BMC Bioinformatics

Methodology article

Open Access

# A relation based measure of semantic similarity for Gene Ontology annotations

Brendan Sheehan*, Aaron Quigley, Benoit Gaudin and Simon Dobson

Address: Systems Research Group, School of Computer Science and Informatics, University College Dublin, Belfield, Dublin 4, Ireland

Email: Brendan Sheehan* - brendan.sheehan@ucd.ie; Aaron Quigley - aaron.quigley@ucd.ie; Benoit Gaudin - benoit.gaudin@ucd.ie; Simon Dobson - simon.dobson@ucd.ie

* Corresponding author

Published: 4 November 2008

*BMC Bioinformatics* 2008, **9**:468 doi:10.1186/1471-2105-9-468

Received: 9 May 2008
Accepted: 4 November 2008

This article is available from: http://www.biomedcentral.com/1471-2105/9/468



## Abstract

**Background:** Various measures of semantic similarity of terms in bio-ontologies such as the Gene Ontology (GO) have been used to compare gene products. Such measures of similarity have been used to annotate uncharacterized gene products and group gene products into functional groups. There are various ways to measure semantic similarity, either using the topological structure of the ontology, the instances (gene products) associated with terms or a mixture of both. We focus on an instance level definition of semantic similarity while using the information contained in the ontology, both in the graphical structure of the ontology and the semantics of relations between terms, to provide constraints on our instance level description.

Semantic similarity of terms is extended to annotations by various approaches, either though aggregation operations such as min, max and average or through an extrapolative method. These approaches introduce assumptions about how semantic similarity of terms relates to the semantic similarity of annotations that do not necessarily reflect how terms relate to each other.

**Results:** We exploit the semantics of relations in the GO to construct an algorithm called SSA that provides the basis of a framework that naturally extends instance based methods of semantic similarity of terms, such as Resnik's measure, to describing annotations and not just terms. Our measure attempts to correctly interpret how terms combine via their relationships in the ontological hierarchy. SSA uses these relationships to identify the most specific common ancestors between terms. We outline the set of cases in which terms can combine and associate partial order constraints with each case that order the specificity of terms. These cases form the basis for the SSA algorithm. The set of associated constraints also provide a set of principles that any improvement on our method should seek to satisfy.

**Conclusion:** We derive a measure of semantic similarity between annotations that exploits all available information without introducing assumptions about the nature of the ontology or data. We preserve the principles underlying instance based methods of semantic similarity of terms at the annotation level. As a result our measure better describes the information contained in annotations associated with gene products and as a result is better suited to characterizing and classifying gene products through their annotations.

## Background

Although the semantic similarity between two GO terms has been extensively investigated [1-4], how to define similarity between two gene products based on GO annotations for a specific application remains unclear [5]. To date annotation similarity has been computed by four general approaches: the *set-based* approach; the *graph-based* approach; the *vector-based* approach; and the *term-based* approach. In the set-based approach an annotation is viewed as a 'bag of words'. Two annotations are similar if there is a large overlap between their sets of terms. A graph-based approach views similarity as a graph-matching procedure. Vector-based methods embed annotations in a vector space where each possible term in the ontology forms a dimension. Term-based approaches compute similarity between individual terms and then combine these similarities to produce a measure of annotation similarity.

All the above approaches do not consider the semantics of relationships between terms. How terms are related can significantly alter how an annotation, which is a set of terms, is interpreted. In the GO there are two main types of relations: is_a and part_of. The is_a relation represents a taxonomic relationship between terms that can be modeled using the improper subset relation, which is a partial ordering of terms. The part_of relation represents a partonomic relationship between terms that can also be modeled in terms of a partial order. Though the partial orders represented by taxonomies and partonomies are well understood there has been little attention given as to how these two partial orderings combine. Using the various cases identified by combining taxonomies and partonomies we construct an algorithm called SSA (*S*emantic *S*imilarity of *A*nnotations) that identifies the terms that can be associated with an annotation and terms that relate to both annotations. Instances associated with these terms are then used to construct a Resnik-like measure of annotation similarity thus extending the underlying intuitions behind this term-based measure to the annotation level.

A measure of term or annotation similarity should be based on a set of principles that form the basis for what is considered similar. The nature of similarity has been the focus of intense research in the areas of *aesthetics* [6,7] and *psychology* [8]. In mathematics properties such as *identity*, *symmetry* and the *triangle inequality* have been used to form the basis of measures of similarity of mathematical objects. Principles of term and annotation similarity have been suggested by various authors. This work intends to build on these principles and introduce additional principles that a measure of similarity should seek to satisfy.

Similarity between objects is normally expressed as a number that ranges along an interval on the real numbers $\mathbb{R}$. However the main purpose of similarity is usually to determine whether two or more objects are similar to a reference object. For this reason a measure of similarity can be viewed as a partial order on a set of objects, the actual numbers play only a secondary purpose. For example, we may say that an object $X$ is more similar to $Z$ than another object $Y$. Formally this is expressed as $sim(X, Z) > sim(Y, Z)$.

In the study of ontological similarity Lin [9] develops the principles of *commonality* and *difference* when constructing a measure of *term similarity*. The greater the commonality between objects the greater the similarity. Likewise, the greater the difference between objects the greater the dissimilarity. The source of both the commonality and difference between terms depends on the method chosen to measure the *descriptiveness* of terms. Different sources of descriptiveness may result in different orderings of similarity between terms or annotations.

Popescu *et al*. [10] recognize that an important property of *term similarity* is that two different terms should have a non-zero similarity value if the terms are related. They also recognize that an important property of *annotation similarity* is that the descriptiveness of annotations should be greater than or equal to the descriptiveness of its constituent terms. In this paper this property is called the *monotonicity property*.

In defining a measure of similarity a set of relevant properties that objects can be compared along are identified. In ontological similarity, whether of terms or annotations, there are two main sources of similarity: the *conceptual* or *structural* level; and the *instance* level. At the structural level we may consider such properties as graph distance, graph similarity, relation types, common ancestors, etc. At the instance level we consider the set of instances associated with a term or annotation. Our measure of ontological similarity combines aspects from both levels. Here we survey how various measures of annotation similarity combine these properties in various ways to form the basis for a measure of descriptiveness of a term or annotation.

### *Set-Based Approaches*

Set based methods for measuring the similarity of annotations are based on the Tversky ratio model of similarity [8,11] which is a general model of distance between sets of terms. It is represented by the formula

$$\frac{f(G_1 \cap G_2)}{f(G_1 \cap G_2) + \alpha * f(G_1 - G_2) + \beta * f(G_2 - G_1)}$$

where $G_1$ and $G_1$ are sets of terms or annotations from the same ontology and $f$ is an additive function on sets (usu-

ally set cardinality). For $\alpha = \beta = 1$ we get the Jaccard distance between sets:

$$S_{Jaccard} = \frac{f(G_1 \cap G_2)}{f(G_1 \cup G_2)}$$

and for $\alpha = \beta = \frac{1}{2}$ we get the Dice distance between sets [11]:

$$S_{Dice} = \frac{2 * f(G_1 \cap G_2)}{f(G_1) + f(G_2)}$$

In this situation the source of descriptiveness of an annotation is its set of terms. Each term and its set of associated instances is considered independent of other terms. The commonality and difference between annotations is modeled as set intersection and difference of sets of terms respectively. Set-based approaches return a similarity of zero if they do not share common terms ignoring the fact that terms may be closely related. Because of the atomic nature of terms in the set-based approach the monotonicity property does not apply.

#### *Vector-Based Approaches*

Vector-based methods embed ontological terms in a vector space by associating each term with a dimension. Usually a vector is binary consisting of 0's and 1's where 0 denotes the absence (resp. presence) of a term (along a particular dimension) in an annotation. This has the advantage that standard clustering techniques on vector spaces such as k-means can be applied to group similar terms. What is required is a means of measuring the size of vectors. This can be achieved by embedding terms in a metric space (usually Euclidean). The most common method of measuring similarity between vectors of terms is the cosine similarity

$$s_v(G_1, G_2) = \frac{v_1 \bullet v_2}{|v_1||v_2|}$$

where $v_i$ represents a vector of terms constructed from an annotation (group of terms) $G_i$. $|\cdot|$ corresponds to the size of the vector and $\bullet$ corresponds to the dot product between two vectors. The source of descriptiveness, commonality and difference is the same as the situation for set-based approaches.

#### *Graph-Based Approaches*

An ontology is a directed, acyclic graph (DAG) whose edges correspond to relationships between terms. Thus it is natural to compare terms using methods for graph matching and graph similarity. We may consider the similarity between annotations in terms of the sub-graph that connects terms within each annotation. Annotation similarity is then measured in terms of similarity between two graphs. Graph matching has only a weak correlation with similarity between terms. It is also computationally expensive to compute, graph matching being an NP-complete problem on general graphs [12].

The descriptiveness of an annotation is modeled by the set of nodes and edges associated with a subgraph. Commonality between annotations is based on the set intersection while difference is modeled by the set difference where each set consists of the nodes and edges associated with each subgraph. Alternatively, the set of edges may be ignored and only common terms of both graphs are considered [13-15].

#### *Improving Similarity Measures by Weighting Terms*

Set, vector and graph-based methods for measuring similarity between annotations can be improved by introducing a weighting function into the similarity measure. For example, the weighted Jaccard distance can be formulated as:

$$S_{WeightedJaccard}(G_1, G_2) = \frac{\sum_{\{T_i \in G_1 \cap G_2\}} m(T_i)}{\sum_{\{T_j \in G_1 \cup G_2\}} m(T_j)}$$

where, as before, $G_1$ and $G_2$ are annotations or sets of terms describing data (e.g. a gene product), $T_x$ is the $x^{th}$ term from a set of terms and $m(T_x)$ denotes the weight of $T_x$. This weighting function can be used to represent various properties of a term or annotation such as a measure of vagueness, uncertainty, sense of preference or a combination of the above. The vector-based approach may be extended so that values along a particular dimension can lie on the interval [0, 1] or [0, ∞). The graph-based approach can be extended by weighting the edges between terms in the graph.

Assigning a weight to each term in an annotation allows for the possibility of introducing the monotonicity property into a similarity measure. Using the monotonicity property, the weight associated with an annotation should be greater than or equal to the weight associated with any of its constituent terms. Weights can form an additional basis on which to measure the descriptiveness of a term or annotation.

##### *Instance-Based Weights*

One approach to assigning weight to an ontological term is to measure how *informative* a term is in describing data. A method of measuring information is to analyze a term's use in a corpus against the general use of ontological terms in the same corpus. Information is measured using the *surprisal* function:

$$IC_{Corpus}(T_i) = -\log(p(T_i)) \quad (1)$$

where $p(T_i)$ corresponds to the probability of a term $T_i$ or its taxonomic descendants occurring in a corpus. For example, consider the case where there are 30 distinct instances in a corpus and 5, 3 and 2 of these instances are annotated by the terms $T_i$, $T_j$ and $T_k$ respectively. If $T_j$ and $T_k$ are sub-types or children of $T_i$ and do not have child terms themselves then

$IC_{Corpus}(T_i) = -\log\left(\frac{5+3+2}{30}\right) \approx 1.099$ .

*Other Weighting Approaches*

Other measures of information can be used not necessarily relying on corpus data. One measure [16] relies on the assumption that how the ontology is constructed is semantically meaningful:

$$IC_{Ont}(T_i) = 1 - \frac{\log(desc(T_i)+1)}{\log(numTerms)}$$

where $desc(T_i)$ returns the number of descendants of term $T_i$ and *numTerms* refers to the total number of terms in the ontology.

### *Term-Based Approaches*

In term-based approaches similarity between pairs of terms from each annotation are computed. These weightings are then combined in order to characterize the similarity between annotations as a whole. There are several ways to combine similarities of pairs of terms such as the min, max or average operations. Term-based approaches depend on a function $s(T_i, T_j)$ where $T_i$ and $T_j$ are terms from two annotations $G_1$ and $G_2$ respectively. $s(T_i, T_j)$ provides a measure of distance/similarity between these two terms. Once distances has been measured between all possible pairs of terms they are then aggregated using an operation such as max or the average of all distances. For example:

$$S_{avg}(G_1, G_2) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} s(T_i, T_j)}{m*n}$$

More sophisticated term based approaches combine multiple measures of term similarity and aggregate similarity values using more complex functions, for example [17].

*Graphical Measures of Term Similarity*

The simplest approach to measuring similarity between ontological terms using the graph structure is to measure the shortest path distance between terms in the graph [18,19]. Referring to figure 1, in terms of graph distance, we may consider the terms 'muscle cell proliferation' and 'fibroblast cell proliferation' (graph distance of 2) as being more similar than the former term with 'fibroblast regulation' (graph distance of 3). However the graph distance has only a weak correlation with similarity of terms. The semantic similarity between 'positive fibroblast regulation' and 'negative fibroblast regulation' is far greater than the similarity between 'muscle cell proliferation' and 'fibroblast cell proliferation' even though both examples have a graph distance of two. A simple graph distance-based measure of similarity does not model in a consistent way any notion of commonality or difference between terms.

A more refined use of graph distance as a basis for a measure of term similarity is found in the *Wu-Palmer* measure of similarity [20]. It uses the idea that the distance from the root to the *lowest common taxonomic ancestor* (LCTA) measures the commonality between two terms while the sum of the distance between the LCTA and each term measures the difference between two terms. Combining these aspects results in the formula:

$$s_{Wu-Palmer}(T_1, T_2) = \frac{2*dist(T_{lcta}, T_{root})}{dist(T_1, T_{lcta})+dist(T_2, T_{lcta})+2*dist(T_{lcta}, T_{root})}$$

Where $T_1$ and $T_2$ are the two terms being compared, $T_{lcta}$ is the term that corresponds to the lowest common taxonomic ancestor between $T_1$ and $T_2$. $T_{root}$ denotes to root node of the ontology (assuming that the ontology has only one root). $dist(T_i, T_j)$ denotes the graph distance between terms $T_i$ and $T_j$. The $2 * dist(T_{lcta}, T_{root})$ component of the denominator serves to normalize the measure.

*Instance-Based Measures of Term Similarity*

Similarity may be measured using an instance based measure of semantic similarity as computed by either Resnik (eqn. 2) or Lin (eqn. 3). Resnik [21,22] exploits the informativeness of the lowest common ancestor between terms as a measure of semantic similarity:

$$s_{Resnik}(T_i, T_j) = IC_{Corpus}(T_{lcta}) \quad (2)$$

where $T_{lcta}$ denotes the lowest common taxonomic ancestor between ontological terms $T_i$ and $T_j$. This measure only accounts for the commonality between terms.

Another method of measuring similarity derived by Lin [9] is:

$$s_{Lin}(T_i, T_j) = \frac{2*IC_{Corpus}(T_{lcta})}{IC_{Corpus}(T_i)+IC_{Corpus}(T_j)} \quad (3)$$

which has the advantage that it maps onto values on the interval [0, 1] unlike Resnik's measure which maps onto the interval [0, ∞). Lin's measure also accounts for both the commonality and difference between terms. Resnik's

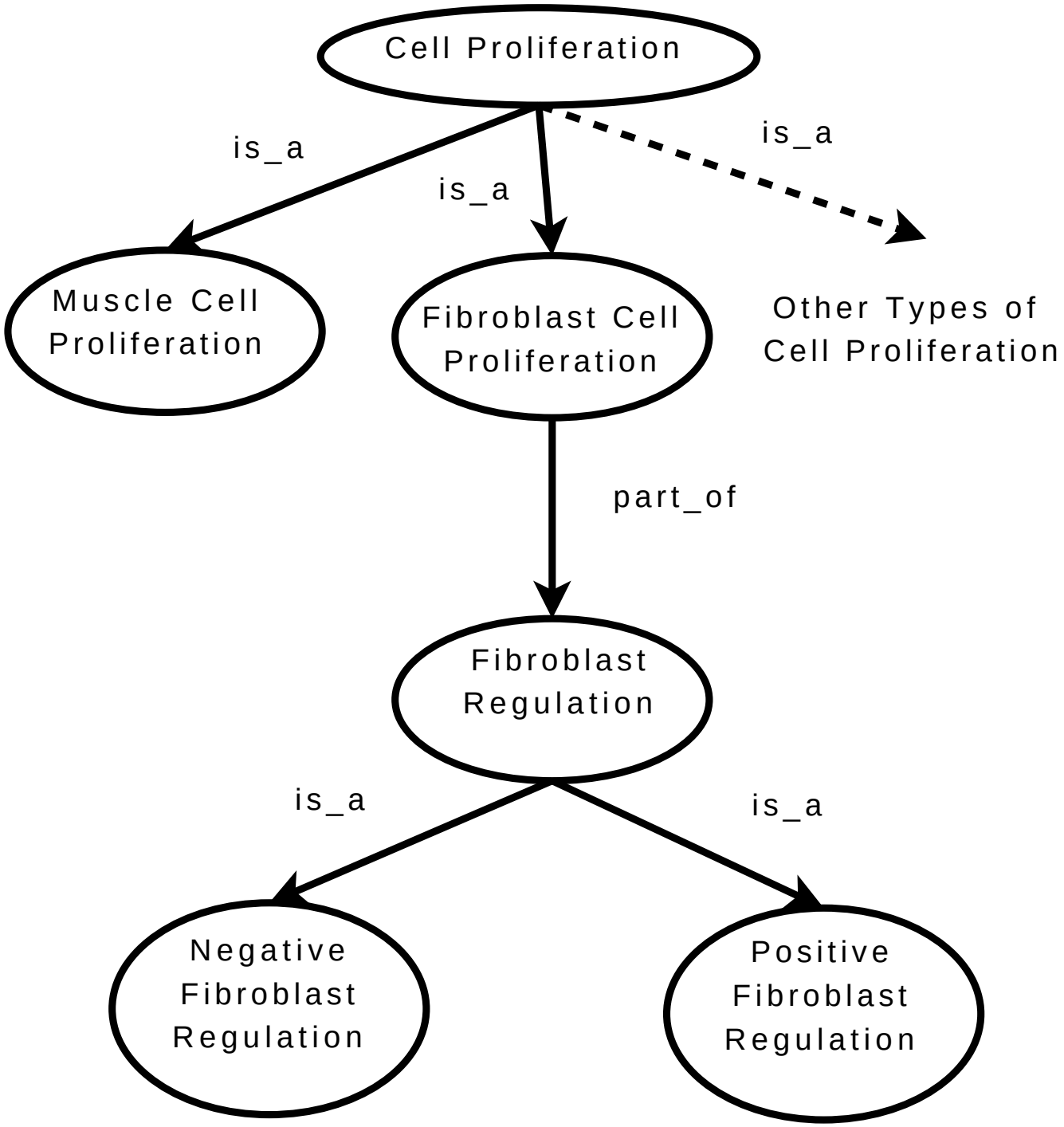



**Figure 1**
**An Example of an Ontology of GO Terms**. Nodes in the graph correspond to ontological terms. Edges correspond to relations between terms. Lower down terms in the diagram are descendants of terms higher up in the diagram if connected by an edge.

measure does have the desirable property that terms close to the root of the ontology have a low similarity however. This is not the case for Lin's measure.

The only structural property that both Resnik and Lin exploit is the lowest common taxonomic ancestor. To overcome this weakness Jiang and Conrath [23] integrate graph distance based measures of similarity into information based approaches. They construct a generalized weighting measure between a child and its immediate parent that accounts for the number of out edges and depth of terms along the shortest path between the compared terms in the ontology. While they acknowledge that other relation types might be relevant to measuring similarity their measure is based solely on the taxonomic or is_a relations in the ontology.

### *New Approaches to Annotation Similarity*

Beyond the set, vector, graph and term-based approaches to measuring similarity of annotations exist other methods that introduce the additional properties discussed above such as *monotonicity* and taking into account the semantics of ontological relations.

#### *Similarity Based on Fuzzy Measures*

The monotonicity property leads naturally to the use of fuzzy measures as a basis for measuring the descriptiveness of an annotation. Using the information content measure of terms described in eqn. 1 as the basis for measuring similarity a fuzzy measure is constructed. A fuzzy measure is a weighting on sets of terms such that the weight associated with a set of terms is greater than or equal to the weight associated with any of its subsets.

Popescu *et al*. [10] use fuzzy measures to induce a weighting $m$ for an annotation from its constituent terms. This weight is extrapolated from the weights of individual terms by using the formula for constructing a Sugeno $\lambda$-fuzzy measure: For a set of terms $G_a$, $G_b$ and $G_c$ where $G_c = G_a \cup G_b$ and $G_a \cap G_b = \emptyset$ a $\lambda$-fuzzy measure for $G_c$ is

$$m_\lambda(G_c) = m_\lambda(G_a) + m_\lambda(G_b) + \lambda * m_\lambda(G_a) * m_\lambda(G_b)$$

where $\lambda$ is a value that ensures that $m(G_c) \geq m(G_a)$ and $m(G_c) \geq m(G_b)$. Given that the weights (fuzzy measure densities) $m$ for individual terms $T_i$ in an annotation are known then $\lambda$ can be determined by solving the following equation:

$$1 + \lambda = \prod_{T_i} (1 + \lambda m(T_i))$$

In [10] the weight for each term is based on the ICCorpus measure (eqn. 1). The similarity of two annotations, represented by a set of terms $G_1$ and $G_1$ from the same ontology, are compared using the similarity function:

$$S_{FMS}(G_1, G_2) = \frac{m_{G_1}(G_1 \cap G_2) + m_{G_2}(G_1 \cap G_2)}{2}$$

where $m_{G_1}$ and $m_{G_2}$ are the $\lambda$-fuzzy measure functions that characterize $G_1$ and $G_2$ respectively. The relatedness of terms is accounted for by augmenting each annotation with the lowest common ancestors for each pair of terms from each annotation. This ensures a non-zero similarity between annotations containing related terms.

However, an ontology models other aspects of relatedness that should be taken into account. Relations between terms in an annotation can be used to identify redundant terms whose relevance to the descriptiveness of an annotation is already accounted for by other terms. For example, if two terms in an annotation are taxonomically related the existence of the parent term is implied by the existence of the child term.

If redundancy of terms is not taken into account it may lead to too many or too few instances being associated with the term. This is especially true when a term is part_of another term. The instances associated with the annotation consist of the parts and not what the instances are part of.

#### *Exploiting Semantics of Ontological Relations*

Wang *et al*. [14] account for the different contributions that terms related by is_a and part_of relations make to the descriptiveness of a term. The semantic contribution that ancestor terms make to a child term is calculated by:

$$SV(T_i) = \sum_{T_j \in T_{anc,i}} s_{T_i}(T_j)$$

where $T_{anc,i}$ denotes the ancestors of term $T_i$ and $s_{T_i}$ is calculated as

$$\begin{cases} s_{T_i}(T_i) = 1 \\ s_{T_i}(T_j) = \max\{w_e * s_{T_i}(T_k) \\ \mid T_k \in \text{childrenOf}(T_j)\} \text{ if } T_j \neq T_i \end{cases}$$

where $w_e \in [0, 1]$ is a number that corresponds to the semantic contribution factor for edge $e$. $childrenOf(T_x)$ is a function that returns the immediate children of $T_x$ that are ancestor terms of $T_i$. In this paper $w_{\text{is\_a}} = 0.8$ and $w_{\text{part\_of}} = 0.6$. The similarity of two terms is computed by the formula

$$s(T_i, T_j) = \frac{\sum_{T_k \in T_{anc,i} \cap T_{anc,j}} (s_{T_i}(T_k) + s_{T_j}(T_k))}{SV(T_i) + SV(T_j)}$$

A term-based approach is taken to measuring the similarity between annotations $G_1$ and $G_2$. The similarities of the most similar pairs of terms from each annotation are averaged over to calculate the similarity between annotations:

$$S_{Wang}(G_1, G_2) = \frac{\sum_{T_i \in G_1} s(T_i, G_2) + \sum_{T_j \in G_2} s(T_j, G_1)}{|G_1| + |G_2|}$$

where $s(T_x, G_y) = \max_{T_y \in G_y}(s(T_x, T_y))$ and $|G_y|$ denotes the number of terms in annotation $G_y$.

Wang *et al*. make the observation that the instance based measures of term similarity will produce varying results based on the corpus chosen. They keep a fixed value for the contribution each relation type makes to the descriptiveness of a term. This does not account for the varying influence of terms on each other throughout the ontology even if the graph distance is the same. Exploiting the corpus statistics, if used appropriately, may account for this drawback. As with all term-based methods, where terms from each annotation are compared in a pairwise fashion, it is difficult to see how the monotonicity property is ensured when measuring the similarities between two annotations.

## Methods

The Gene Ontology relates terms using is_a and part_of relations. We develop a measure of informativeness that provides a description of an annotation that takes into consideration the relations between terms. We use the informativeness measure of a term (eqn. 1) as the basis for providing a description of an annotation. We define an algorithm called *SSA* that combines the instances of terms while taking into account how these sets of instances are related by how their associated terms are related in the ontology. This results in a set of instances that can be said to be associated with an annotation and not just a term. We can then extend the concept of instance based semantic similarity of terms, such as Resnik's measure, to annotations.

### *Interpreting Annotations from Taxonomies*

A taxonomy induces a partial ordering on a set of terms by the improper subset relation $\subseteq$. If $T_i$ is_a $T_k$ and $T_j$ is_a $T_k$ then the set of instances associated with both $T_i$ and $T_j$ are subsets of $T_k$. Assuming that we know of all possible instances that can be associated with a term, whatever properties that instances of both $T_i$ and $T_j$ share can be associated with any of the instances that can be associated with $T_k$. This forms the basis for measuring the commonality between terms used in instance-based measures of similarity between terms.

The difference between terms $T_i$ and $T_j$ is modeled by the difference between the set of instances associated with each term. If we have two or more terms from a taxonomy in an annotation then it is reasonable to argue that the set of instances associated with an annotation should be the intersection of the set of instances associated with each term. The informativeness of the annotation is then based on the set of instances resulting from this intersection.

### *Interpreting Annotations from Partonomies*

The part_of relation between terms denotes the concept that one term is '*part of* ' another. It provides an alternative notion of relatedness between terms. An ontology consisting only of part_of relations is known as a *partonomy*. An example of a simple partonomy is *wheel* part_of *car*. It would not make sense to say that a *wheel* is_a *car*. The study of partness is complicated by the fact that there are many kinds of part_of relations. Yet the study of partness, known as *mereology* [24], has shown that there are also common aspects to all types of part_of relations, namely that part_of relations form a partial ordering on the sets of instances associated with each term.

According to the GO Consortium's usage guidelines since 2004 [25] the part_of relation should be interpreted as 'necessarily part of' where $T_i$ part_of $T_j$ means that all instances of $T_i$ are part of one or more instances of $T_j$. The converse is not necessarily true. For example, all nuclei are part of cells but not all cells contain a nucleus. Bittner [26] models such a part_of relation using an improper partial order i.e. for term $T_i$ with descendant terms $T_j$.

$$T_j \leq_{part\_of} T_i \; \forall T_j \text{ part\_of } T_i \tag{4}$$

Annotations consisting of terms such that one term is part_of another should view the child term as being relevant to the annotation while the parent term provides redundant, contextual information. For example, consider an annotation consisting of two terms $T_i$ and $T_j$ from a partonomy. If $T_j$ part_of $T_i$ then the annotation should be interpreted as *the set of instances of* $T_j$. All we can say is that the number of instances of $T_i$ associated with the annotation can be no more than the number of instances of $T_j$. In general, an annotation consisting of terms belonging to a partonomy consists of terms that provide the set of instances that can be associated with the annotation while other terms provide the context in which these instances are embedded.

***Partial Order Constraints for GO Annotations***

Figure 2 shows a subset of the GO consisting of both part_of and is_a relations. According to the taxonomic is_a relations both 'mitochondrial chromosome' and 'mitochondrial nucleoid ' are 'mitochondrial part's. A measure of descriptiveness of a term should at least say that both 'mitochondrial chromosome' (a) and 'mitochondrial nucleoid ' (b) are more descriptive than 'mitochondrial part' (c), i.e. $a, b \subseteq c$. Likewise, the part_of relation in figure 2 indicates that $a \leq_{part\_of} b$. Here we can see how the part_of relation provides additional indirect information about descriptiveness not represented by the taxonomic relations. If an annotation consists of the terms 'mitochondrial chromosome' and 'mitochondrial nucleoid' then the annotation should be interpreted as *the set of instances of 'mitochondrial chromosome'*. If the terms 'mitochondrial part' and 'chromosome' are added to the annotation then the same set of instances should be associated with the annotation. All additional terms are already implied by the existence of 'mitochondrial chromosome' in the annotation. If we had either treated the part_of relation as an is_a relation or ignored it then the

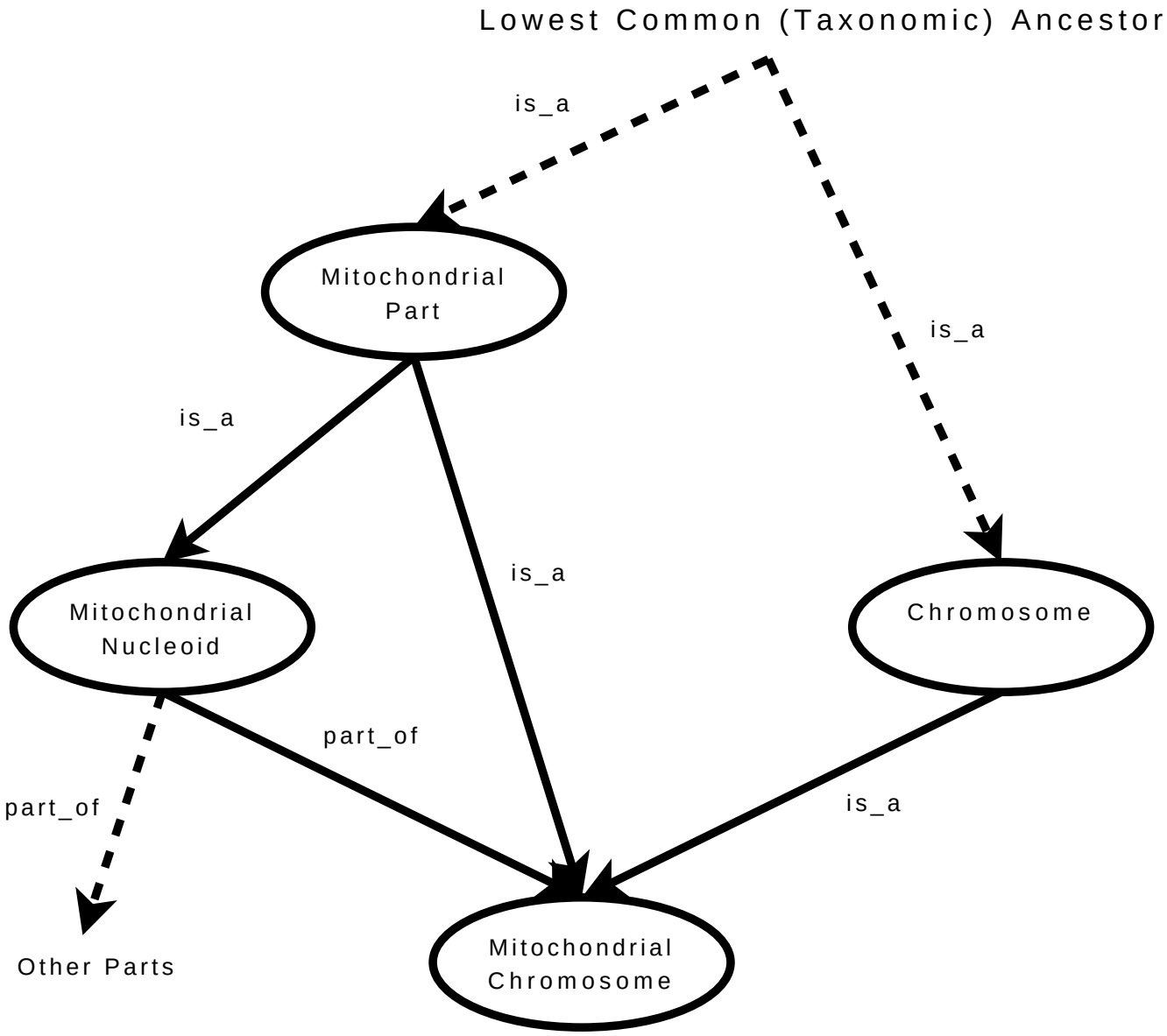



**Figure 2**

**A Subset of GO Terms and Relations**. An example of where the part_of relation plays an important role in interpreting annotations. If an annotation contains the term 'mitochondrial chromosome' then all other terms shown in the graph are redundant. The diagram also shows various cases that describe how terms relate to each other.

annotation would have been interpreted as *the set of instances that are both 'mitochondrial chromosome' and 'mitochondrial nucleoid'*. With this interpretation we would have possibly returned an empty set of instances since chromosomes are not nucleoids.

The GO consists of many examples similar to the one described above. In general, the GO can be viewed as a taxonomy interspersed with part_of relations. Two terms are said to be *directly related* if there exists a series of relations on a single path between them. Terms that are not directly related along a path in the graph are *indirectly related* via a common ancestor. For example there may be other terms that are part_of 'mitochondrial nucleoid' in which case the term 'mitochondrial chromosome' is only related to the other parts by an indirect path of part_of relations. Though not shown, the terms 'mitochondrial nucleoid' and 'chromosome' are only indirectly related via a common ancestor through a number of is_a relations. When interpreting an annotation it is necessary to account for such situations.

In general, as described in table 1, there are nine cases to handle when trying to account for how terms are related. Terms or their taxonomic descendants may be directly related to each other in the ontology via a single path. Alternatively they may be indirectly related to each other via a common ancestor in which case we consider the two paths from the common ancestor to each term. A path may be homogeneous in that it consists of relations of only one type i.e. all relations are either only is_a or only part_of. Such paths are denoted by IS and PART respectively. A path that is inhomogeneous, consisting of both is_a and part_of relations, is denoted by MIXED.

*Directly Related Cases*

There are three cases to handle when there exists a single path between terms in the ontology: IS, PART and MIXED paths. The first case is the generalized case of taxonomic relations where $T_i$ IS $T_j$. For two terms $T_i$ and $T_j$, where $T_j$ is the parent term and $T_i$ is a descendant, and a set of $n$ intermediate terms $\{T^n\}$ such that:

$$T_i \subseteq T_1^n \subseteq T_2^n \ldots \subseteq T_{n-1}^n \subseteq T_n^n \subseteq T_j$$

it can be inferred that $T_i \subseteq T_j$. Where terms are related by a PART path a similar argument can be inferred for how two terms are ordered.

For the MIXED case there exists a mixture of is_a and part_of relations. The nature of the MIXED relationship is ultimately determined by the part_of relations. For example, if $T_i$ MIXED $T_j$ then this can be interpreted as $T_i$ part_of $T_j$. There may be several is_a relations traversed along a MIXED path from $T_j$ to $T_i$ before a part_of relation is encountered. This means that $T_i$ can only be part_of a subset of the instances of $T_j$. This subset is identified by the set of instances associated with the term (labeled $T_k$ in table 1) which is the parent term of the first part_of relation encountered along a MIXED path from $T_j$ to $T_i$. This results in the partial order:

$$T_i \leq T_k \leq T_j$$

where $T_i$ is the descendant of $T_j$, $T_i$ is the parent and $T_k$ denotes the first term before a part_of relation is encountered while traversing the MIXED path in the ontology from $T_j$ to $T_i$. This form of reasoning can be further extended along the rest of the MIXED path to produce a more detailed partial order. However if the ultimate goal is to only determine the partial order between $T_i$ and $T_j$ then such induction of this reasoning is unnecessary.

*Indirectly Related Homogeneous Cases*

There are three cases to handle where both the paths to the common ancestor between terms are homogeneous: IS –

**Table 1: Partial Order Constraints**

| | Situation | Ordering |
|---|---|---|
| Directly* | $T_i$ IS $T_j$ | $\rho(T_i) \leq \rho(T_j)$ |
| | $T_i$ PART $T_j$ | $\rho(T_i) \leq \rho(T_j)$ |
| | $T_i$ MIXED $T_j$ via $T_k$ | $\rho(T_i) \leq \rho(T_k) \leq \rho(T_j)$ |
| Indirectly Via $T_{lca}$* | $T_i$ IS $T_{lca}$, $T_j$ IS $T_{lca}$ | $\rho(T_i), \rho(T_j) \leq \rho(T_{lca})$ |
| | $T_i$ PART $T_{lca}$, $T_j$ IS $T_{lca}$ | $\rho(T_i) \leq \rho(T_j) \leq \rho(T_{lca})$ |
| | $T_i$ PART $T_{lca}$, $T_j$ PART $T_{lca}$ | $\rho(T_i), \rho(T_j) \leq \rho(T_{lca})$ |
| | $T_i$ MIXED $T_{lca}$ via $T_k$, $T_j$ IS $T_{lca}$ | $(\rho(T_i) \leq \rho(T_k)), \rho(T_j) \leq \rho(T_{lca})$ |
| | $T_i$ MIXED $T_{lca}$ via $T_k$, $T_j$ PART $T_{lca}$ | $\rho(T_i), \rho(T_j) \leq \rho(T_k) \leq \rho(T_{lca})$ |
| | $T_i$ MIXED $T_{lca}$ via $T_k$, $T_j$ MIXED $T_{lca}$ via $T_m$ | $(\rho(T_i) \leq \rho(T_k)), (\rho(T_j) \leq \rho(T_m)) \leq \rho(T_{lca})$ |

Overview of general forms of relation based ordering for directly and indirectly related terms. Terms are indirectly related via a common ancestor term $T_{lca}$. Instances of terms $T_i$ and $T_j$ may be part of the common ancestor $T_{lca}$ via terms $T_k$ and $T_m$ respectively. $\rho$ denotes a function that measures the number of instances (our source of descriptiveness) of terms. These orderings assume complete knowledge of all instances associated with a term.

IS, PART – PART and IS – PART (or PART – IS). In the first case, where $T_i$ IS $T_{lca}$ and $T_j$ IS $T_{lca}$, since both terms $T_i$ and $T_j$ are taxonomic descendants of a lowest common ancestor $T_{lca}$ then it should be expected that the number of instances associated with $T_i$ and $T_j$ are less than the number of instances associated with $T_{lca}$. This results in the partial order

$$T_i, T_j \leq T_{lca}$$

An annotation consisting of two such related terms can be interpreted as *the set of instances that are associated with both $T_i$ and $T_j$*. A similar form of reasoning can be applied to the PART – PART case. The partial order for the final case IS – PART (or PART – IS) can be derived in a similar fashion to the inhomogeneous direct MIXED case. If $T_i$ IS $T_{lca}$ and $T_j$ PART $T_{lca}$ then it can be inferred that $T_j$ PART $T_i$. If an annotation consists of two such terms then it should be interpreted as *the set of instances of $T_j$*. As a partial order constraint this can be modeled as

$$T_j \leq T_i \leq T_{lca}$$

*Indirectly Related Inhomogeneous Cases*

Indirectly related inhomogeneous cases occur when terms are related by a common ancestor in the ontology and one or both of the paths connecting the common ancestor with each term consists of an inhomogeneous set of relation types. There are three such cases to account for: IS – MIXED (or MIXED – IS), PART – MIXED (or MIXED – PART) and MIXED – MIXED.

The partial order for the first case IS – MIXED (or MIXED – IS) can be handled by considering each path separately. The partial order for the $T_i$ IS $T_{lca}$ path is $T_i \leq T_{lca}$. The partial order for the MIXED path is $T_j \leq T_k \leq T_{lca}$ which is derived in the same way as the directly related MIXED case. Combining the two partial orders results in

$$(T_j \leq T_k),\ T_i \leq T_{lca}$$

If an annotation consists of two such terms then it should be interpreted as *the set of instances of $T_j$ that are part of instances that are of type $T_i$ and $T_k$*.

The PART – MIXED (or MIXED – PART) case requires slightly more reasoning about to construct its associated partial order. If $T_i$ PART $T_{lca}$ and $T_j$ MIXED $T_{lca}$ then it can be inferred that both $T_i$ and $T_j$ are part of $T_{lca}$. Because $T_j$ is only part of a subset of the instances associated with $T_{lca}$, the instances associated with $T_k$, then $T_i$ can only be part of the set of instances associated with $T_k$ also. This results in the partial order

$$T_j, T_i \leq T_k \leq T_{lca}$$

An annotation consisting of two such related terms should be interpreted as *the set of instances of $T_i$ and $T_j$ that are part of the same instances of $T_k$*.

The final case MIXED – MIXED occurs when paths from both terms to the common ancestor consist of a mixture of relation types. The partial order for such a case can be constructed by looking at each path separately. If $T_i$ MIXED $T_{lca}$ then the partial ordering is $T_i \leq T_k \leq T_{lca}$. Similarly for $T_j$ MIXED $T_{lca}$ we get $T_j \leq T_m \leq T_{lca}$. Combining the two partial orders results in

$$(T_i \leq T_k),\ (T_j \leq T_m) \leq T_{lca}$$

If an annotation consists of two such terms then it should be interpreted as *the set of instances of $T_i$and $T_j$that are part of the same instances of $T_k$and $T_m$*.

### *The SSA Algorithm*

The SSA algorithm is based on the nine cases of term relatedness described above. The SSA algorithm derives the set of instances that can be associated with an annotation from the set of instances associated with that annotation's constituent terms. There are two aspects to the algorithm: identifying which terms are the contextual, redundant instances and which terms' instances can be associated with the annotation. For example, a contextual instance may be 'mitochondrial nucleoid' that provides the context for the set of instances of 'chromosome'. Throughout we denote the set of contextual terms by *exclTerms* and the set of terms whose instances can be associated with the annotation as *inclTerms*. *numInst($T_i$)* denotes the number of instances associated with $T_i$.

The above partial order constraints were constructed under the ideal assumptions assumed by the partial orderings in taxonomies and partonomies. In reality there only ever exists an incomplete set of instances associated with terms and some adjustment of the number of instances is required if the partial order constraints are to be satisfied. Terms that are taxonomically related are guaranteed to satisfy the taxonomic constraints. However, terms that are partonomically related may not satisfy their associated partial order constraints. In these cases some adjustment of the number of instances associated with a term is necessary. For example, if $T_i$ PART $T_j$ and there are no instances associated with $T_j$ in the corpus while there are a number of instances associated with $T_i$ then in order to satisfy the PART constraint the number of instances of $T_j$ is set equal to the number of instances associated with $T_i$.

The algorithm consists of the following steps:

• For each distinct ordered pair $(T_i, T_j)$ of terms in annotations $G_1$ and $G_2$ respectively

- Identify the case that corresponds to how $T_i$ is related to $T_j$

  * Terms are assigned to *inclTerms* or *exclTerms* depending on case

  * The number of instances associated with a term may be adjusted if the case allows

• Remove any terms from *inclTerms* also found in *exclTerms*

• Return the sets *inclTerms* and *exclTerms*

where an ordered pair of terms ($T_i$, $T_j$) means that ($T_i$, $T_j$) ≠ ($T_j$, $T_i$). In the following sections we identify how each case assigns terms to *inclTerms* and *exclTerms* and adjusts the number of instances associated with each term used to compare annotations.

*Direct Cases*

The IS constraint where one term in an annotation is a special case of another term can be implemented as follows:

**1** if ($T_i$ IS $T_j$)

inclTerms ← inclTerms ∪ $T_i$

exclTerms ← exclTerms ∪ $T_j$

In this situation the term $T_j$ is viewed as being the common taxonomic ancestor of both terms.

The PART constraint where one term is a part of another term can be implemented as:

**2** if ($T_i$ PART $T_j$)

inclTerms ← inclTerms ∪ $T_i$

exclTerms ← exclTerms ∪ $T_j$

if (numInst ($T_j$) < numInst($T_i$))

numInst($T_j$) = numInst($T_i$)

In this situation the term $T_j$ is viewed as providing the context that instances of $T_i$ are part of.

The case is similar for $T_i$ MIXED $T_j$. In these cases we are relating terms that belong to two different lines of taxonomic inheritance where terms have a possibly incomplete set of associated instances. In order to ensure that the partial order constraint associated with this case is implemented correctly if $T_j$ has fewer instances associated with it than $T_i$ then we adjust the number of instances associated with $T_j$ to be equal to the number of instances associated with $T_i$.

The MIXED constraint where $T_i$ is a part of another term $T_j$ via an intermediate term $T_k$ can be implemented similarly to the PART case:

**3** if ($T_i$ MIXED $T_j$)

inclTerms ← inclTerms ∪ $T_i$

exclTerms ← exclTerms ∪ $T_j$

exclTerms ← exclTerms ∪ $T_k$

if (numInst($T_k$) < numInst($T_i$))

numInst($T_k$) = numInst($T_i$)

if (numInst($T_j$) < numInst($T_i$))

numInst($T_j$) = numInst($T_i$)

In this situation the term $T_k$ is viewed as providing the context that instances of $T_i$ are part of.

*Indirect Homogeneous Cases*

In the indirect homogeneous cases compared terms $T_i$ and $T_j$ are indirectly related via a common ancestor $T_{lca}$ along homogeneous paths. The first such case is where $T_i$ IS $T_{lca}$ and $T_j$ IS $T_{lca.}$ In this situation the number of instances associated with $T_{lca}$ provides a measure of similarity between $T_i$ and $T_j$:

**4** if ($T_i$ IS $T_{lca}$ &$T_j$ IS $T_{lca}$)

numInst($T_i$), numInst($T_j$) ← min(numInst($T_i$), numInst($T_j$))

inclTerms ← inclTerms ∪ $T_j$ ∪ $T_i$

exclTerms ← exclTerms ∪ $T_{lca}$

In the case where $T_i$ PART $T_{lca}$ and $T_j$ PART $T_{lca}$$T_{lca}$ provides the context in which instances of $T_i$ and $T_j$ are embedded.

**5** if ($T_i$ PART $T_{lca}$ &$T_j$ PART $T_{lca}$)

numInst($T_i$), numInst($T_j$) ← min(numInst($T_i$) ∩ numInst($T_j$))

inclTerms ← inclTerms ∪ $T_j$ ∪ $T_i$

exclTerms ← exclTerms ∪ $T_{lca}$

if (numInst($T_{lca}$) < numInst($T_i$))

numInst($T_{lca}$) = numInst($T_i$)

Since terms from two different lines of taxonomic inheritance are being compared and the set of instances associated with each term is incomplete an adjustment of the number of instances associated with each term is necessary.

The final homogeneous indirect case occurs when $T_i$ PART $T_{lca}$ and $T_j$ IS $T_{lca}$. This is equivalent to $T_i$ PART $T_j$ since if $T_i$ is a part of $T_{lca}$ and $T_j$ is a kind of $T_{lca}$ then $T_i$ is a part of $T_j$.

**6** else if ($T_i$ PART $T_{lca}$ &$T_j$ IS $T_{lca}$)

inclTerms ← inclTerms ∪ $T_i$

exclTerms ← exclTerms ∪ $T_j$

exclTerms ← exclTerms ∪ $T_{lca}$

if (numInst($T_j$) < numInst($T_i$))

numInst($T_j$) = numInst($T_i$)

if (numInst($T_{lca}$) < numInst($T_i$))

numInst($T_{lca}$) = numInst($T_i$)

As with other cases the number of instances associated with each term are adjusted to ensure that the partial order constraint associated with the case is satisfied.

*Indirect Inhomogeneous Cases*

In these cases one or both paths from $T_{lca}$ to terms $T_i$ and $T_j$ contain inhomogeneous types of relations. Throughout this section the term $T_k$ is a term in the ontology such that $T_m$ MIXED $T_k$ and $T_k$ IS $T_n$ if $T_n$ is an ancestor of $T_m$ in the ontology.

The first such case occurs where for two indirectly related terms being compared, $T_i$ and $T_j$, there exists an MIXED path from $T_i$ to $T_{lca}$ via $T_k$ and an IS path from $T_j$ to $T_{lca}$.

**7** if ($T_i$ MIXED $T_{lca}$ &$T_j$ IS $T_{lca}$)

inclTerms ← inclTerms ∪ $T_i$

exclTerms ← exclTerms ∪ $T_{lca}$

if (numInst($T_k$) < numInst($T_i$))

numInst($T_k$) = numInst($T_i$)

if (numInst($T_{lca}$) < numInst($T_k$))

numInst($T_{lca}$) = numInst($T_k$)

Since the relationship between $T_i$ and $T_j$ cannot be refined further than their relationship via $T_{lca}$ only $T_{lca}$ is assigned to *exclTerms*.

The second case occurs when $T_i$ MIXED $T_{lca}$ via $T_k$ and $T_j$ PART $T_{lca}$. Since $T_j$ is part of $T_{lca}$ and $T_i$ is part of $T_k$ which is a kind of $T_{lca}$ then $T_j$ is a part of $T_k$.

**8** if ($T_i$ MIXED $T_{lca}$ &$T_j$ PART $T_{lca}$)

inclTerms ← inclTerms ∪ $T_i$

inclTerms ← inclTerms ∪ $T_j$

exclTerms ← exclTerms ∪ $T_k$

exclTerms ← exclTerms ∪ $T_{lca}$

if (numInst($T_k$) < numInst($T_i$))

numInst($T_k$) = numInst($T_i$)

if (numInst($T_k$) < numInst($T_j$))

numInst($T_k$) = numInst($T_j$)

if (numInst($T_{lca}$) < numInst($T_k$))

numInst($T_{lca}$) = numInst($T_k$)

The final case occurs when both terms $T_i$ and $T_j$ are MIXED related to $T_{lca}$ via $T_k$ and $T_m$ respectively. What is common between both terms $T_i$ and $T_j$ is that they are both part of $T_{lca}$. The number of instances associated with each term is adjusted to satisfy the partial order constraints associated with this case.

**9** if ($T_i$ MIXED $T_{lca}$ &$T_j$ MIXED $T_{lca}$)

inclTerms ← inclTerms ∪ $T_i$

inclTerms ← inclTerms ∪ $T_j$

exclTerms ← exclTerms ∪ $T_{lca}$

if (numInst($T_k$) < numInst($T_i$))

numInst($T_k$) = numInst($T_i$)

if (numInst($T_m$) < numInst($T_j$))

numInst($T_m$) = numInst($T_j$)

if (numInst($T_{lca}$) < numInst($T_k$))

numInst($T_{lca}$) = numInst($T_k$)

if (numInst($T_{lca}$) < numInst($T_m$))

numInst($T_{lca}$) = numInst($T_m$)

After all terms have been compared with each other it is necessary to remove any terms from *inclTerms* that are found in *exclTerms*. This can occur when one comparison assigns a term to *inclTerms* while another comparison identifies the term as belonging to the excluded set. After all terms are compared each term in *inclTerms* should have the same number of instances associated with it. The number of instances that are associated with an annotation $G$ is equal to the minimum number of instances that can be associated with any of the terms in $inclTerms \cap G$.

### *Finding the Nearest Common Annotation*

Just as in semantic similarity of terms, where there is a common ancestor between two terms, there exists a nearest common annotation between two annotations. The concept of a nearest common annotation allows the extension of information based semantic similarity measures of terms, such as Resnik's and Lin's measures, to information based measures of semantic similarity of annotations.

We define the *nearest common annotation* (NCA) between two annotations $G_1$ and $G_2$ to be the annotation containing terms related to both annotations. The NCA should have the minimum possible number of instances associated with it such that either $G_1$ or $G_2$ can be derived from it. The set of terms *exclTerms* which results from applying SSA to two annotations $G_1$ and $G_2$ will return the set of terms associated with the NCA.

### *Measuring Similarity*

By introducing the notion of nearest common annotation we can naturally extend Resnik's measure to measuring similarity of annotation. The LCA between two terms is replaced with the NCA of two annotations $G_1$ and $G_2$. Likewise, instead of applying $IC_{Corpus}$ (eqn. 1) to instances associated with a term we apply $IC_{Corpus}$ to instances of an annotation. Thus the extension of Resnik's measure from terms to annotations $G_1$ and $G_2$, $SSA_{Resnik}$, becomes:

$$\text{exclTerms} \leftarrow SSA(G_1, G_2)$$

$$SSA_{Resnik}(G_1, G_2) = -\log\left(\frac{\min_{T_i \in exclTerms} numInst(T_i)}{\max NumInst}\right)$$

where *maxNumInst* is the number of distinct instances in the corpus.

Lin's measure may be extended as follows:

$$\text{inclTerms1} \leftarrow SSA(G_1, G_1)$$

$$\text{inclTerms2} \leftarrow SSA(G_2, G_2)$$

$$icG1 \leftarrow -\log\left(\frac{\min_{T_i \in inclTerms1} numInst(T_i)}{maxNumInst}\right)$$

$$icG2 \leftarrow -\log\left(\frac{\min_{T_j \in inclTerms2} numInst(T_j)}{maxNumInst}\right)$$

$$SSA_{Lin}(G_1, G_2) = \frac{2 * SSA_{Resnik}(G_1, G_2)}{icG1 + icG2}$$

In this case the SSA algorithm is used to find the non redundant terms that can be associated with an annotation.

### *Example*

We compare the similarity of two gene product's annotations that returns a high measure of similarity when compared using our measure $SSA_{Resnik}$. Two gene products, AAH1 and FUR1 whose annotations (listed in table 2) were taken from the SGD database [27] were compared producing a similarity value of 5.678. The number of instances associated with each term were obtained from the GOA [28]*s. cerevisiae* table of GO assignments.

FUR1's annotation consisted of six terms: {GO:0004845, GO:0005622, GO:0008655, GO:0009116, GO:0016740, GO:0016757}. Each term's description is found in table 2. Likewise, AAH1's annotation consists of twelve terms: {GO:0000034, GO:0004000, GO:0005634, GO:0005737, GO:0006146, GO:0009117, GO:0009168, GO:0016787, GO:0019239, GO:0042254, GO:0043101, GO:0043103}. The NCA is constructed by applying the SSA algorithm to identify the set of contextual terms common to both annotations. Terms such as the root term 'all' are immediately added to *exclTerms*. The term 'cellular component' (GO:0005575) is added to *exclTerms* since another term 'cell part' is is_a related to it. The term 'nucleobase metabolic process' (GO:0009112) is a more specific

**Table 2: Example Annotations and Their Descriptions**

| Gene | Term | Description |
|---|---|---|
| AAH1 | GO:0000034 | adenine deaminase activity |
| | GO:0004000 | adenosine deaminase activity |
| | GO:0005634 | nucleus |
| | GO:0005737 | cytoplasm |
| | GO:0006146 | adenine catabolic process |
| | GO:0009117 | nucleotide metabolic process |
| | GO:0009168 | purine ribonucleoside monophosphate biosynthetic process |
| | GO:0016787 | hydrolase activity |
| | GO:0019239 | deaminase activity |
| | GO:0042254 | ribosome biogenesis and assembly |
| | GO:0043101 | purine salvage |
| | GO:0043103 | hypoxanthine salvage |
| FUR1 | GO:0004845 | uracil phosphoribosyltransferase activity |
| | GO:0005622 | intracellular |
| | GO:0008655 | pyrimidine salvage |
| | GO:0009116 | nucleoside metabolic process |
| | GO:0016740 | transferase activity |
| | GO:0016757 | transferase activity, transferring glycosyl groups |

type of 'nucloebase, nucleoside and nucleotide process' (GO:0055086) and the terms are added to *inclTerms* and *exclTerms* respectively. Similar assignments occur for 'nucleobase metabolic process' (GO:0009112)/'cellular metabolic process' (GO:0044237), 'nucleobase metabolic process' (GO:0009112)/'cellular process' (GO:0009987) as well as other terms.

The SSA algorithm return nine contextual terms, {'all' (all), 'cellular process' (GO:0009987), 'cellular metabolic process' (GO:0044237), 'nucleobase metabolic process' (GO:0009112), 'nucleobase, nucleoside, nucleotide and nucleic acid metabolic process' (GO:0006139), 'nucleobase, nucleoside and nucleotide metabolic process' (GO:0055086), 'cell part' (GO:0044464), 'intracellular' (GO:0005622), 'catalytic activity' (GO:0003824), 'metabolic compound salvage' (GO:0043094)}. The resulting annotation contains terms from all three ontologies in the GO. There are 19 instances associated with the annotation. The number of instances is determined by the most specific term: 'metabolic compound salvage' (GO:0043094). The total number of instances in the corpus is 5554. $SSA_{Resnick} = -\log\left(\frac{19}{5554}\right) \approx 5.678$ . Since the highest value that $SSA_{Resnik}$ could return for the chosen corpus is ∼8.622, taking the natural log of $\frac{1}{5554}$ , 5.678 corresponds to high degree of similarity.

## Results

To validate our approach the discriminatory power of our method to identify clusters of related gene products was compared against Wang's measure of annotation similarity that also exploits the differences between types of relations. The average similarity of gene products found in the same biochemical pathway in the SGD database was compared against the average similarity of the same gene products compared with gene products found in other pathways. A large difference between these two values indicates the effectiveness of a similarity measure in discovering new pathways in a set of gene products. Average similarity of annotations inside and outside pathways was measured under four conditions: all terms; cellular component terms only; biological process terms only; and molecular function terms only.

A better test would be to take the average similarity of a set of gene products found in the same pathway and find the average or max of the average similarities of all other similarly sized sets of gene products. Of course this is intractable since the computational complexity of such a test is O(n!) since there are $\binom{N}{n}$ ways of creating a set of size n from a set of N elements.

Figure 3 show the results of a comparison of $SSA_{Resnik}$ with Wang's method and $Max_{Resnik}$ on measuring the average annotation similarity, using all terms, of gene products inside and outside a pathway [data for figures 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 is found in Additional file 1]. The first 35 pathways are insuffi-

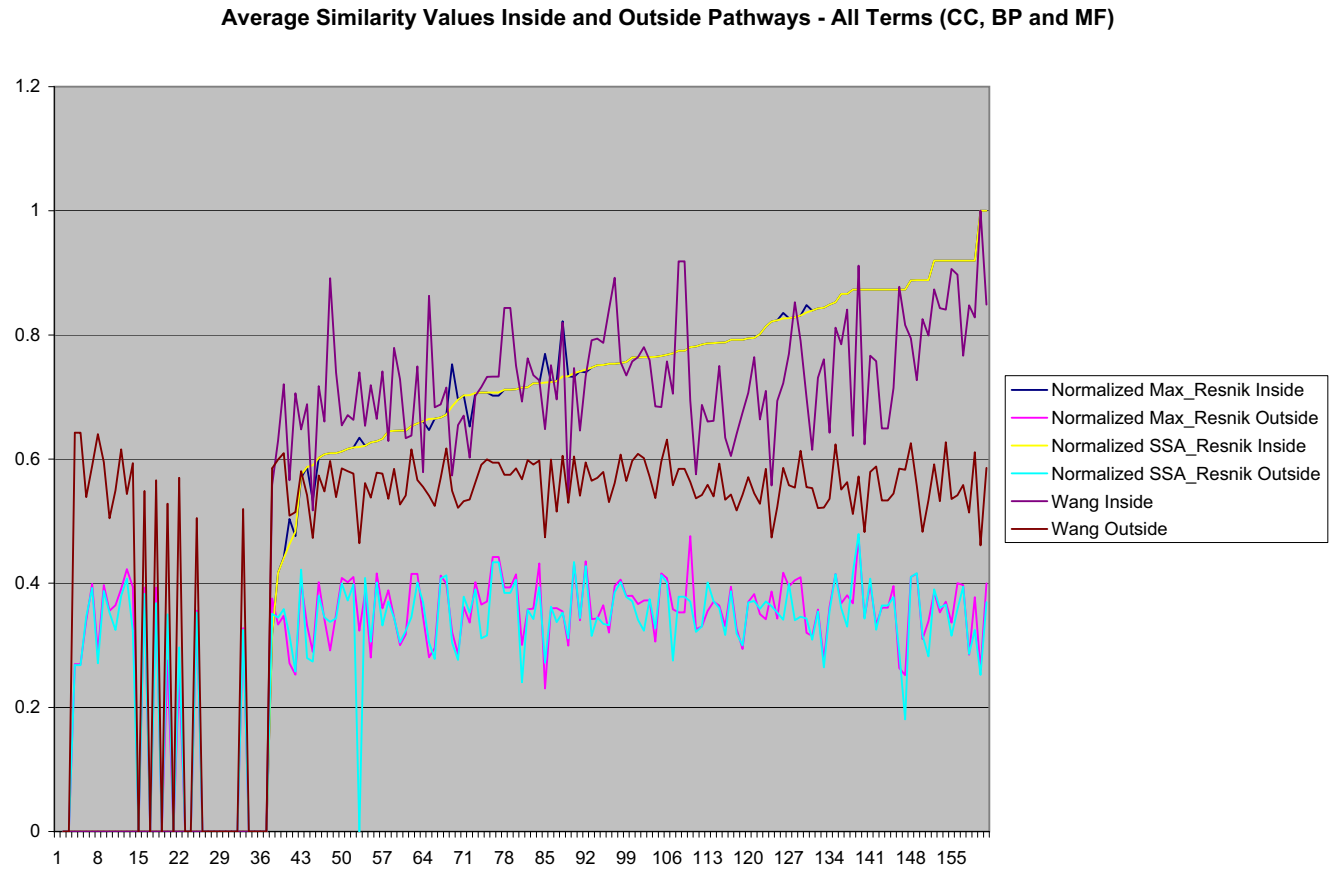



**Figure 3**

**Normalized $SSA_{Resnik}$ vs Wang's Method vs Normalized $Max_{Resnik}$.** Values shown correspond to the average annotation similarity values between gene products with other gene products in the same pathway (taken from the SGD biochemical pathways database) and between gene products in a pathway with other gene products not found in the pathway.

ciently annotated to produce meaningful results. Similarity values for $SSA_{Resnik}$ and $Max_{Resnik}$ were normalized to allow for direct comparison between similarity values. All measures behave similarly, the similarity values returned by Wang's method tends to increase as values returned by $SSA_{Resnik}$ increase. All measures tend to settle to an average similarity value when genes inside and outside a pathway are compared. Wang's method returns a higher value on average with values ranging between 0.5 and 0.6 as internal gene similarity increases. $SSA_{Resnik}$ and $Max_{Resnik}$ returns values between 0.3 and 0.4 for the average similarity value of genes inside a pathway with genes outside a pathway as similarity of genes within a pathway increases. If pathways are identified by the difference between the average similarity of gene products inside and outside a cluster then $SSA_{Resnik}$ and $Max_{Resnik}$ have greater discriminatory power. $SSA_{Resnik}$ and $Max_{Resnik}$ behave identically for most pathways when all terms are considered.

As shown in figures 4, 5, 6, when only terms from the cellular component sub-ontology are used the difference between SSAResnik and MaxResnik becomes clear. MaxResnik returns a very high average similarity value between terms inside and outside a pathway. This may be an artifact of the low number of instances associated with cellular component terms. However when SSA is applied the average similarity values between annotations inside and outside pathways remains consistently low. SSAResnik returns a comparatively high average similarity value for annotations inside pathways for approximately half the cases to which it can reasonably be applied. Wang's method behaves similarly to MaxResnik in this situation.

As shown in figures 7, 8, 9, if only biological process terms are used further dissimilarity between $Max_{Resnik}$ and $SSA_{Resnik}$ can be observed. The average similarity values of annotations inside a pathway with annotations outside a pathway is much higher for $Max_{Resnik}$ than for $SSA_{Resnik}$. Wang's method and $SSA_{Resnik}$ behave similarly. Similarity

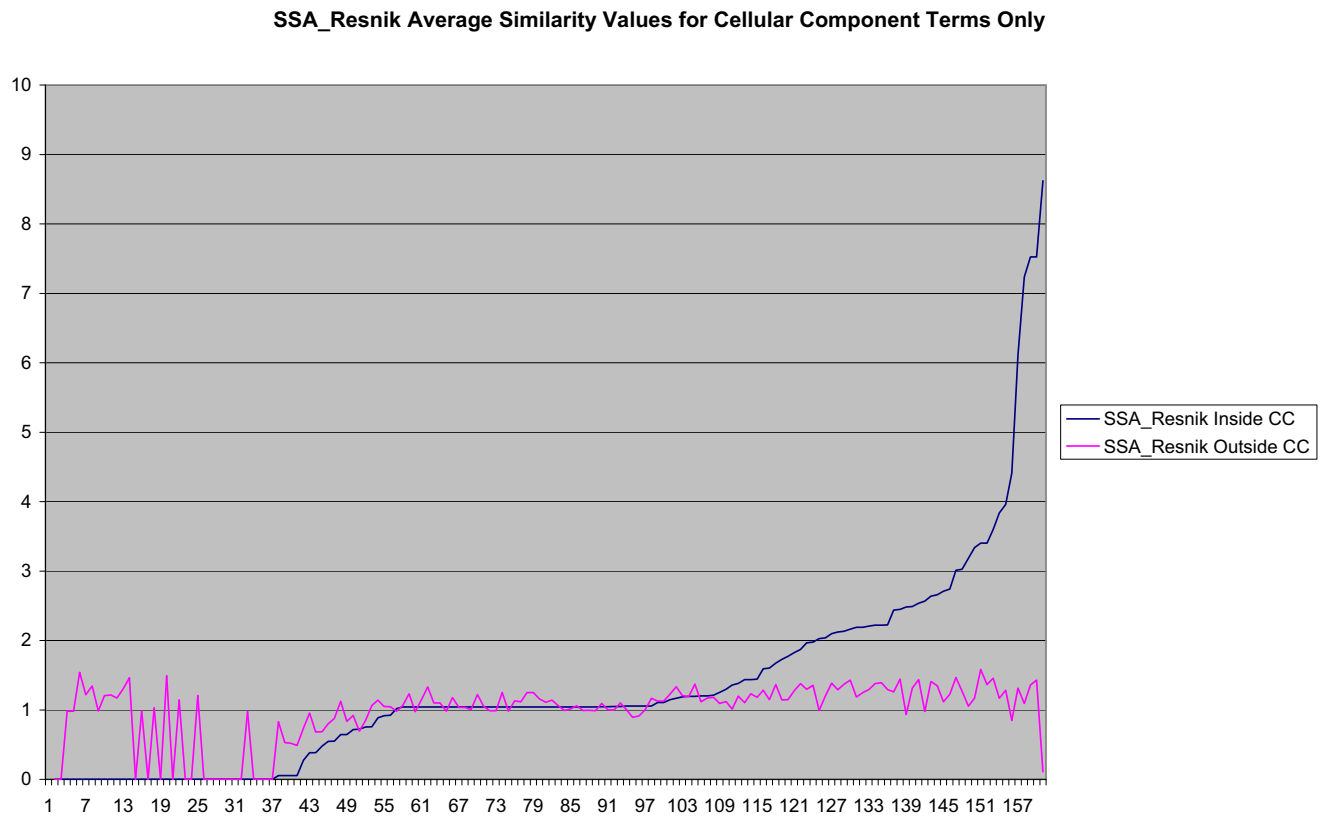



**Figure 4**
**Average Pathway Similarity Values of Annotations Consisting only of Cellular Component Terms Using $SSA_{Resnik}$.** Average of $SSA_{Resnik}$ similarity values of gene products inside and outside a pathway.

values of annotations inside a pathway remain consistently higher than when the same annotations are compared with annotations outside the pathway for all methods.

The source of the similarity between $SSA_{Resnik}$ and $Max_{Resnik}$ can be identified when only molecular function terms are used, as shown in figures 10 and 11. In this case both methods behave exactly the same since there are no part of relations to exploit when comparing terms. Wang's method, shown in figure 12, returns a consistently high average similarity value for annotations inside a pathway compared with annotations outside a pathway.

Further discriminatory power can be achieved by considering the standard deviation of similarity values inside and outside a pathway. A set of gene products paired with other gene products in a pathway tend to have a high standard deviation of similarity values over all pairs mainly due to the small number of pairs being compared. Conversely, pairing gene products inside a pathway with those found outside the pathway should produce a set of similarity values with a lower standard deviation since annotations are expected to be dissimilar and values come from a larger set.

Figures 13, 14, 15 shows the standard deviation of similarity values of annotations consisting of cellular component terms inside pathways. $Max_{Resnik}$ returns a low internal standard deviation while reporting a consistently high standard deviation of similarity values when annotations inside a pathway are compared with annotations outside a pathway. The standard deviation of annotation similarity values between different pathways returned by both $SSA_{Resnik}$ and Wang's method are both consistently low. The standard deviation of all methods behave similarly as average similarity of annotations, consisting only of biological process terms, within pathways increase, as shown in figures 16, 17, 18. The same is also true of annotations consisting of molecular function terms, as shown in figures 19, 20, 21.

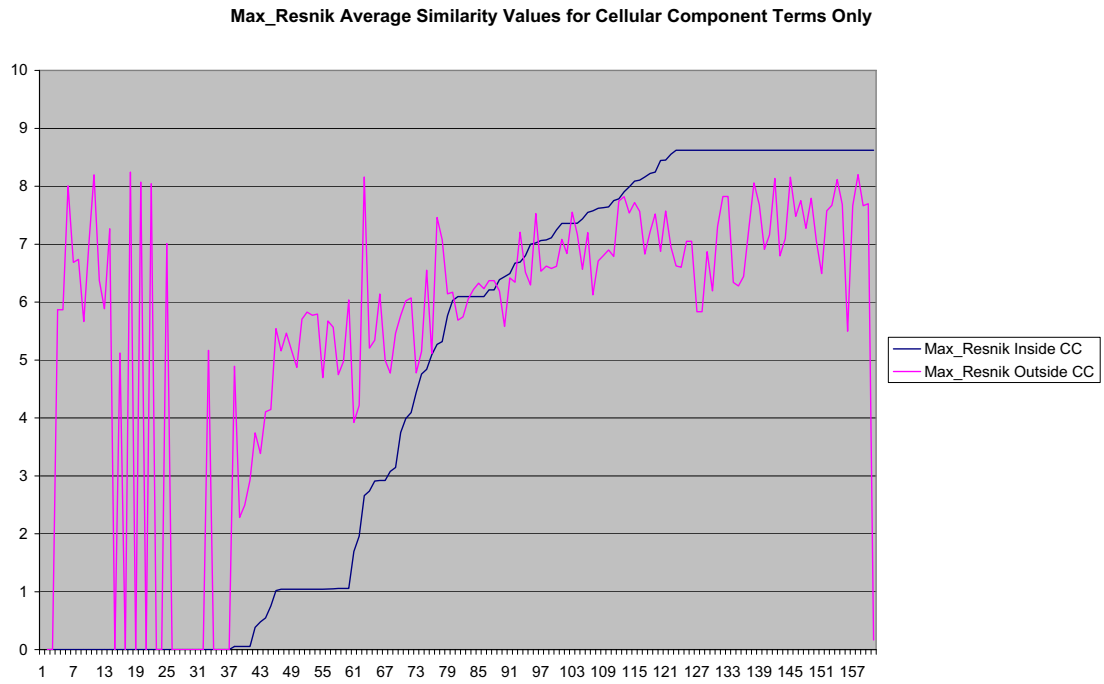



**Figure 5**
**Average Pathway Similarity Values of Annotations Consisting only of Cellular Component Terms Using $Max_{Resnik}$.** Average of $Max_{Resnik}$ similarity values of gene products inside and outside a pathway.

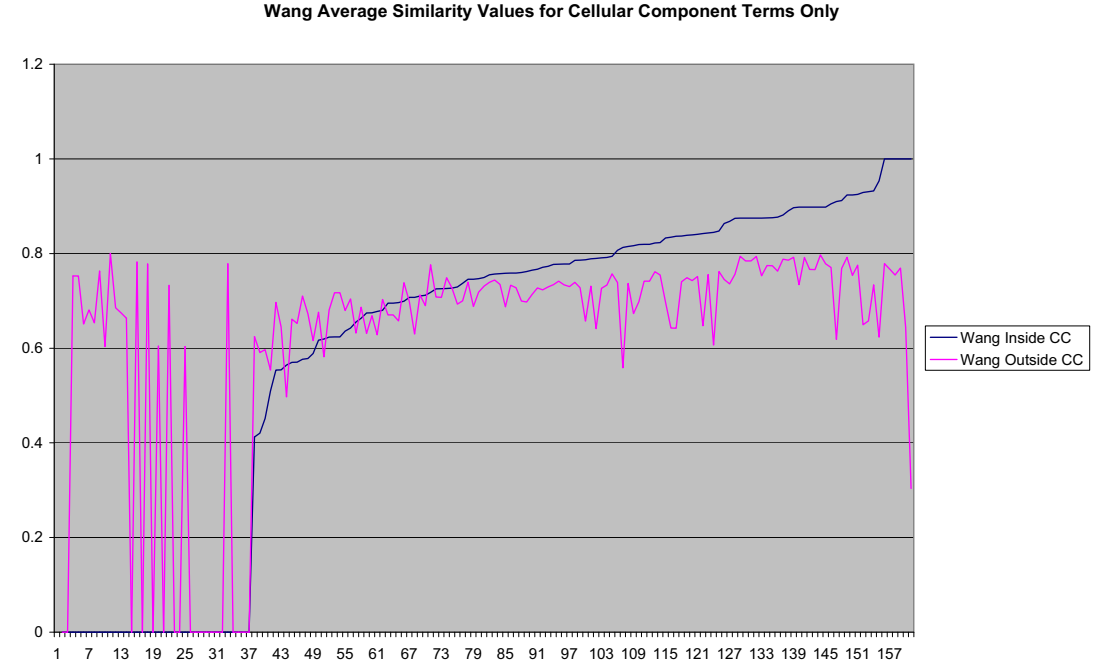
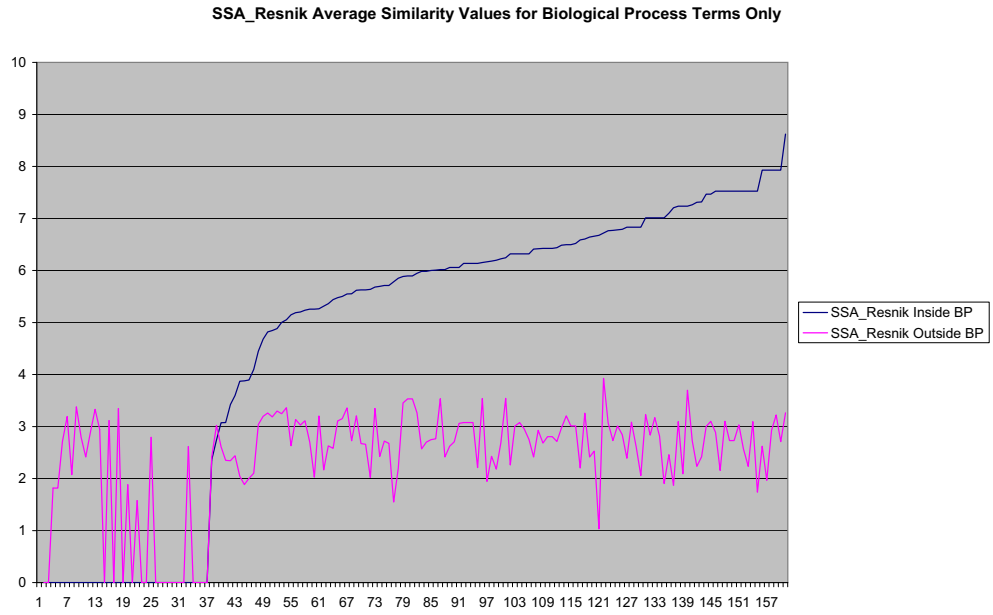



**Figure 6**
**Average Pathway Similarity Values of Annotations Consisting only of Cellular Component Terms Using Wang's Method**. Average of Wang's measure of similarity of gene products inside and outside a pathway.

SSA_Resnik Average BP Only




Page 1

**Figure 7**
**Average Pathway Similarity Values of Annotations Consisting only of Biological Process Terms Using $SSA_{Resnik}$.** Average of $SSA_{Resnik}$ similarity values of gene products inside and outside a pathway.

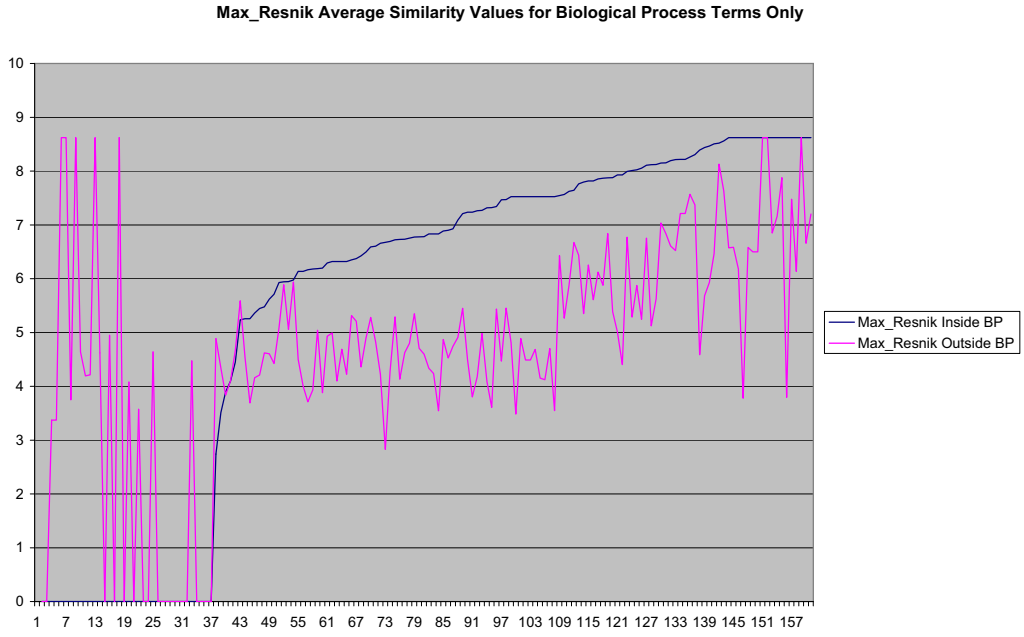



**Figure 8**
**Average Pathway Similarity Values of Annotations Consisting only of Biological Process Terms Using $Max_{Resnik}$.** Average of $Max_{Resnik}$ similarity values of gene products inside and outside a pathway.

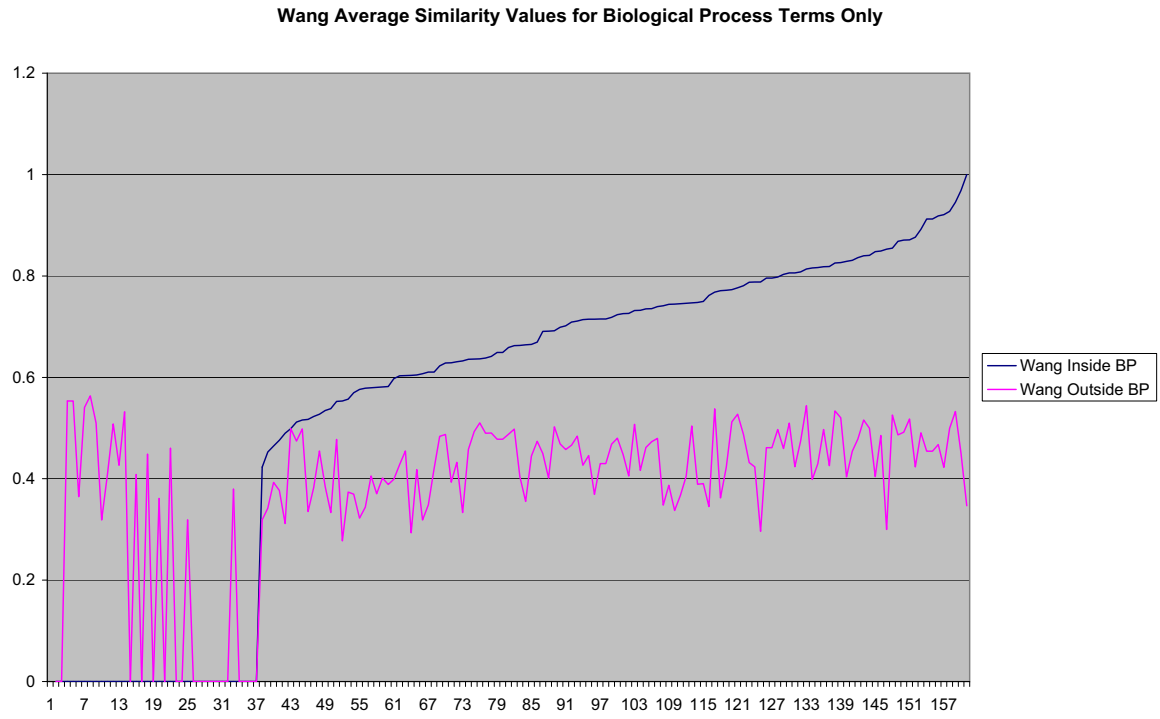



**Figure 9**
**Average Pathway Similarity Values of Annotations Consisting only of Biological Process Terms Using Wang's Method**. Average of Wang's measure of similarity of gene products inside and outside a pathway.

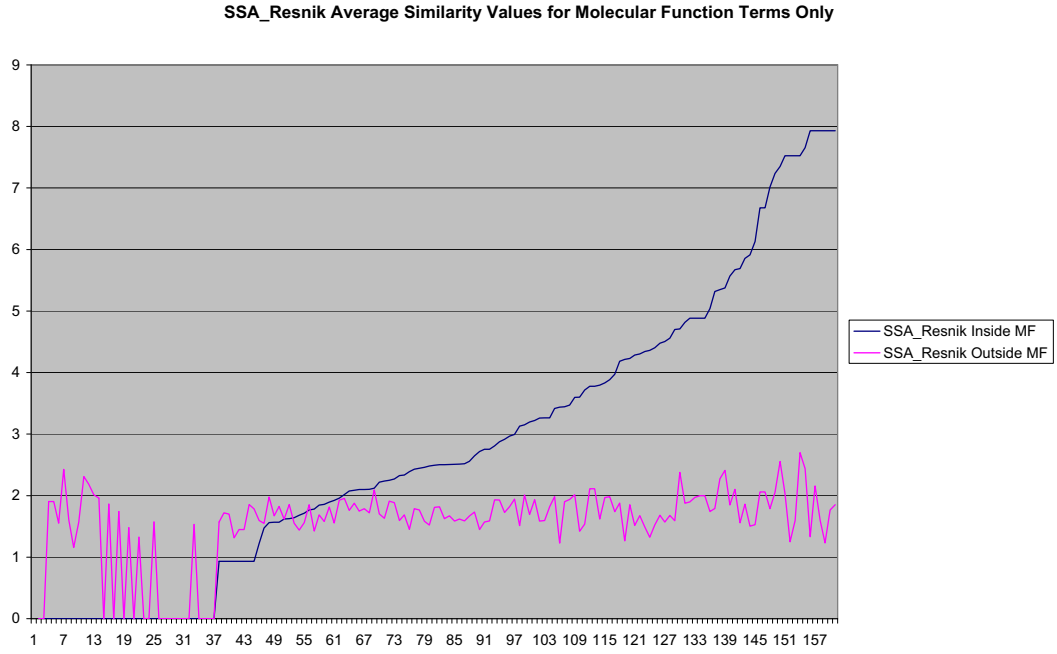



**Figure 10**
**Average Pathway Similarity Values of Annotations Consisting only of Molecular Function Terms Using $SSA_{Resnik}$.** Average of $SSA_{Resnik}$ similarity values of gene products inside and outside a pathway.

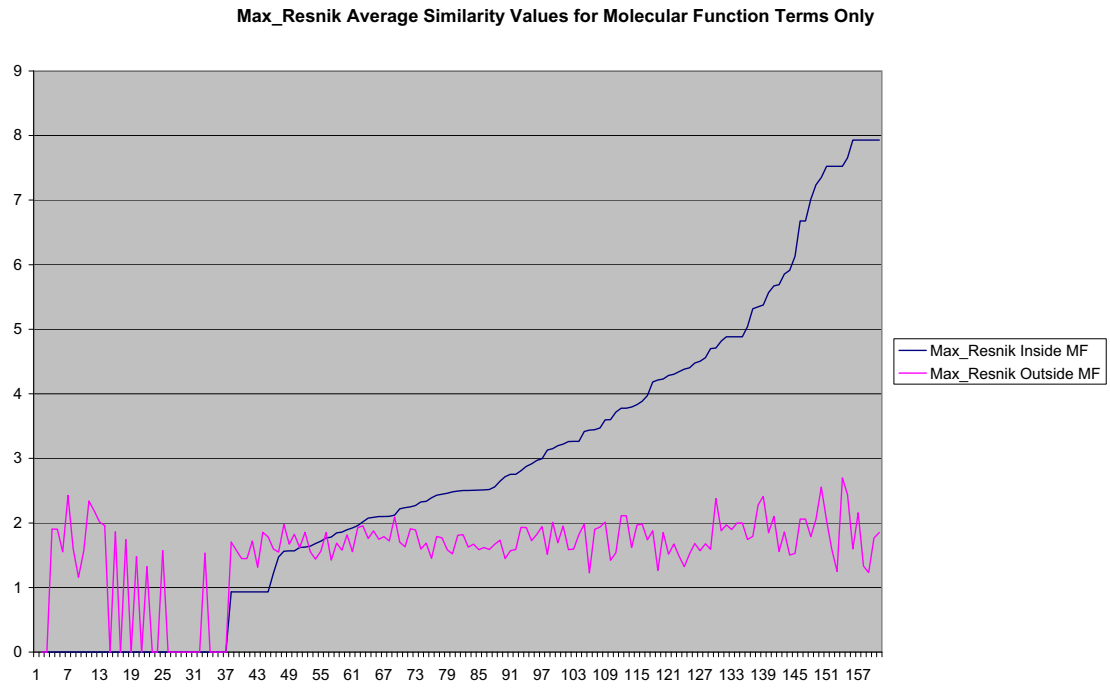



**Figure 11**
**Average Pathway Similarity Values of Annotations Consisting only of Molecular Function Terms Using $Max_{Resnik}$.** Average of $Max_{Resnik}$ similarity values of gene products inside and outside a pathway.

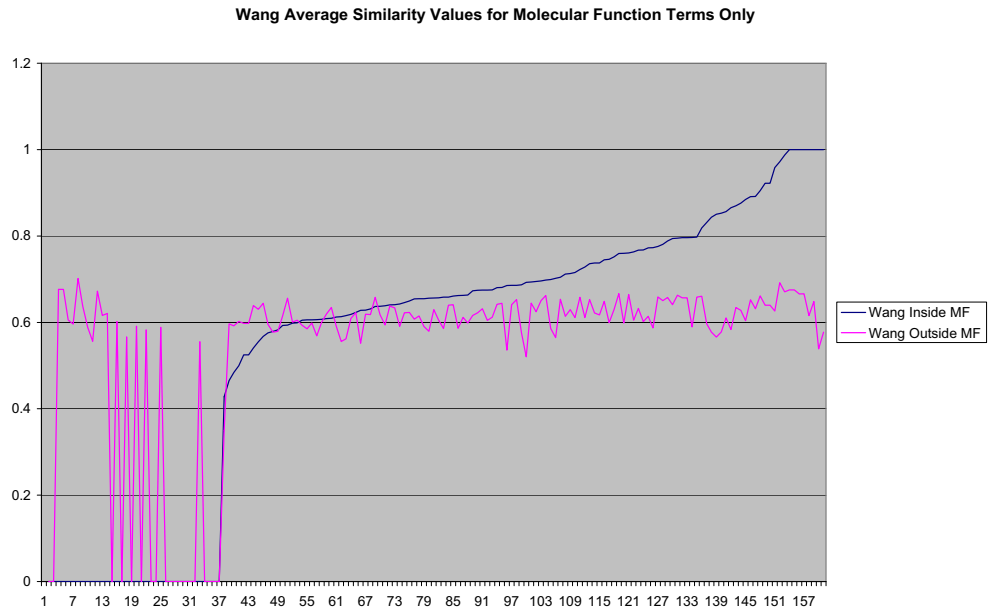



**Figure 12**
**Average Pathway Similarity Values of Annotations Consisting only of Molecular Function Terms Using Wang's Method**. Average of Wang's measure of similarity of gene products inside and outside a pathway.

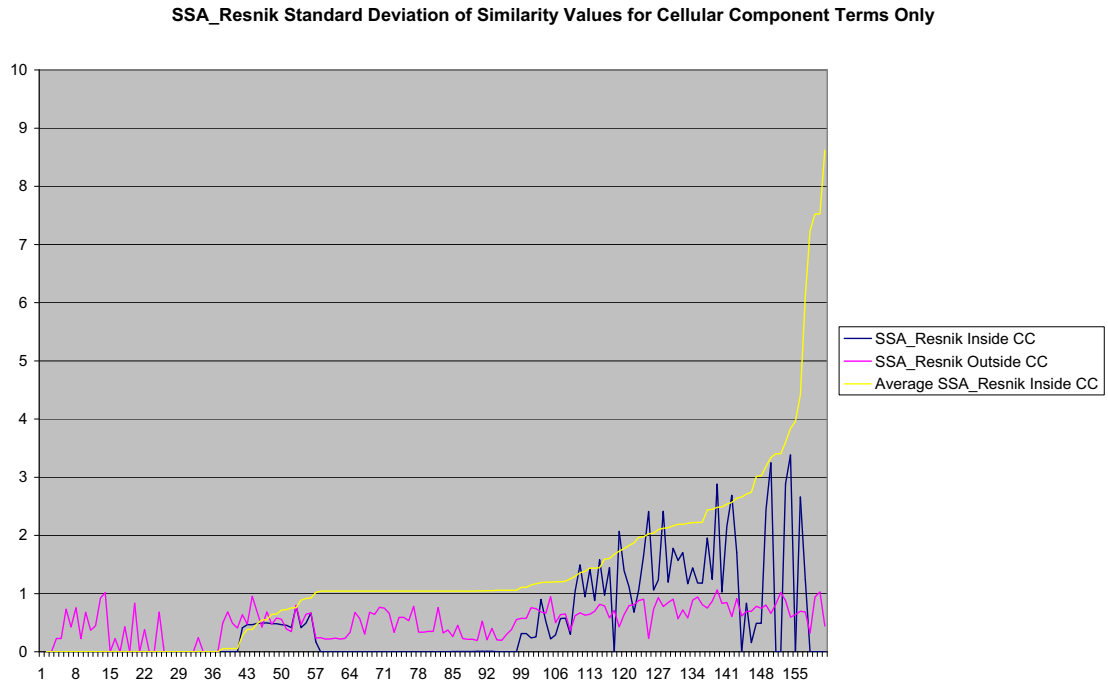



**Figure 13**
**Standard Deviation of Pathway Similarity Values of Annotations Consisting only of Cellular Component Terms Using $SSA_{Resnik}$.** Standard deviation of $SSA_{Resnik}$ similarity values of gene products inside and outside a pathway.

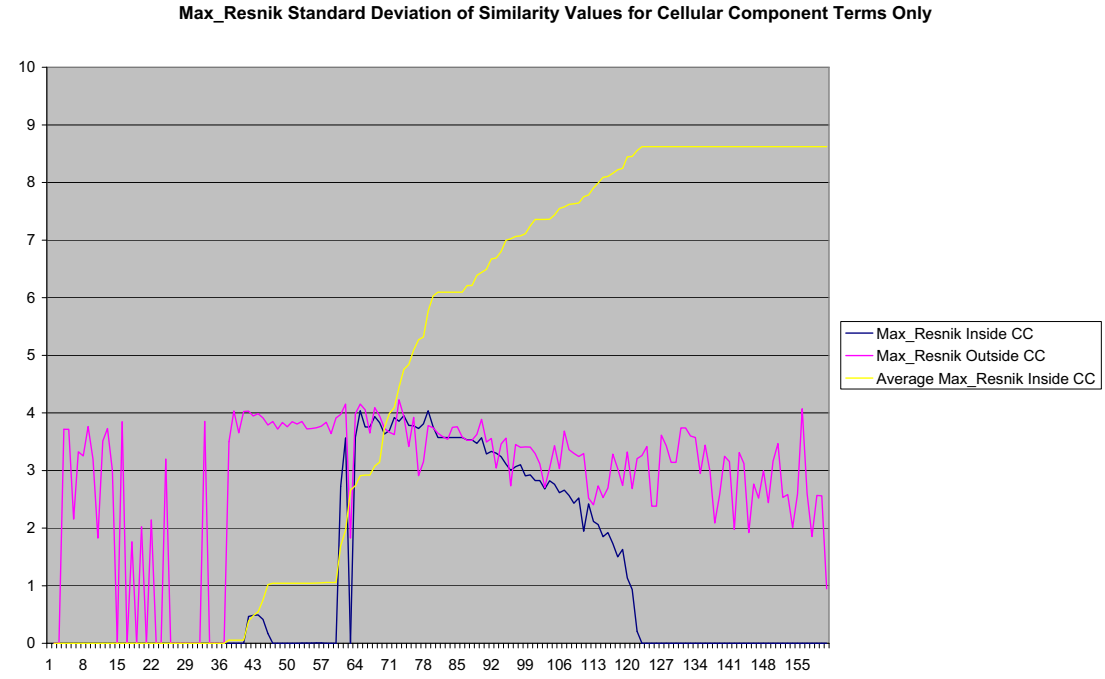



**Figure 14**
**Standard Deviation of Pathway Similarity Values of Annotations Consisting only of Cellular Component Terms Using $Max_{Resnik}$.** Standard deviation of $Max_{Resnik}$ similarity values of gene products inside and outside a pathway.

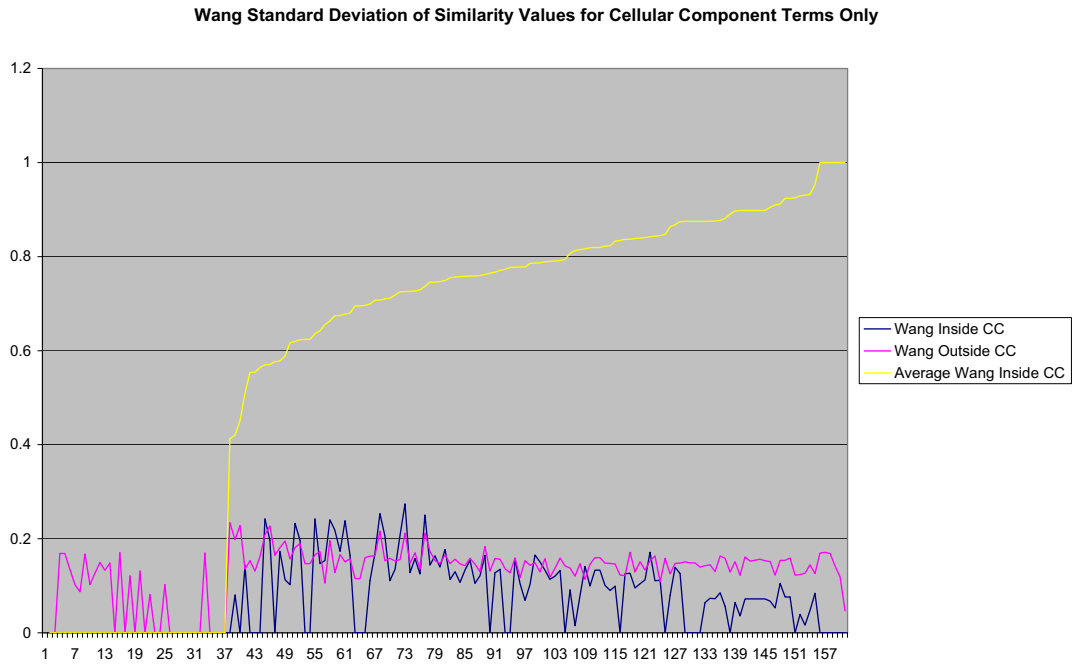



**Figure 15**
**Standard Deviation of Pathway Similarity Values of Annotations Consisting only of Cellular Component Terms Using Wang's Method**. Standard deviation of values of Wang's measure of similarity of gene products inside and outside a pathway.

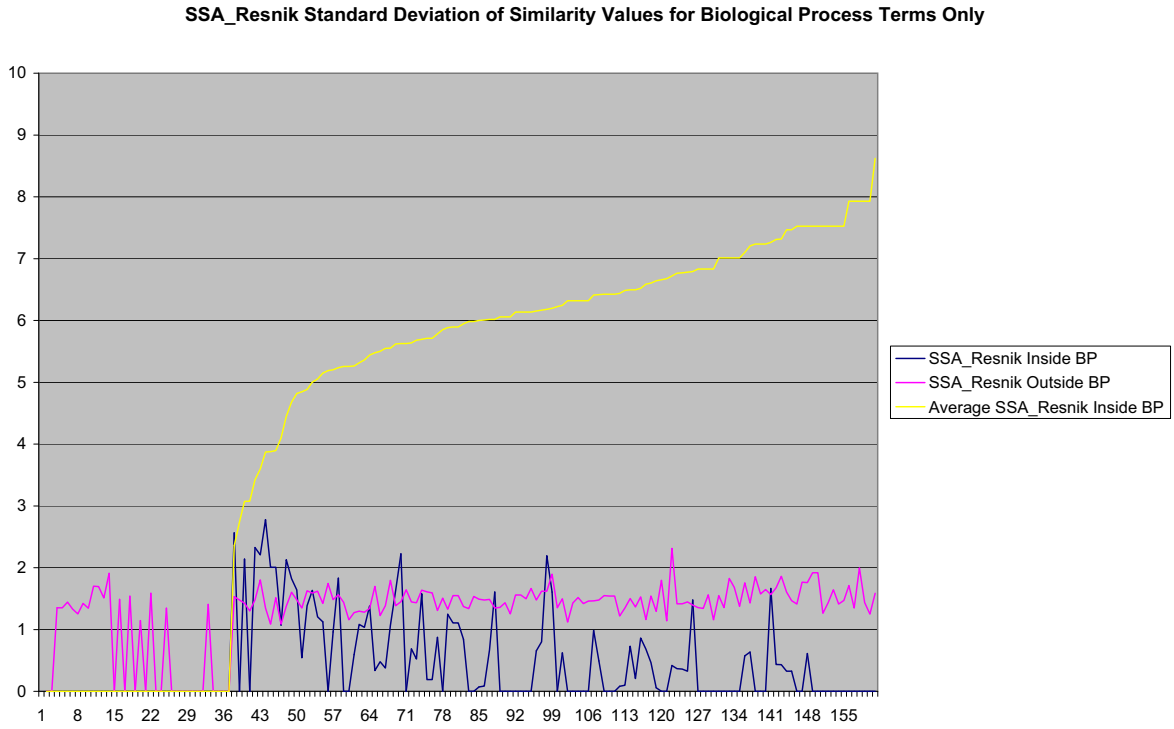



**Figure 16**
**Standard Deviation of Pathway Similarity Values of Annotations Consisting only of Biological Process Terms Using $SSA_{Resnik}$.** Standard deviation of $SSA_{Resnik}$ similarity values of gene products inside and outside a pathway.

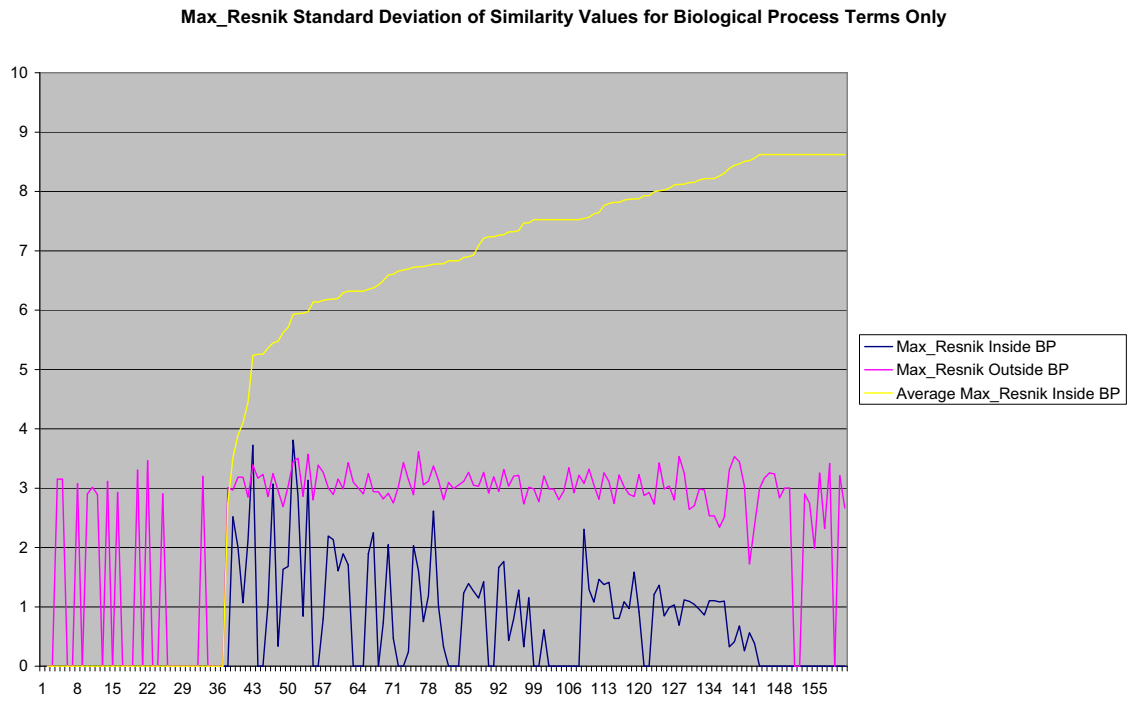



**Figure 17**
**Standard Deviation of Pathway Similarity Values of Annotations Consisting only of Biological Process Terms Using $Max_{Resnik}$.** Standard deviation of $Max_{Resnik}$ similarity values of gene products inside and outside a pathway.

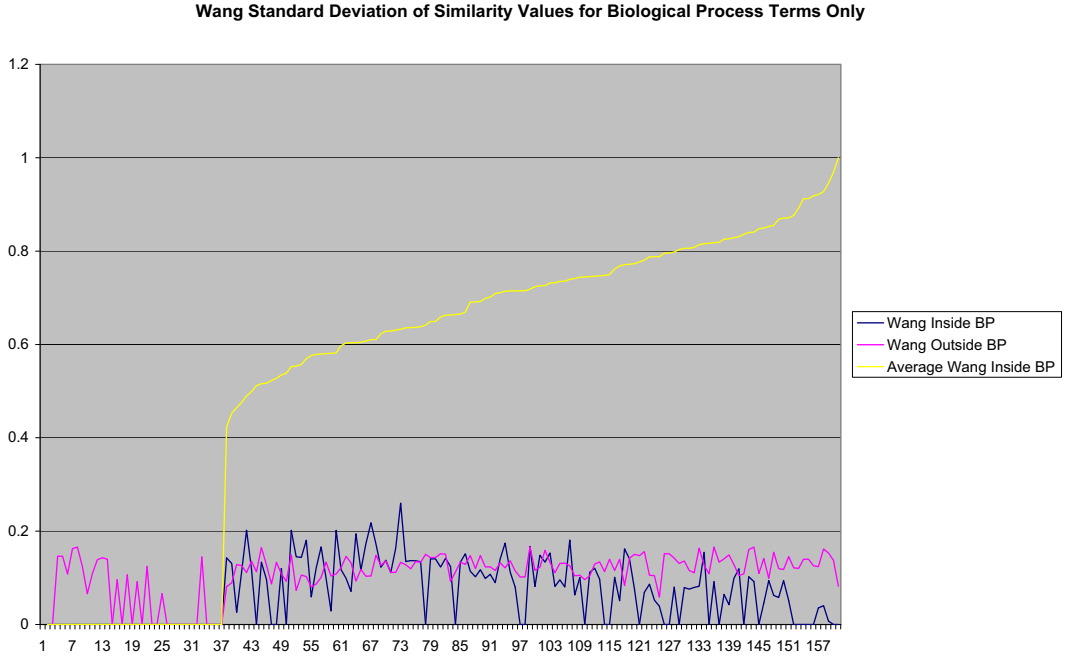



**Figure 18**
**Standard Deviation of Pathway Similarity Values of Annotations Consisting only of Biological Process Terms Using Wang's Method**. Standard deviation of values of Wang's measure of similarity of gene products inside and outside a pathway.

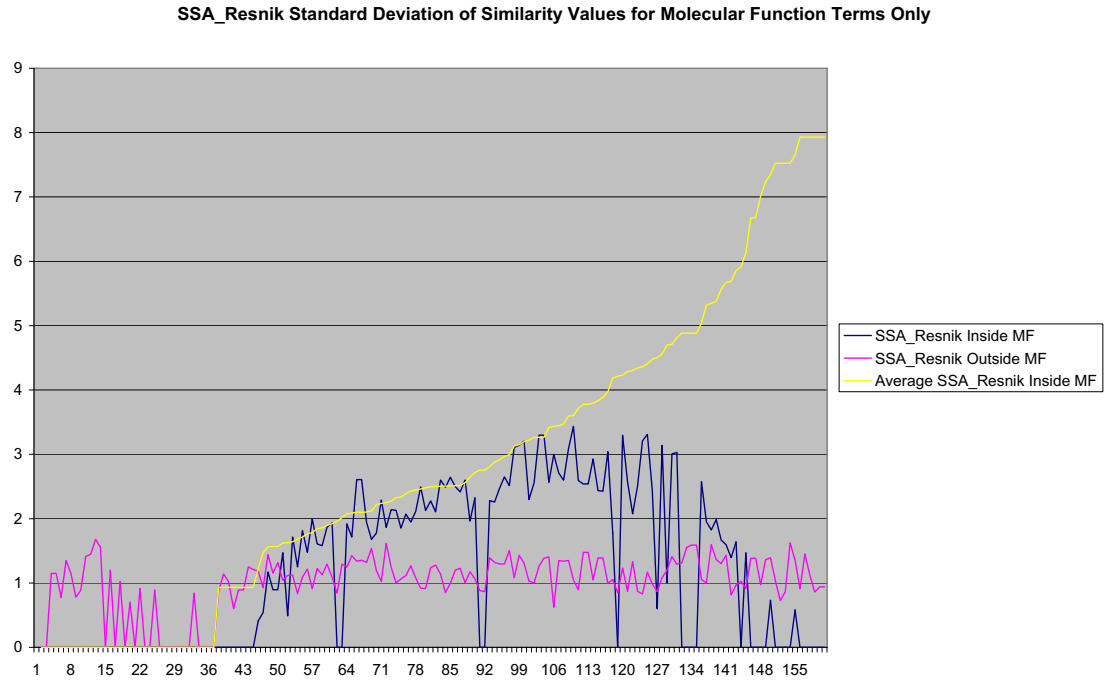



**Figure 19**
**Standard Deviation of Pathway Similarity Values of Annotations Consisting only of Molecular Function Terms Using $SSA_{Resnik}$.** Standard deviation of $SSA_{Resnik}$ similarity values of gene products inside and outside a pathway.

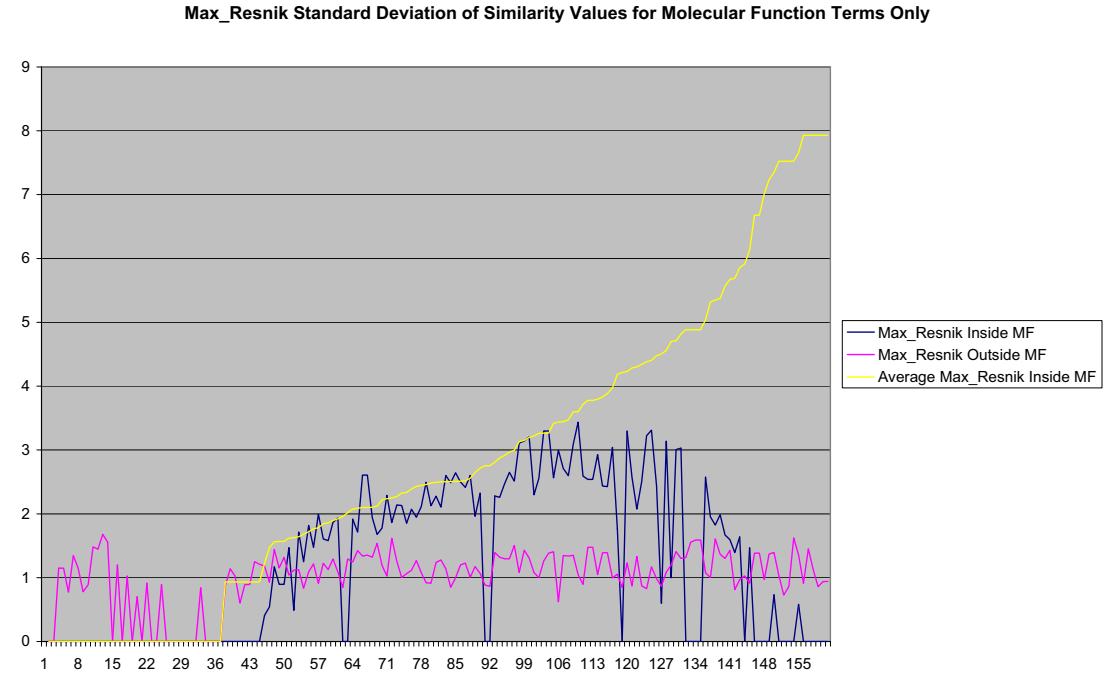



**Figure 20**
**Standard Deviation of Pathway Similarity Values of Annotations Consisting only of Molecular Function Terms Using $Max_{Resnik}$.** Standard deviation of $Max_{Resnik}$ similarity values of gene products inside and outside a pathway.

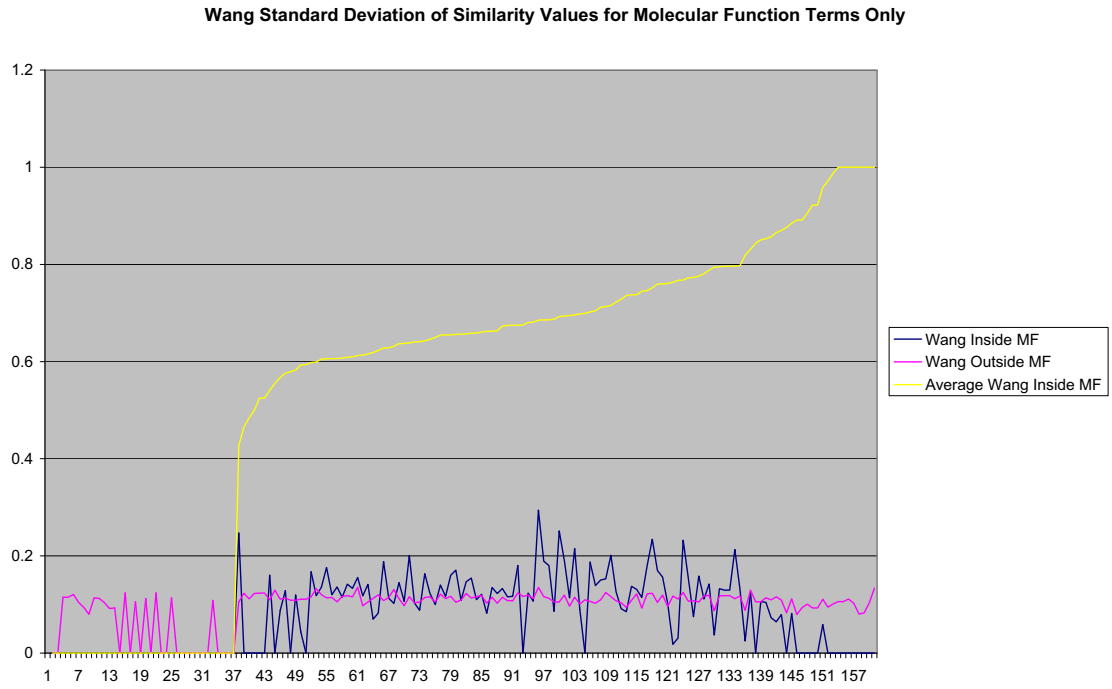



**Figure 21**
**Standard Deviation of Pathway Similarity Values of Annotations Consisting only of Molecular Function Terms Using Wang's Method**. Standard deviation of values of Wang's measure of similarity of gene products inside and outside a pathway.

## Discussion and conclusion

The SSA algorithm provides the basis of a framework for extending instance based measures of term similarity to annotations. The algorithm's construction is based on the set of cases for how terms are related to each other when the ontology consists only of is_a and part_of relations. Due to the incomplete nature of the set of instances associated with a term it is necessary to adjust the number of instances associated with a term in order to satisfy the partial order constraints of each case fully. As the number of annotations of gene products increase and ontological terms are applied more consistently it may be possible to satisfy the constraints without such adjustment. Alternatively, the partial order constraints can be used to develop a similarity method which is less dependent on the set of instances associated with terms.

When terms from all three sub-ontologies (CC, BP and MF) are used similarity of annotations between $Max_{Resnik}$ and $SSA_{Resnik}$ are equivalent on proteins found in the SGD database. This is due to the high degree of specificity of molecular function terms, which are not related partonomically, which causes the two measures to return the same values. When only cellular component and biological process terms are used, based on the experimental evidence, $SSA_{Resnik}$ becomes a better identifier of proteins belonging to pathways. $SSA_{Resnik}$ may identify new gene products that belong to pathways but have a different molecular function to those proteins already identified as belonging to the pathway. Molecular function terms only play a small role in identifying new pathway proteins since proteins tend to have different molecular functions inside pathways.

By finding the set of instances that can be associated with an annotation it is possible to preserve, at the annotation level, the properties of instance based methods used to measure the similarity of terms. For two given annotations, the nearest common annotation (NCA) is a minimal set of terms such that either annotation could be derived from it. The SSA algorithm provides a method for finding the set of terms associated with the NCA.

By combining the SSA algorithm with Resnik's measure and the concept of nearest common annotation we have developed a measure that provides good discriminatory power to identify possible pathways and other functional groups from gene product annotations. More generally, the set of cases and their associated constraints further

extend the set of principles that a reasonable measure of annotation similarity should be built on.

## Competing interests

The authors declare that they have no competing interests.

## Authors' contributions

BS proposed, designed and implemented the algorithm and table of constraints. BS wrote the manuscript. AQ and BG supervised and approved the production of this paper. SD contributed helpful suggestions for the final manuscript.

## Additional material

### Additional file 1

***Averages and Standard Deviations of Similarity Values.*** *Averages and standard deviations of similarity values of* $Max_{Resnik}$, $SSA_{Resnik}$ *and Wang's method for each pathway in SGD.*
Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-9-468-S1.xls]

## Acknowledgements

This work has been supported by Microsoft Research Cambridge and the Irish Research Council for Science, Engineering and Technology.

## References

1. Lord P, Stevens R, Brass A, Goble CA: **Semantic Similarity Measures as Tools for Exploring the Gene Ontology.** *Pacific Symposium on Biocomputing* 2003, **8:**601-612.
2. Lord P, Stevens R, Brass A, Goble C: **Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation.** *Bioinformatics* 2003, **19(10):**1275-1283.
3. Sevilla J, Segura V, Podhorski A, Guruceaga JE Mato, Martinez-Cruz L, Corrales F, Rubio A: **Correlation between gene expression and GO semantic similarity.** *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2005, **2(4):**330-338.
4. Couto FM, Silva MJ, Coutinho PM: **Measuring semantic similarity between Gene Ontology terms, Data and Knowledge Engineering.** *Business Process Management – Where business processes and web services meet* 2007, **61:**137-152.
5. Lei Z, Dai Y: **Assessing protein similarity with Gene Ontology and its use in subnuclear localization prediction.** *BMC Bioinformatics* 2006, **7:**491.
6. Goodman N: **Seven strictures on similarity.** In *Problems and Projects* Edited by: Goodman N. New York: Bobbs-Merrill; 1972:437-447.
7. Arrell D: **What Goodman Should Have Said about Representation.** *The Journal of Aesthetics and Art Criticism Autumn* 1987, **46:**41-49.
8. Tversky A: **Features of Similarity.** *Psychological Rev* 1977, **84:**327-352.
9. Lin D: **An Information-Theoretic Definition of Similarity.** In *Fifteenth International Conference on Machine Learning (ICML'98)* Madison, WI: Morgan-Kaufmann; 1998.
10. Popescu M, Keller J, Mitchell J: **Fuzzy Measures on the Gene Ontology for Gene Product Similarity.** *IEEEIACM Transactions on computational biology and bioinformatics* 2006, **3(3):**263-274.
11. Cross V: **Tversky's Parameterized Similarity Ratio Model: A Basis for Semantic Relatedness.** *Fuzzy Information Processing Society, 2006. NAFIPS 2006. Annual meeting of the North American* :541-546. 3–6 June 2006
12. Torsello A, Hidovic D, Pelillo M: **Four Metrics for Efficiently Comparing Attributed Trees.** *Proc of 17th International Conference on Pattern Recognition* 2004, **2:**467-470.
13. 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.
14. Wang JZZ, Du Z, Payattakool R, Yu PSS, Chen CFF: **A New Method to Measure the Semantic Similarity of GO Terms.** *Bioinformatics* 2007.
15. Pesquita C, Faria D, Bastos H, Falcao A, Couto F: **Evaluating GO-based Semantic Similarity Measures.** *BioOntologies SIG at ISMB/ECCB – 15th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB)* 2007.
16. Veale N , Seco JHT: **An Intrinsic Information Content Metric for Semantic Similarity in WordNet.** *ECAI 2004* 2004:1089-1090.
17. Schlicker A, Albrecht M: **FunSimMat: a comprehensive functional similarity database.** *Nucl Acids Res* 2007. gkm806+
18. Rada R, Mili H, Bicknell E, Bletner M: **Development and Application of a Metric on Semantic Nets.** *IEEE Transactions on Systems, Man, and Cybernetics* 1989, **19:**17-30.
19. Lee JH, Kim MH, Lee YJ: **Information Retrieval Based on Conceptual Distance in IS-A Hierarchies.** *Journal of Documentation* 1993, **49:**188-207.
20. Wu Z, Palmer M: **Verb semantics and lexical selection.** In *32nd. Annual Meeting of the Association for Computational Linguistics* New Mexico State University, Las Cruces, New Mexico; 1994:133-138.
21. Resnik P: **Using Information Content to Evaluate Semantic Similarity in a Taxonomy.** *Proceedings of IJCAI-95* 1995.
22. Resnik P: **Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language.** *Journal of Artificial Intelligence Research* 1999, **11:**95-130.
23. Jiang J, Conrath D: **Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy.** *Proc Int'l Conf Research in Computational Linguistics, ROCLING X* 1997.
24. Simon P: *Parts: a study in ontology* Oxford: Clarendon Press; 1987.
25. Gene Ontology Consortium: *GO Editorial Style Guide* 2004 [http://www.geneontology.org/GO.usage.html].
26. Bittner T: **Axioms for parthood and containment relations in bio-ontologies.** *Unknown* 2004.
27. Cherry JM, Adler C, Ball C, Chervitz SA, Dwight SS, Hester ET, Jia Y, Juvik G, Roe T, Schroeder M, Weng S, Botstein D: **SGD: Saccharomyces Genome Database.** *Nucleic Acids Res* 1998, **26:**73-79.
28. Camon E, Magrane M, Barrell D, Lee V, Dimmer E, Maslen J, Binns D, Harte N, Lopez R, Apweiler R: **The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology.** *Nucleic Acids Research* 2004:D262-D266.