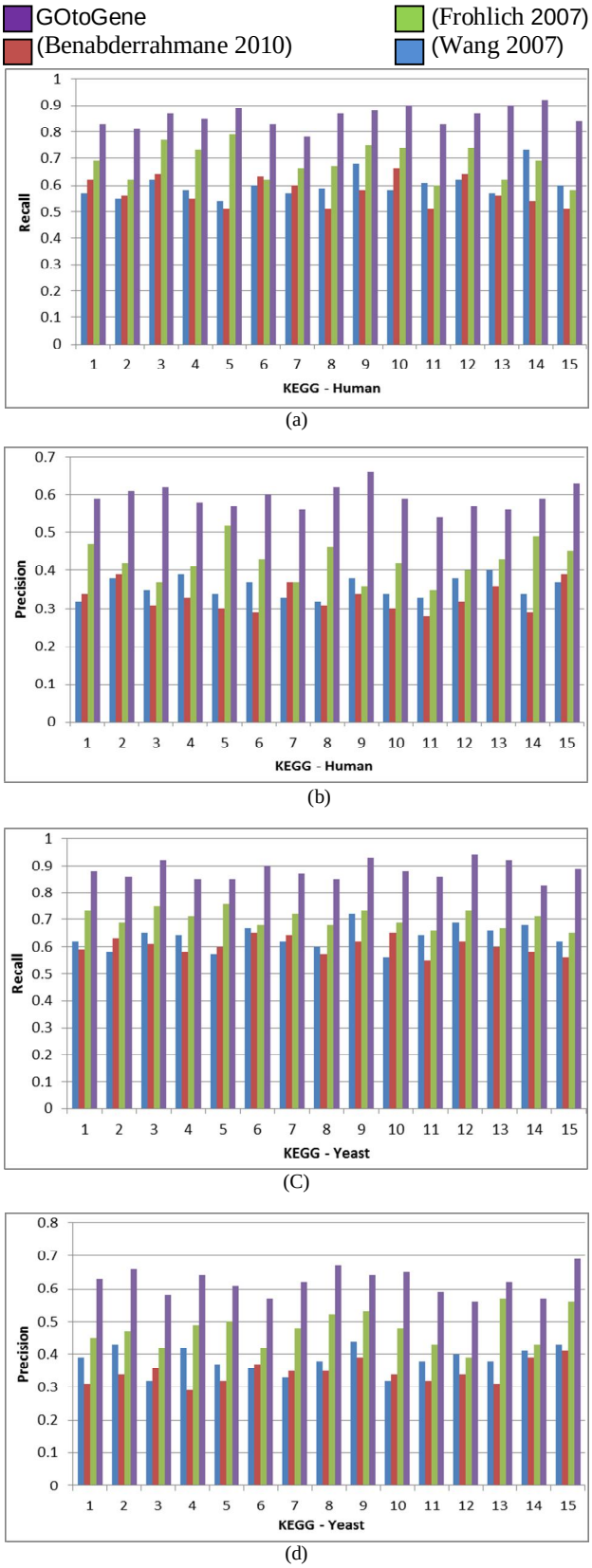

Fig. 10: Recall and precision using KEGG benchmark

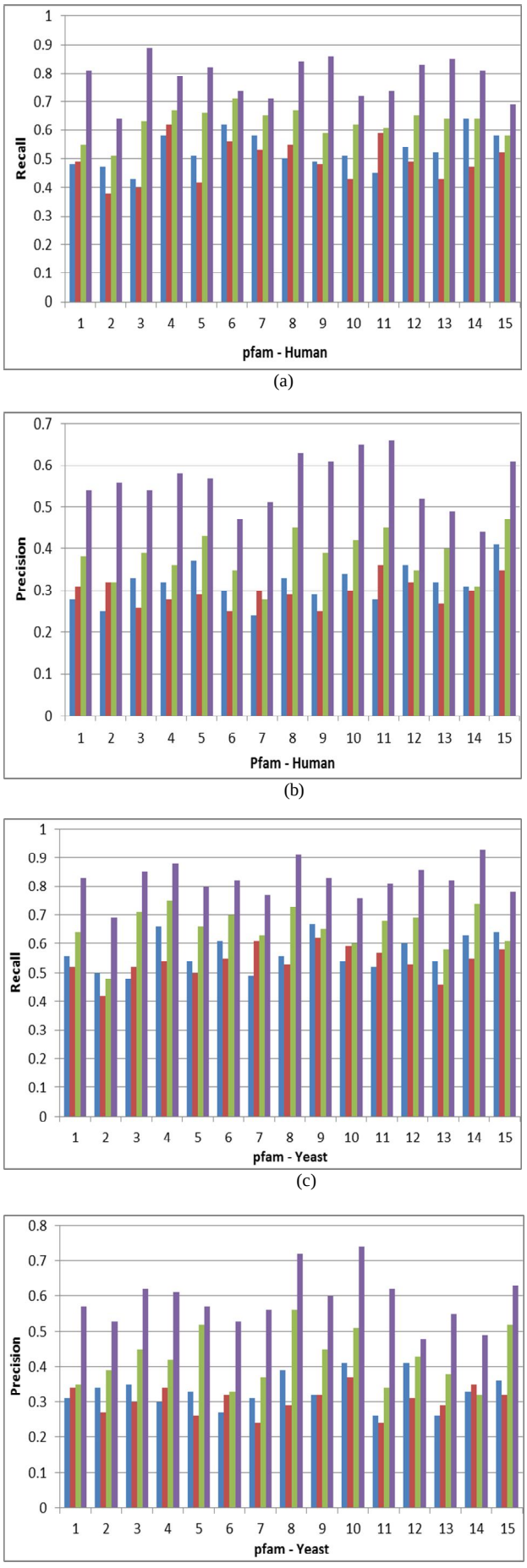

Fig. 11: Recall and precision using Pfam benchmark

8 Conclusion

In this paper, we proposed a system called GOtoGene that determines the semantic relationships among genes and gene products using the concept of existence dependency. Given a set of genes g_1, g_2, …g_n, GOtoGene identifies the Semantic Relevant Lowest Common Ancestor (SRLCA) of the terms annotating g_1, g_2, …g_n in the GO graph. The genes annotated by the SRLCA have the closest semantic relationships with g_1, g_2, …g_n. We experimentally evaluated the quality of GOtoGene and compared it with (Benabderrahmane et al. 2010, Frohlich 2007, Wang et al. 2007). Results showed that GOtoGene outperforms the other three methods.

9 References

Bada, M., Turi, D., McEntire, R. & Stevens, R. Using Reasoning to Guide Annotation with Gene Ontology Terms in GOAT. SIGMOD Record 33(2004).

Benabderrahmane, S., Smail-Tabbone, M., Poch, O.,Napoli, A., Devignes, M. IntelliGO: a new vector-based semantic similarity measure including annotation origin. BMC Bioinformatics 11:588, 2010.

Coute F, et al. DI/FCUL TR 03-29. Department of Informatics, University of Lisbon; 2003. Implementation of a Functional Semantic Similarity Measure between Gene-Products, 2003.

Elmasri, R., Navathe, S. "Fundamentals of Database Systems", Addison-Wesley Computing, six edition, 2011.

Frohlich, H. GOSim – an R-package for computation of information theoretic GO similarities between terms and gene products. BMC Bioinformatics 8 (1), 166, 2007.

Gene Ontology (2011). http://www.geneontology.org/GO.ontology.relations.shtml

Guo X, Liu R, Shriver CD, Hu H, Liebman MN: Assessing semantic similarity measures for the characterization of human regulatory pathways. *Bioinformatics* 2006, 22(8)**:**967-973

Lee SG, et al. A graph-theoretic modeling on GO space for biological interpretation of gene clusters. Bioinformatics 2004; 20:381-388.

Lin D. An information-theoretic definition of similarity, Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy; In Proc. 15th Int'l Conference on Machine Learning, 1998, pp.296-304.

Lord, P.W., Stevens, R.D., Brass, A. and Goble, C.A. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. (2003) Bioinformatics 19, 1275-83

Nagar A, Al-Mubaid H: A New Path Length Measure Based on GO for Gene Similarity with Evaluation using SGD Pathways. 21st IEEE International Symposium on Computer-Based Medical Systems (CBMS 08)

Pesquita C, Faria D, Bastos H, Ferreira A, Falcao AO, Couto F: Metrics for GO based protein semantic similarity: a systematic evaluation. BMC Bioinformatics 2008, 9(Suppl 5):S4.

Pesquita C, Faria D, Falcao AO, lord P, Couto F: Semantic Similarity in Biomedical Ontologies. PLoS Comut Biol 2009, 5(7):e1000443.

Resnik P. Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. J. Artificial Intelligence Res. 1999; 11:95-130.

Schlicker A, Domingues F, Rahnenfuhrer J, Lengauer T. A new measure for functional similarity of gene products based on Gene Ontology. BMC Bioinformatics 2006;7:302.

Sevilla JL, Segura V, Podhorski A, Mato JM, Corrales FJ, Rubio A: Correlation between Gene Expression and GO Semantic Similarity. IEEE/ACM Trans. Comput. Biol. Bioinformatics 2005, 2(4):330{338}.

Snoeck, M. Dedene, G. "Existence Dependency: The key to semantic integrity between structural and behavioral aspects of object types", IEEE Transactions on Software Engineering, Vol. 24, No. 24, pp.233- 251, 1998.

Wang, H. et al. (2004) Gene Expression Correlation and Gene Ontology-Based Similarity: An Assessment of Quantitative Relationships. IEEE Symposium on Computational Intelligence in Bioinformatics.

Wang J., Du Z, Payattakool R, Yu, P., Chen, C. A new method to measure the semantic similarity of go terms. *Bioinformatics* 2007; 23:1274-1281

Widjaya, N., Taniar, D., Rahayu, W. "Aggregation Transformation of XML Schema to Object-Relational Databases", Innovative Internet Community Systems, LNCS 2877, pp. 251-262, 2003.

Xu, Q., Shi, Y., Lu, Q., Zhang, G., Luo, Q., Li, Y. GORouter: an RDF model for providing semantic query and inference services for Gene Ontology and its associations. BMC Bioinformatics 2008.

XML Query Use Cases, *W3C Working Draft 2007*. Available: http://www.w3.org/TR/xquery-use-cases/

CRAN - Package GOSim, 2012. Available at: http://cran.r-project.org/web/packages/GOSim/index.html