simona: a Comprehensive R package for Semantic Similarity Analysis on Bio-Ontologies

Zuguang Gu[1,*]

[1]Molecular Precision Oncology Program, National Center for Tumor Diseases (NCT), Im Neuenheimer Feld 280, 69120 Heidelberg, Germany

*Corresponding author

Email address:
ZG: z.gu@dkfz.de

Abstract

Bio-ontologies are keys in structuring complex biological information for effective data integration and knowledge representation. In this paper, we present *simona*, a novel R package for semantic similarity analysis on general bio-ontolgies. *Simona* implements infrastructures for ontology analysis by offering efficient data structures, fast ontology traversal methods, and elegant visualizations. Moreover, it provides a robust toolbox supporting over 70 methods for semantic similarity analysis. With *simona*, we conduct a benchmark against current semantic similarity methods. The results demonstrate methods are clustered based on their mathematical methodologies, providing guidance for researchers in the selection of appropriate methods. Additionally, we explore annotation-based versus topology-based methods, revealing that semantic similarities solely based on ontology topology can efficiently reveal semantic similarity structures, facilitating analysis on less-studied organisms and other ontologies. *Simona* is freely available from https://bioconductor.org/packages/simona/.

Keywords: Bio-ontology, Semantic similarity, R, Bioconductor.

Background

Bio-ontologies play a crucial role in organizing and standardizing biological information to facilitate effective data integration and knowledge representation. They are typically constructed as directed acyclic graphs (DAGs) where biological concepts are modeled as terms connected by hierarchical relations. The most widely used bio-ontology, Gene Ontology (GO), provides a controlled vocabulary to describe functions, processes and cellular components of genes, playing a central role in functional interpretation in biology studies. Besides GO, there are a huge amount of bio-ontology resources developed in various biological domains, such as Medical Subject Headings (MeSH), Cell Ontology [1] and the Human Disease Ontology [2]. Additionally, there are bio-ontology databases that allow querying hierarchical relations of biology terms, such as the OBO Foundry [3], BioPortal [4] and Ontobee [5]. These resources are essential for data integration, enabling researchers to combine information from diverse sources and analyze it in a coherent and standardized way.

Semantic similarity analysis on bio-ontologies is a computational approach that quantitatively assesses the degree of similarity between biological concepts based on the semantics encoded in ontologies. It has wide applications in the biology field, such as gene function prediction [6], clustering and summarization of biological entities [7], interpretation of protein-protein interactions [8], cross-species comparisons [9], and biomedical text mining [10]. As biomedical data continues to increase dramatically in complexity and size, semantic similarity analysis is becoming an important tool for structured and meaningful interpretations and integration of complex data from multiple biological domains.

Semantic similarity analysis is a well-established topic in general ontology studies. Over the last three decades, numerous semantic similarity methods have been proposed. The development of these methods began with the analysis of WordNet [11], a collection of English words organized in a hierarchical structure. As the GO project got developed and matured, methods were adapted to apply to GO, and new methods specifically for GO were further developed. A comprehensive overview of state-of-the-art methods for semantic similarity analysis was provided in [12]. However, to date, only a few methods have been implemented as tools in the biology field, and most of them are only designed for GO. In the R programming ecosystem, there are currently only three packages for semantic similarity analysis: *ontologySimilarity* [13], *GOSemSim* [14] and *GOSim* [15]. These packages are either restricted to GO or have very limited functionalities. Therefore, the current tool development for semantic similarity analysis does not fully leverage the theoretical efforts in this field, limiting practical applications in general bio-ontology analysis.

In this paper, we introduced a new R package called *simona*, designed to be a general-purpose package for semantic similarity analysis on bio-ontologies. *Simona* makes two major contributions: 1) the establishment of an infrastructure for ontology analysis, including an efficient data structure for storing ontology data, fast methods to traverse ontologies, and elegant visualisations; 2) providing a comprehensive toolbox for semantic similarity analysis with more than 70 different methods. *Simona* is implemented by efficient algorithms, and has runtime improvement of approximately 2x, 25x, and >3000x compared to *ontologySimilarity*, *GOSemSim*, and *GOSim*, respectively, using GO as the test ontology.

Despite the comprehensive set of semantic similarity methods proposed in the field, there is a lack of quantitative comparisons on methods based on real-world datasets in current studies. We performed comparisons of semantic similarity methods, leveraging the comprehensive toolbox provided by *simona*. We grouped methods into clusters, which are reflected by commonness of their mathematical methodologies, providing users guidance to select a proper one for their studies. Most existing tools for semantic similarity analysis rely on external gene annotations and are majorly restricted to GO, limiting their applicability for organisms with incomplete gene annotation or on other ontologies. We compared the annotation-based method and the method solely based on the topology of the ontology without relying on external data. The results demonstrated that the two methods generate very similar similarity patterns. This analysis provides evidence to extend the similarity analysis to broader ontologies where gene annotations may not be available or may be incomplete.

Overview of the semantic similarity methods

Mazandu et al. [12] provides an excellent and comprehensive overview of current methods for semantic similarity analysis on GO. Nevertheless, these methods can be applied to general ontologies without modification. In *simona*, we have implemented the methods reviewed in [12] and an additional semantic similarity methods proposed by [16]. Furthermore, we introduced several new methods with simplified forms, primarily for comparing with other complex methods to reveal their mathematical attributes. In addition to the standard settings introduced in [12] and original papers, we support more flexible configurations. One typical scenario is for all distance-based methods, we support not only shortest distance-based methods as introduced in the original papers, but also longest distance-based methods. This is because distance is typically a measure of the semantic specificity that a term has or the distinctiveness that two terms exhibit in the ontology.

The semantic similarity analysis can be applied on the following three levels:

- *Single-term level*. At the level of individual terms, a numeric score is assigned to each term in the ontology, measuring its semantic specificity. This measure is often referred to as information content (IC). Generally, the information content is higher for terms deeper in the DAG, indicating a greater level of semantic information.
- *Term-pair level*. A similarity is calculated for a pair of terms. This score reflects how closely related two terms are in the ontology, determined by the specificity of their common ancestors or their proximity in the DAG.
- *Group-pair level*. Semantic similarity is also calculated for pairs of term groups. This calculation aggregates similarities from individual terms within the two groups. The resulting score provides a measure of overall similarity between the groups.

In the following three subsections, we only provide brief introductions to the methods. Readers are encouraged to refer to [12] or the vignettes of *simona* for the comprehensive definitions of the

methods. We adopt similar mathematical notations as presented in [12], available in its supplementary file.

In total, *simona* provides 11 methods for information content, 34 methods for term similarity, and 28 methods for group similarity. Note that for many similarity methods, users have the flexibility to choose a specific IC method. Additionally, for a group similarity method, users can select both a specific IC method and a term similarity method. Thus, *simona* provides extensive support for a wide range of methods in semantic similarity analysis.

Information content

Information content (IC) is a general concept that quantifies the amount of information a term can provide. In information theory, IC is defined as the negative logarithm of the probability of a specific occurring event. Formally, for term x,

$$\mathrm{IC}(x) = -\log(p(x))$$

where $p(x)$ is the probability of observing term x if taken as a random variable. In this way, when a term is associated with a small probability, indicating that the corresponding event is rare to occur, it is called informative if it is observed in a dataset.

There are a large number of methods that define $p(x)$, categorized as either extrinsic, relying on data from external sources, or intrinsic, which is solely dependent on the ontology itself. In many applications of semantic analysis on GO, $p(x)$ is often derived from external gene annotations to GO terms, which is defined as

$$p(x) = n(x)/N$$

where $n(x)$ is the number of unique genes annotated to GO term x, and N is the total number of genes annotated to the whole GO. Due to the hierarchical structure of GO, when a gene is annotated to a specific term, it is also implicitly annotated to all its parent and ancestor terms. Thus, $n(x)$ also includes genes that are indirectly annotated to the offspring terms of x. Then, $p(x)$ is the probability of a term being annotated if randomly selecting genes from N. It is easy to see, if a term is annotated to fewer genes, indicating a more specific biological concept, it becomes more informative for downstream biological meaning interpretation.

A simple example of the intrinsic definition of $p(x)$, which only relies on the ontology's topology, can be the fraction of terms that are offspring of x to the total number of terms in the ontology, defined as

$$p(x) = \frac{|D_x^+|}{|D_r^+|}$$

where D^+_x is the set of offspring terms of x, including x itself, D^+_r is the set of offspring terms of the root term r, including r itself (assuming there is only one root term for simplicity), and $|A|$ is the number of elements in set A. This simple definition of $p(x)$ measures the broadness of a term being connected to the rest of terms in the ontology (we name this method as IC_offspring).

Simona implements another intrinsic $p(x)$ which measures the relative distance to leaf terms in the ontology, defined as:

$$p(x) = \frac{\eta(x)}{\delta(x) + \eta(x)}$$

where $\eta(x)$ is the height of x, *i.e.*, the longest distance from x to leaves, and $\delta(x)$ is the depth of x, i.e., the longest distance from the root to x. Then, $p(x)$ is the proportion of the length from x to the leaf in the longest path from the root to the leaf, passing through x (we name this method as IC_height).

Not necessarily restricted to probability-based forms, IC can be generalized as a numeric score, satisfying the only requirement: $\mathrm{IC(parent)} \leq \mathrm{IC(child)}$, which is that a parent term should not be more specific than its child term.

Term similarity

The calculation of semantic similarity between two terms relies on their topological relations within the ontology. Simply speaking, if the common ancestors of two terms represent highly specific biological concepts, or if the two terms are closely located in the DAG, it can be concluded that the two terms are similar in contributing similar biological meanings. The evaluation of semantic similarity always involves common ancestors (CAs) shared by the two terms, which are denoted as:

$$\mathrm{CA}(a, b) = A_a^+ \cap A_b^+$$

where A^+_a is the set of term a's ancestors, including a itself. The notation is the same for term b.

A simple definition of the semantic similarity implemented in *simona* is:

$$\mathrm{Sim}(a, b) = \frac{\left|A_a^+ \cap A_b^+\right|}{\left|A_a^+ \cup A_b^+\right|}$$

which is a Jaccard-like measurement. Terms a and b have high similarity if they have similar inheritance from the upstream of the ontology. Despite its simplicity, the Results section demonstrates that it performs similarly to more sophisticated methods (we name this method as Sim_Ancestor).

Many similarity methods take use of a specific common ancestor. There are the following three different types of common ancestors.

- *Most informative common ancestor (MICA)*. The common ancestor term with the highest IC is named MICA. ICs can be calculated extrinsically or intrinsically.
$$\mathrm{MICA}(a, b) = \underset{t \in \mathrm{CA}(a,b)}{\arg\max}\, \mathrm{IC}(t)$$
- *Lowest common ancestor (LCA)*. The common ancestor term with the largest depth (i.e., lowest in the DAG) is named LCA.
$$\mathrm{LCA}(a, b) = \underset{t \in \mathrm{CA}(a,b)}{\arg\max}\, \delta(t)$$
- *Nearest common ancestor (NCA).* Some methods depend on the shortest distance between term a and b. The common ancestor of a and b on the shortest path is named NCA. In the following equation, $D_{\mathrm{sp}}(t, a)$ is the shortest distance from term t to a, and the same for $D_{\mathrm{sp}}(t, b)$.
$$\mathrm{NCA}(x) = \underset{t \in \mathrm{CA}(a,b)}{\arg\min}(D_{\mathrm{sp}}(t, a) + D_{\mathrm{sp}}(t, b))$$

The next two semantic similarity methods are widely used in ontology analysis. Lin [17] defined the similarity as the IC of the MICA term c, normalized by the average ICs of term a and b.

$$\mathrm{Sim}_{\mathrm{Lin}}(a, b) = \frac{2 * \mathrm{IC}(c)}{\mathrm{IC}(a) + \mathrm{IC}(b)}.$$

Although originally proposed for general information theory, it has been adopted and widely used for semantic similarities analysis on GO terms, taking ICs calculated from gene annotations.

As a second typical example, Wu and Palmer [18] proposed a similarity method that is purely based on the ontology structure, defined as:

$$\mathrm{Sim}_{\mathrm{WP}}(a, b) = \frac{2 * \delta(c)}{\mathrm{len}(c, a) + \mathrm{len}(c, b) + 2 * \delta(c)}$$

where term c is the LCA of term a and b, $\mathrm{len}(c, a)$ is the longest distance from c to a, and $\delta(c)$ is the depth of c which is the longest distance from root r to c. $\delta(c)$ can be written as $\mathrm{len}(r,c)$, then the WP method can be written in the Lin-like form:

$$\mathrm{Sim}_{\mathrm{WP}}(a, b) = \frac{2 * \mathrm{len}(r, c)}{\mathrm{len}_c(r, a) + \mathrm{len}_c(r, b)}$$

where $\mathrm{len}_c(\mathrm{r}, a)$ is the longest distance from r to a, passing through term c.

In the Results section, we will use the Lin and the WP methods as two representative approaches for comparison.

Group similarity

It calculates a single similarity value between two groups of terms, by aggregating similarities from individual term pairs within the two groups. One typical application is in calculating the semantic similarity of two genes or proteins, where the similarity is calculated based on the two sets of GO terms to which genes are annotated. Denote the two groups of terms as T_p and T_q, in the "best-match average" (BMA) method [19], the semantic similarity of a single term x to a group of terms T is first calculated:

$$S(x, T) = \max_{y \in T} S(x, y)$$

which is the similarity between x and its "best matched" term in T. Based on it, the semantic similarity between group p and q can be calculated, for example, as the average of similarities between terms in one group and their best matched terms in the other group:

$$\mathrm{GroupSim}(p, q) = \frac{1}{2}\left(\frac{1}{|T_p|}\sum_{a \in T_p} S(a, T_q) + \frac{1}{|T_q|}\sum_{b \in T_q} S(b, T_p)\right).$$

The second type of methods treats the two groups of terms as two sub-graphs induced from the complete ontology DAG. For example, the following method calculates the average distance between the two sub-graphs as the similarity measure [20].

$$\mathrm{GroupSim}(p, q) = \exp\left(-\frac{1}{|T_p| * |T_q|}\sum_{a \in T_p, b \in T_q} D_{\mathrm{sp}}(a, b)\right)$$

Simona supports a large number of group similarity methods. However, due to the heterogeneity of similarity patterns within groups of terms, the similarity value on the group-level may vary significantly across different methods. Thus, in the Results section, we will not perform benchmarking on group similarity methods.

Implementation

The package

Simona is a general-purpose package for semantic similarity analysis on bio-ontologies.It employs a starndardized data structure in the *ontology_DAG* class and implements efficient functions for traversing DAGs using depth-first search (DFS) or breadth-first search (BFS). *Simona* can efficiently handle large ontologies containing millions of terms within a reasonable time period.

It is also useful to examine local structures of DAGs, only focusing on a subset of terms of interest. For example, it can be useful to study how information is transmitted from upstream to a specific set

of downstream terms. *Simona* provides a flexible way to filter the complete DAG. Specifically, to obtain a sub-DAG induced from a single term as the root or a set of terms as leaves, the subsetting method on the complete `dag` object is especially convenient, as demonstrated in the following code.

```
dag[term] # a sub-DAG that only contains offspring of `term`
dag[, terms] # a sub-DAG that only contains upstream of `terms`
```

In the semantic similarity analysis, for two terms, `term1` and `term2`, the following code can be used to extract a sub-DAG containing their common ancestors.

```
ancestors1 = dag_ancestors(dag, term1, include_self = TRUE)
ancestors2 = dag_ancestors(dag, term2, include_self = TRUE)
sub_dag = dag_filter(dag, terms = intersect(ancestors2, ancestors2))
```

While to take the union of ancestors of both terms, we can directly use the subsetting methods.

```
dag[, c(term1, term2)]
```

Simona offers a comprehensive set of low-level functions designed for term pairs. These functions include calculations for distances considering DAGs in both directed and undirected modes, supporting both shortest distance and longest distance measurements. Additionally, there are functions for identifying MICA/LCA/NCA terms. These low-level functions serve as the foundation for implementing more specialized methods for semantic similarity analysis in *simona*.

Ontology formats

Simona supports the common formats of ontologies, simplifying the analysis of biological ontologies from public databases. For the *obo* format, *Simona* provides a parser function, `import_obo()`, which parses and imports hierarchical relations of terms along with their relation types. Other formats, such as *owl* (the RDF/XML format) and *ttl* (the Turtle format), are parsed by an external parser, *ROBOT* [21]. Internally, they are converted to the *obo* format by *ROBOT* and then read by `import_obo()`. While *ROBOT* serves as an elegant tool for converting between various ontology formats, it may not be memory-efficient for larger ontologies (e.g., > 200k terms). To address this concern, *Simona* implements native parsers for the *owl* and *ttl* formats through the `import_owl()` and `import_ttl()` functions. These functions can recognize commonly-used tags in the files. However, they may not be universally applicable to all aspects of the two formats.

Treelization

DAG is a generalized form of a tree, allowing a child term to have more than one parent. *Simona* provides a function `dag_treelize()`, which simplifies a DAG to a tree to let each term strictly have one parent. The reduction is executed in a breadth-first manner. Beginning from root and for a term a, a unique link connects a and its child term c only when $\delta_c = \delta_a + 1$, *i.e.*, the depth of c equals the depth of a plus one. In this way, the depths of all terms in the reduced tree remain the same as in the original DAG. The resulting tree is also saved as an *ontology_DAG* object, but explicitly marked as a tree. Treelization is mainly used for partitioning and visualization on DAG, which will be introduced in the following two subsections.

Partitioning

In ontology, a biological concept may have multiple child concepts that describe it more specifically from distinct aspects. Partitioning the ontology helps achieve a more concise view of the system. In *simona*, ontology partitioning is applied to the treelized DAG, where the tree is cut at specific levels. There are two approaches: cutting the tree at a certain level (depth) using the `partition_by_level()` function, or dynamically cutting the tree by controlling the number of terms in sub-trees with the `partition_by_size()` function.

Visualization

Hierarchical layouts, such as radial layout, are widely used for ontology visualization [22]. While these layouts are efficient for visualizing small ontologies, challenges arise when dealing with larger ontologies. *Simona* provides functions for visualizating both small and large ontologies.

The `dag_graphviz()` function is designed to visualize small DAGs using the Graphviz hierarchical layout internally through the *DiagrammeR* package. Together with the subsetting functionality on the *ontology_DAG* object, it is a useful tool to study local structures in DAG. For example, to only visualize ancestors of two terms (an example can be found in Supplementary File 5):

```
sub_dag = dag[, c(term1, term2)]
dag_graphviz(sub_dag)
```

Simona also implements a radial layout for visualizing complete ontologies with the function `dag_circular_viz()`. The layout is applied to the treelized DAG, and additional links in the DAG are added subsequently. In this layout, terms are positioned on circles with different radius. For a circle where terms with depth of d are located, the radius is calculated as

$$d_{\max} * \log\left(\sum_{i}^{N} I(\delta_i \leq d)\right) / \log N$$

where $d_{\max}$ is the maximal depth, N is the total number of terms in the DAG, δ_i is the depth of term i and $I()$ denotes the identity function. The radius is set to zero for the root term. This adjustment on the radius helps get rid of the scenario where a single long branch affects the global visualization.

Each term is associated with a sector on the circle. The width of the sector is proportional to the number of its connectable leaf terms. Importantly, since the layout is from the tree representation of the DAG, the sectors of child terms are mutually exclusive. The sum of their widths equals the width of the sector of their parent term. In other words, the sector of a term entirely includes its offspring in the treelized DAG.

The `dag_circular_viz()` function provides a useful tool for visualizing the global structures of ontologies (examples are shown in Figure 5), thereby revealing specific attributes that may be of interest. Another practical use case is to emphasize a list of terms on the DAG to observe their proximity and relationships. For example, the following code highlights a list of significant GO terms from a GO enrichment analysis (Figure 1), where `sig_go_ids` is a vector containing GO terms.

```
dag_circular_viz(dag, highlight = sig_go_ids, …)
```

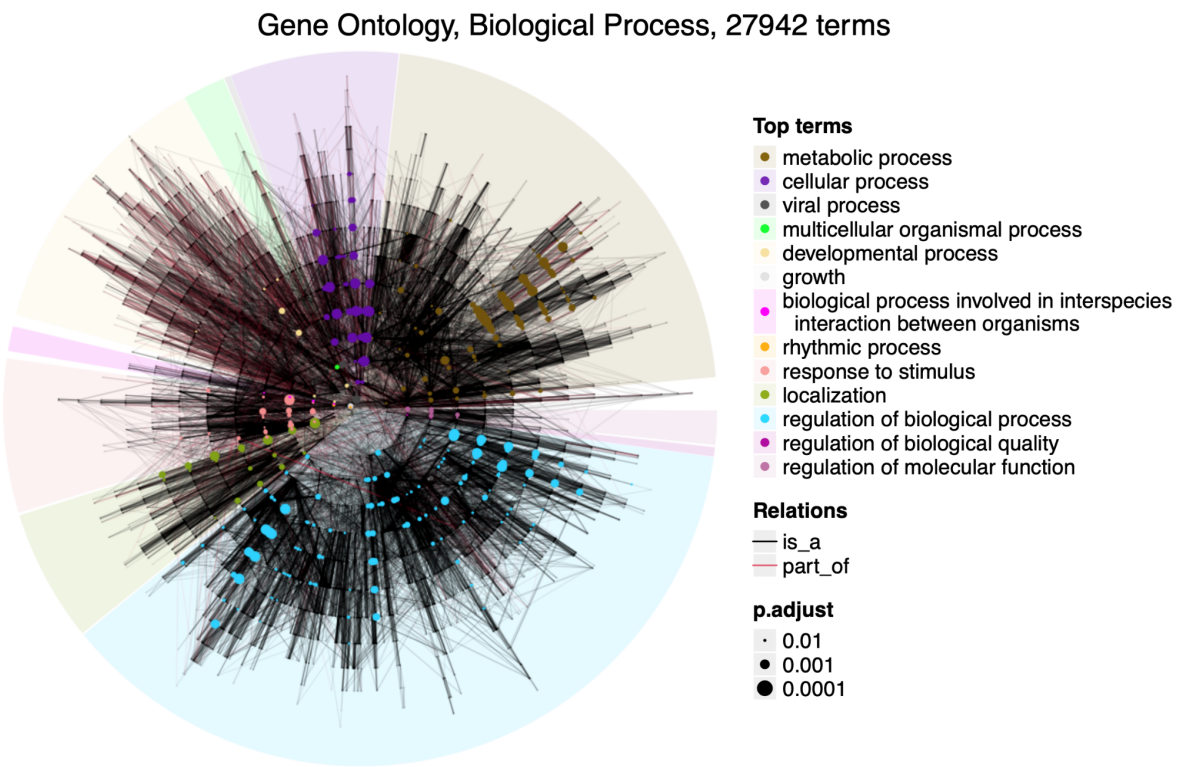


Figure 1. Circular visualization on GO, BP namespace. A list of significant GO terms from a GO enrichment analysis are highlighted on the plot. Dot sizes are proportional to -log10(p.adjust) from the enrichment analysis.

Semantic similarity analysis

Simona provides an extensive set of methods for semantic similarity analysis. The `term_IC()` function supports 11 methods for calculating ICs, `term_sim()` supports 34 methods for calculating term similarities, and `group_sim()` supports 28 methods for calculating group similarities. The core calculations are implemented through efficient algorithms in C++ code, and it has significant performance improvement over existing tools. Detailed description of the algorithm can be found in Supplementary File 2.

Results and discussion

Compare to other tools

We compare *simona* to the following R packages supporting semantic similarity analysis: *ontologyIndex*/*ontologySimilarity*, *GOSemSim* and *GOSim*. *ontologyIndex* and *ontologySimilarity* are part of a suite of packages called *ontologyX*, where *ontologyIndex* offers methods for IC analysis and *ontologySimilarity* offers methods for semantic similarity analysis. *GOSemSim* and *GOSim* only support GO, so for this comparison, we take the Biological Process (BP) namespace in GO as the test ontology. *GOSemSim* and *GOSim* only support IC calculated based on gene annotation. While *ontologyIndex* supports more general annotation-based IC, but when working with GO, it mostly uses genes as the annotation items. Therefore, we only compare IC and semantic similarities based on gene annotations, with human as the selected organism. In the comparison, the GO data is from the *GO.db* package (version 3.17.0) or the *obo* file from GO website with the same version (version 2023-01-01) as in *GO.db*. The relation types of “is a”, “part of”, “regulates”, “positively regulates” and

“negatively regulates” are used. The package *org.Hs.eg.db* (version 3.17.0) is used as the source of gene annotation to GO terms. A detailed analysis report can be found in Supplementary File 1.

We first compare the information contents calculated by the four tools. As shown in Figure 1A-C, *simona* and *ontologyIndex* generate identical IC values, while another two tools of *GOSemSim* and *GOSim* also generate identical IC values. Although there are systematic shifts in ICs between *simona* and *GOSemSim*, they show strong linear relationships.

The difference of ICs between *simona* and *GOSemSim* mainly comes from how they calculate the number of genes annotated to each GO term. Given the DAG structure of GO, when a GO term is annotated with a gene, all its ancestor terms are also associated with that gene. In this way, to reduce the data size, a gene is normally only annotated to the lowest terms in the GO annotation database, and the aggregation of annotated genes on GO terms is recursively applied. It is worth to note, however, there are also cases where a gene is annotated to both ancestor and offspring terms in the database file, potentially due to incomplete annotations.

simona and *ontologyIndex* use the count of unique genes that are directly annotated to the GO term, as well as to its offspring terms. Let’s denote the set of annotated genes for term x as G_x and it is calculated as:

$$G_x = \bigcup_{z \in \mathrm{D}_x^+} G_z^*$$

where D_x^+ is the set of x’s offspring terms, including x itself, and G_z^* is the set of genes *directly* annotated to term z. Then the information content is

$$\mathrm{IC}(x) = -\log\left(\frac{|G_x|}{|G_r|}\right)$$

where G_r is the set of total genes annotated to the root term r, i.e., to the complete ontology.

GOSemsim and *GOSim* have a different way to calculate the numbers of annotated genes. They simply add the numbers of genes directly annotated to term x and its offspring terms without removing duplicated genes. In the following equation

$$N_x = \sum_{z \in \mathrm{D}_x^+} N_z^*,$$

N_x is the number of genes annotated to x after aggregating from its offspring terms, N_z^* is the number of genes directly annotated to term z. The information content is then

$$\mathrm{IC}(x) = -\log\left(\frac{N_x}{N_r}\right)$$

where N_r is the sum of numbers of genes directly annotated to every GO term in the ontology.

It is common for a gene to be annotated to multiple GO terms (on average, a gene is annotated to 3.2 GO BP terms), which results that on the ancestors of these GO terms, this gene will be counted multiple times. Additionally, even for genes directly annotated to the same GO term, it is possible that the gene is duplicated but in different evidence codes (36.7% of GO BP terms are directly annotated to duplicated genes). These factors lead to an overestimation of the number of annotated genes on a GO term when simply adding the numbers from its offspring terms, and in turn, as shown in Figure 1B, it may overestimate ICs. The two methods yield identical results when a gene is only uniquely annotated to a distinct GO term in the ontology. Given that p in the IC definition is the probability of observing a GO term when a gene is randomly picked, the method by *simona* and *ontologyIndex* is more proper under this context.

The method of annotation-based IC [23] was originally proposed and applied to WordNet which is a taxonomy of concepts represented as nouns in English words, e.g., a coin is a subclass of cash. The probability of observing a concept is based on the frequency of that concept, as well as its offspring classes in a corpus gathered from a large collection of text. In that dataset, the frequency of observing the concept can be simply added from its offspring concepts because a concept is unique in the taxonomy. However, when applying this method to GO, adjustments need to be made because now the corpus consists of genes while concepts are GO terms. The mapping between the two is not one-to-one. Many studies adopt the default implementation without distinguishing duplicated gene annotations [24,25], while only a few studies explicitly take unique genes [26,27]. In Supplementary Figure S1.3, we demonstrate that *GOSemSim* relatively overestimates ICs more for terms with smaller depth, i.e., the terms closer to the root.

We next compare the semantic similarity calculated by the four tools, using the method proposed by Lin [17]. Since not all GO terms have gene annotation (57.1% of GO BP terms have genes annotated), we randomly sample 500 genes from the GO terms from BP namespace that have gene annotations. Figure 1D-G illustrate the similarity heatmaps of the 500 random GO terms calculated by the four tools. Once again, *simona* and *ontologySimilarity* produce identical similarity values, and *GOSemSim* and *GOSim* produce identical values. *GOSemSim* estimates systematically higher similarities than *simona*, due to its overestimation of ICs. *GOSemSim* relatively overestimates similarity more for the MICA terms with smaller depth (Supplementary Figure S1.6). Nevertheless, in general, similarities calculated by the four tools are highly similar (Supplementary Figure S1.5).

Last, we benchmark the runtime performance of the four tools. The key step in Lin's method is to traverse the DAG to look for MICA of every term pair. It is also the dominant and most time-consuming part in the computation. We test the runtime performance on calculating Lin's similarity with varying numbers of random terms, ranging from 200 to 15,000 in steps of 200, on the GO BP ontology. Due to *GOSim*'s bad runtime performance, we only run *GOSim* on no more than 1000 terms. Figure 1H clearly illustrates that *simona* and *ontologySimilarity* show significantly better performance than *GOSemSim* and *GOSim*. Among the four tools, *simona* has the best performance. For example, with 10,000 query terms, *simona* achieves 2x speedup compared to *ontologySimilarity* and 25x speedup compared to *GOSemSim* (Supplementary Figure S1.8).

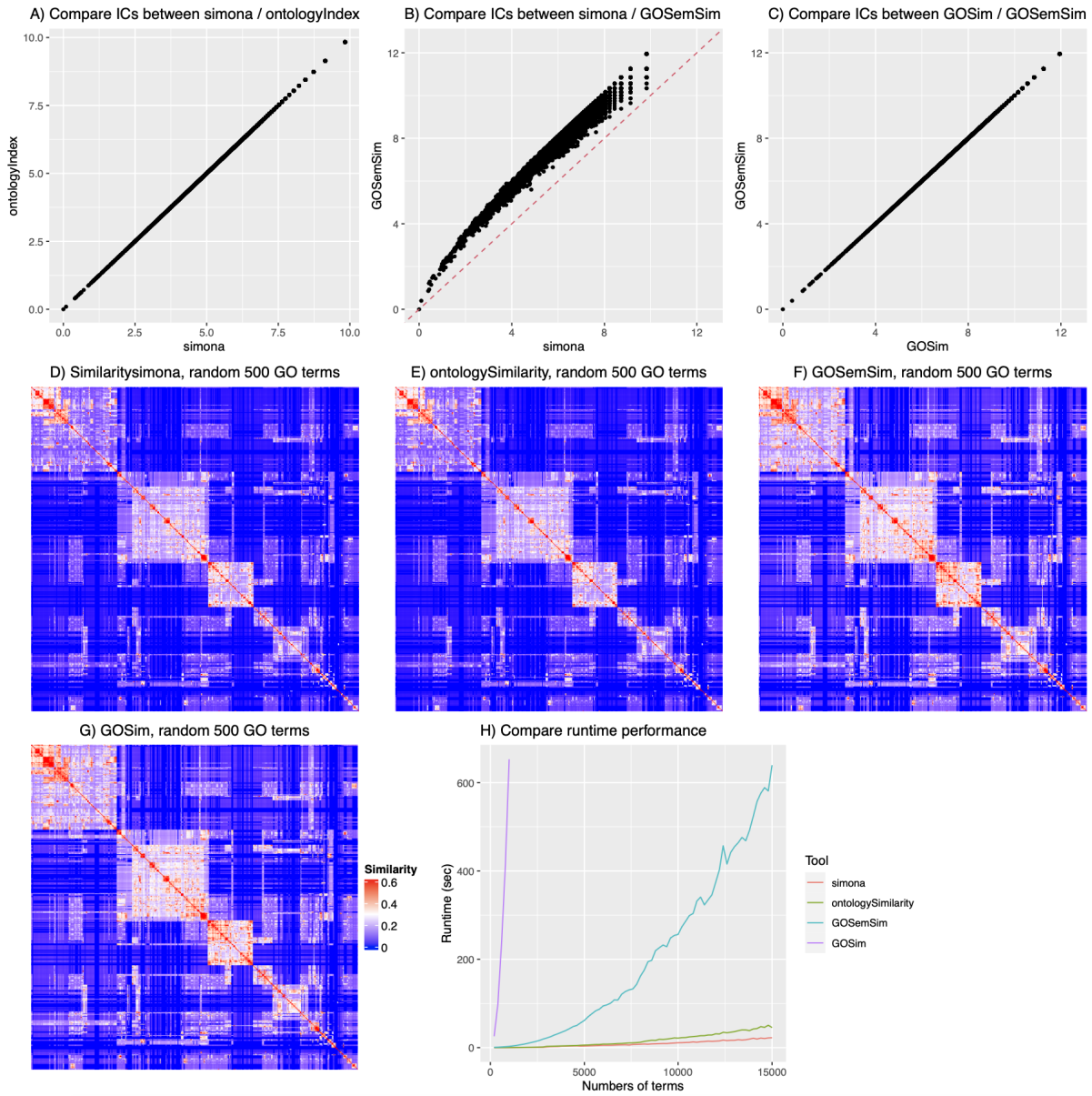


Figure 2. Compare *simona* to other tools. A-C) Pairwise comparisons of information contents of GO terms, calculated by *simona*, *ontologyIndex*, *GOSemSim* and *GOSim*. D-G) Heatmaps of semantic similarities of 500 random GO terms, using the Lin method implemented in the four tools. H) Runtime performance on the number of query GO terms.

Compare IC and semantic similarity methods

Mazandu et al. [12] have provided a comprehensive and theoretical overview of the state-of-the-art methods for semantic similarity analysis. However, a quantitative benchmark of IC and semantic similarity methods based on real-world data is still lacking in current studies. The comprehensive toolkit that *simona* provides makes it possible to perform comparisons of different methods on specific ontologies. In this section, we compare both IC and semantic similarity methods implemented in *simona*. For the benchmark, we take the Biological Process (BP) namespace in GO as the test ontology, using relation types of “is a” and “part of”. The gene annotations to GO terms are based on human as the organism. Readers are encouraged to refer to the vignettes of *simona* for the comprehensive definitions of the methods mentioned in this section.

ICs for all GO terms are calculated using 11 IC methods implemented in *simona*. Figure 3A illustrates the heatmap of the Pearson correlation of ICs from different IC methods. Partitioning on rows and columns of heatmap is based on hierarchical clustering on the dissimilarity matrix of $1 - cor$ where cor is the correlation matrix of ICs (Supplementary Figure S3.1). In Figure 3A, the two methods of IC_Wang and IC_universal are far away from other methods, which is also confirmed in the multidimensional scaling (MDS) analysis in Figure 3B. The remaining IC methods can be classified into two groups. It is interesting to note that, for methods in group 1, the heights or the depths of terms in the DAG play a major role in calculating IC values, while for methods in group 2, the aggregation of all offspring terms plays the major role. Supplementary Figure S3.3 illustrates pairwise scatterplots of ICs from every two IC methods, which helps to inspect the trend of the correlation and how term depths are weighted in different IC methods.

We next compare term similarity methods. Semantic similarities of 124,750 term pairs, derived from 500 random GO BP terms with gene annotations, are calculated using 27 similarity methods implemented in *simona*. Seven out of total 34 similarity methods are removed from the comparison due to large differences from all other methods (the complete analysis can be found in Supplementary Figure S4.1). In this comparison, default settings are used for all similarity methods. Figure 3C illustrates the heatmap of the Pearson correlation of semantic similarities from 27 methods. Methods are grouped into 5 clusters based on hierarchical clustering on $1-cor$ where cor is the correlation matrix of semantic similarities (Supplementary Figure S4.2). The grouping of methods can be confirmed on the MDS plot in Figure 3D.

It is also interesting to note that the grouping of similarity methods reflects their methodologies. Group 1 mainly contains methods utilizing ICs from gene annotations. Methods in group 2 are heterogeneous where their mathematical forms are quite different. Methods in group 3 mainly relate to depths of LCA terms. Methods in group 4 relate to aggregation from all common ancestors, specifically the fraction of aggregation from the intersection of common ancestors to the aggregation from the union of common ancestors. Group 5 contains methods more related to distances of two terms in the DAG. Supplementary Figure S4.4 illustrates pairwise scatterplots of similarity values from every two term similarity methods, helping to inspect the trend of the correlation and how the depths of MICA/LCA terms are weighted in different similarity methods. Supplementary Figure S4.5 illustrates similarity heatmaps of the 500 random GO terms from every similarity method for direct comparisons.

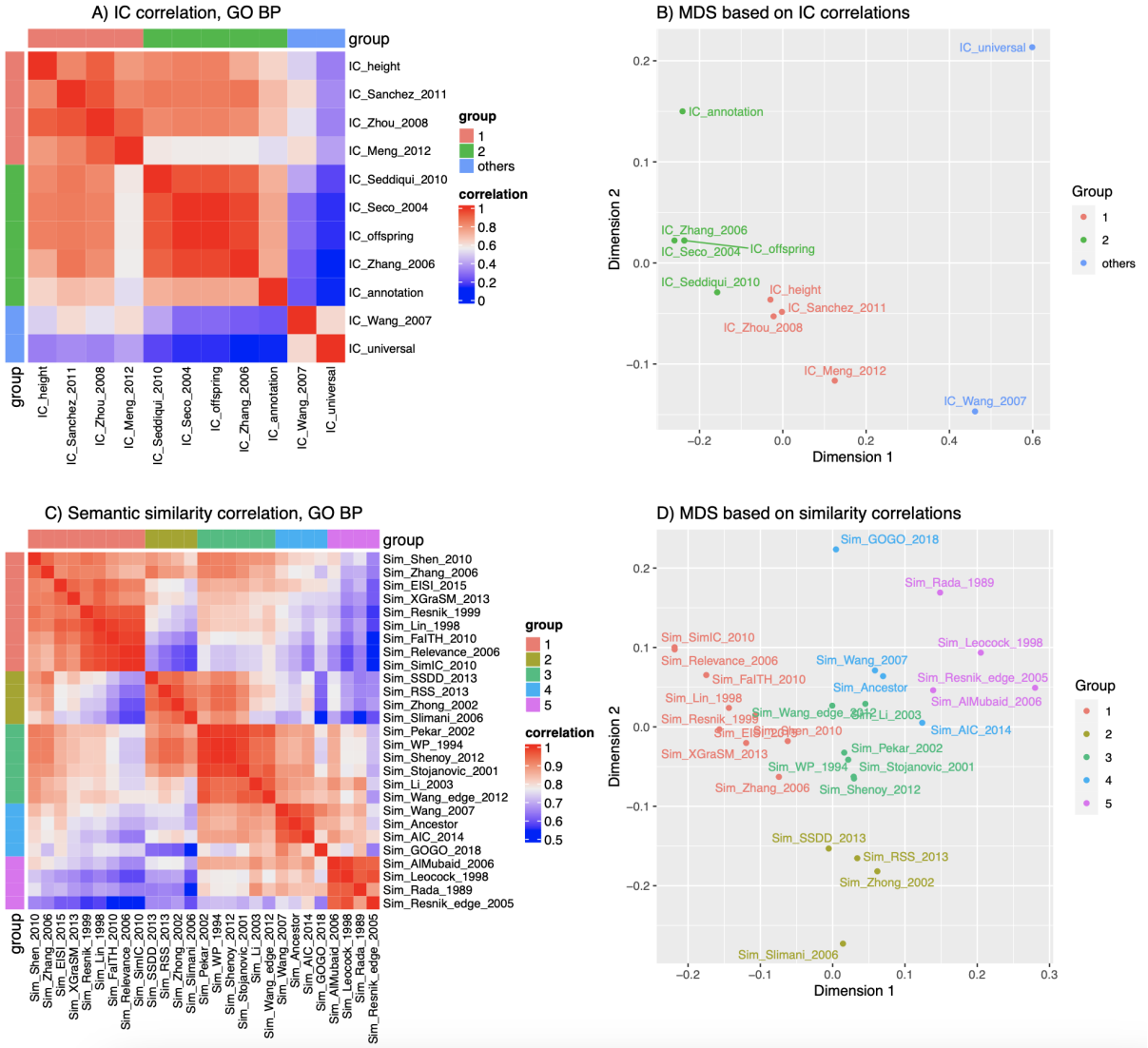


Figure 3. Compare IC and term similarity methods. A) A heatmap of correlations of IC values from different methods. B) The MDS on the correlation matrix in Figure A. C) A heatmap of correlations of semantic similarity values from different methods. D) The MDS plot on the correlation matrix in Figure C. In Figure A and C, partitionings on rows and columns are based on hierarchical clustering on the corresponding matrix.

Compare annotation-based and topology-based methods

Existing R packages as well as other implementations (e.g., [28] and [29]) for semantic similarity analysis are only based on ICs calculated from gene annotations. Therefore, they are primarily applied to GO only with well-annotated organisms such as human and mouse. In addition to annotation-based methods, *simona* supports a diverse range of methods based on the topology of the GO DAG, with no need for external annotation data. Here, we compare semantic similarity calculated based on gene annotations with that based on the topology of the GO DAG. For the benchmark, we take the Biological Process (BP) namespace in GO as the test ontology, using relation types of “is a” and “part of”. The gene annotations to GO terms are based on human as the organism.

For the IC-based method (referred to here as annotation-based, denoted as “IC_annotation” for clarity), the semantic similarity is determined by the IC of the MICA of the pair of terms, using the Lin method; while for the topology-based method, the semantic similarity is determined by the depth of the LCA, using the WP method. As introduced in the Methods section, the two methods have very

similar forms. We compare, for a given set of 500 random GO terms resulting in 124,750 term pairs, how different their MICA terms and LCA terms are.

Figure 4A indicates that for more than 98% of term pairs, their MICA and LCA terms are the same. Among the remaining 2% (2287 term pairs), we observe that none of their MICA and LCA terms have ancestor or offspring relations (Supplementary File 5). However, Figure 4B reveals that for 76.3% of these 2287 term pairs, their MICA and LCA terms are siblings, meaning they share the same parents, with an undirected distance of 2. Similar close relationships between MICA and LCA terms were also observed for other term pairs (Figure 4B. An example can be found in Supplementary Figure S5.1).

Note that the depth of a parent is always smaller than the depth of its child terms because the depth is defined as the longest distance from the root. This satisfies the requirement of the general IC definition: $\mathrm{IC(parent)} \leq \mathrm{IC(child)}$. Under this context, depth can also be considered as a general definition of IC. If we also consider that IC_annotation is calculated based on the aggregation of gene annotations from offspring terms, which actually also takes into account the DAG topology, we would expect the depth and IC_annotaiton to show a strong positive correlation (Supplementary Figure S5.3).

As depth can be thought of as an IC measure, the LCA is actually a special type of MICA. According to the high correlation between IC_annotation and depth, as well as the large overlap between MICA and LCA terms, we would expect that the semantic similarities based on MICA and LCA should also be highly similar. In Figure 4C and 4D, the two heatmaps illustrate that the similarity patterns from the Lin method and the WP method are highly similar, where the WP method generates a higher-level of similarity but also increases the inter-block signals off diagonals. Supplementary Figure S5.5 illustrates that the two types of similarities exhibit strong positive correlations.

Most current analyses on semantic similarities rely on gene annotations, which requires high quality and complete annotation data. Unfortunately, only a very small number of ontologies, such as GO, have well-annotated data, and they are also limited to well-studied organisms. This significantly restricts the use of semantic similarity analysis on other less-studied organisms and makes it difficult to extend to other ontologies. The results presented in this section deliver a clear message that methods solely based on the topology of the ontology can also efficiently reveal the structure of semantic similarity. Together with the wide range of topology-based methods implemented in *simona*, it will greatly extend the semantic analysis on other organisms and ontologies.

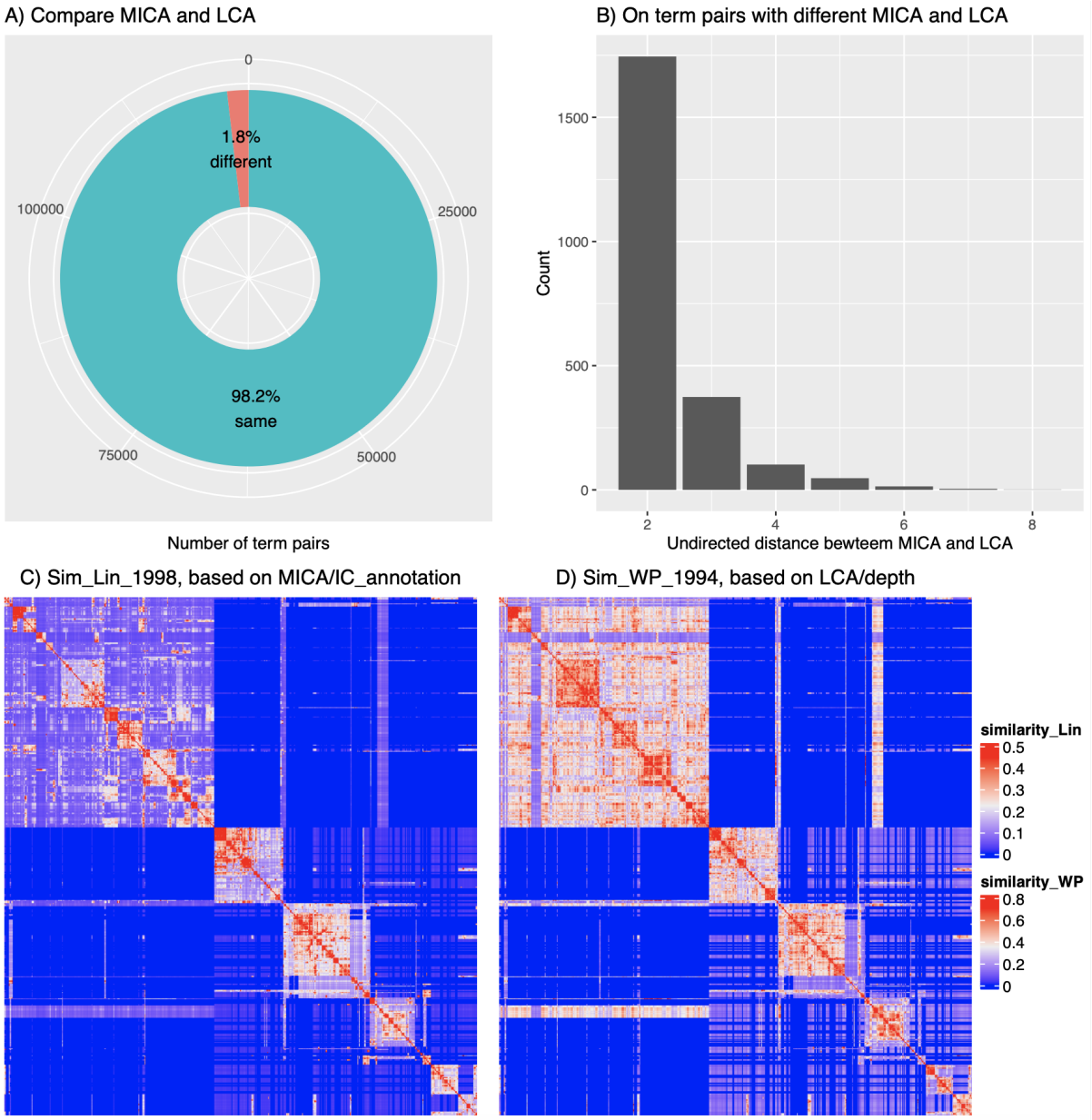


Figure 4. Compare the MICA and LCA of 500 random GO terms. A) Numbers and proportions of term pairs with identical or distinct terms as their MICA and LCA. B) Distribution of the undirected distance between MICA and LCA for term pairs with differing MICA and LCA. C) A heatmap of semantic similarity using the Lin method. D) A heatmap of semantic similarity using the WP method. Row orders and column orders are the same between Figure C and D.

A gallery of OBO Foundry ontologies

The OBO Foundry [3] serves as a repository for public biological ontologies. We import 206 ontologies from the OBO Foundry (data retrieved on 2023-08-07) and generate global circular visualization, similarity heatmaps and runtime performance analysis for each ontology. The complete gallery is available in Supplementary File 6.

One use of the OBO Foundry gallery is to explore ontology-specific global structures. Figure 5 illustrates examples of four ontologies with different structures where each ontology is partitioned by automatically choosing a proper number of clusters.

In Figure 5A, we visualize the NCBI Taxonomy [30] which contains over 2.5 million terms. NCBI taxonomy has a strict tree structure, with leaf terms as organisms and non-leaf terms indicating taxon classifications on specific levels. The widths of sectors correspond to the numbers of organisms in sub-classifications. For example, the largest cluster of organisms is under the taxon Endopterygota (a subset of insects, the deep orange cluster on the bottom left) and the term Endopterygota is comparably low in the complete taxonomy. If the depth of the taxonomy represents the extent of classification, we can find taxa of Viruses, Bacteria and Fungi (two green clusters on the right and one red cluster on the top) have small depths, while the taxon led by Deuterostomia (a subset of animals, yellow cluster on the bottom right), although only includes a small fraction of organisms in the taxonomy, extends very deep, indicating a well-studies group.

Figure 5B illustrates the Chemical Entities of Biological Interest [31], a hierarchical organization of approximately 165,000 small chemical compounds. Being different from other ontologies in Figure 5, it is more densely connected. The average number of parents is 1.4 which is also among top ontologies in the OBO Foundry. Chemical compounds may be involved in multiple types of chemical reactions, thus they are associated with more than one parent chemical function.

Figure 5C illustrates the Ontology of Genes and Genomes (OGG) [32]. It has a very different structure characterized by huge numbers of leaf terms connecting to parents. This structure reflects the construction of OGG. The upper levels contain hierarchical relationships of gene types, e.g., “protein-coding gene of Homo sapiens” being a subclass of “gene of Homo sapiens”, while at the lowest level, specific genes are assigned to the most specific gene types within corresponding branches, such as protein-coding genes, pseudogenes, RNA genes, etc.

Finally, Figure 5D visualizes the Vaccine Ontology [33]. Despite its relatively small size, it exhibits a well-defined hierarchical structure, closely resembling a tree, with only 1.7% additional links from the treelized DAG.

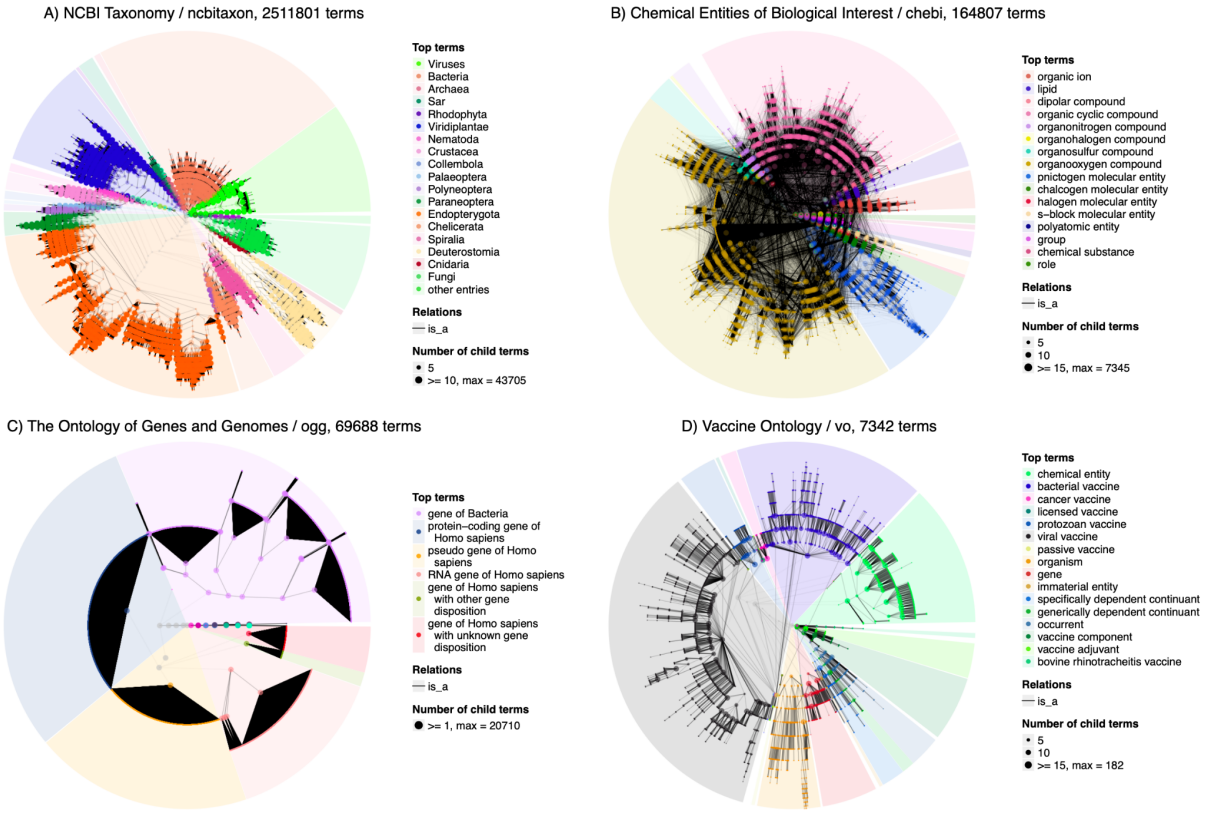


Figure 5. Circular visualization of four bio-ontologies. A) NCBI Taxonomy. B) Chemical Entities of Biological Interest. C) The Ontology of Genes and Genomes, D) Vaccine Ontology.

Conclusion

Simona offers a versatile interface and efficient implementation for processing, visualization, and semantic similarity analysis on bio-ontologies. We believe that *simona* will serve as a robust tool for uncovering relationships and enhancing the interoperability of biological knowledge systems.

Abbreviations

GO: Gene Ontology; BP: biological process; DAG: directed acyclic graph; MICA: most informative common ancestor; LCA: lowest common ancestor; NCA: nearest common ancestor; CA: common ancestor; IC: information content; BFS: breadth-first search; DFS: depth-first search; MDS: multidimensional scaling.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and materials

The *simona* package as well as its documentation are freely available from https://bioconductor.org/packages/simona/. Supplementary files are also available from https://jokergoo.github.io/simona_supplementary/.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

ZG implemented the software, performed the analyses and wrote the manuscript. All authors read and approved the final manuscript.

Acknowledgements

Not applicable.

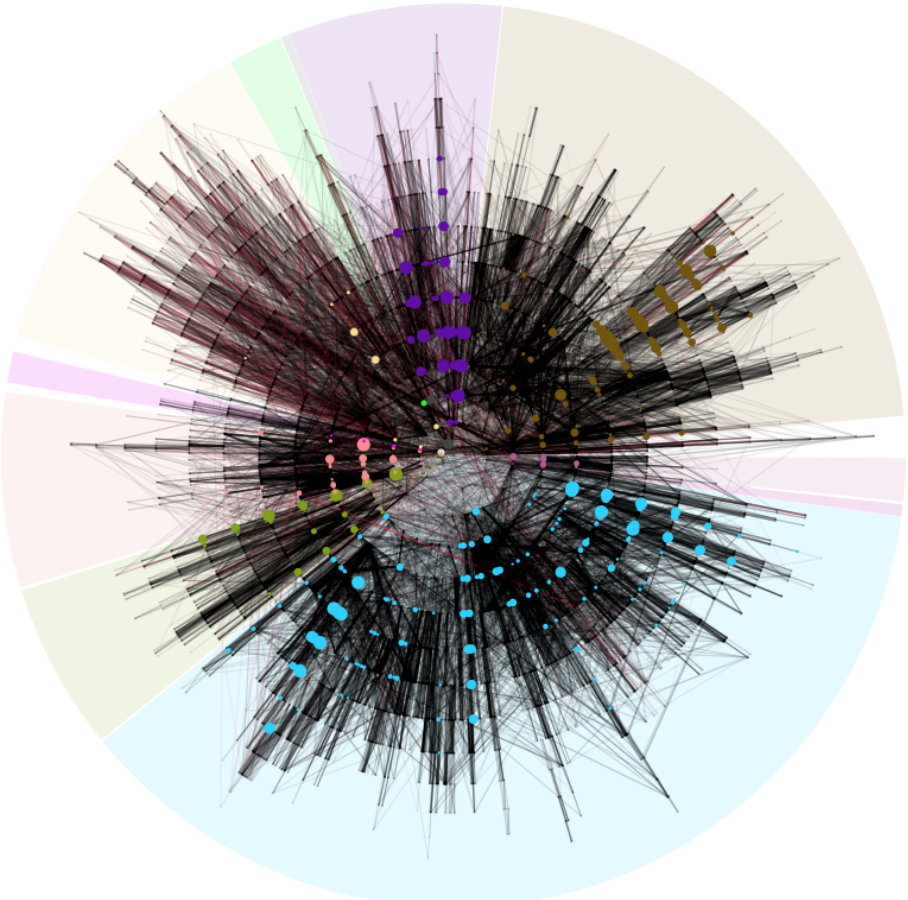
References

1. Osumi-Sutherland D, Xu C, Keays M, Levine AP, Kharchenko PV, Regev A, et al. Cell type

ontologies of the Human Cell Atlas. Nat Cell Biol. 2021;23:1129–35.
2. Schriml LM, Munro JB, Schor M, Olley D, McCracken C, Felix V, et al. The Human Disease Ontology 2022 update. Nucleic Acids Res. 2021;50:D1255–61.
3. Smith B, Ashburner M, Rosse C, Bard J, Bug W, Ceusters W, et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nat Biotechnol. 2007;25:1251–5.
4. Whetzel PL, Noy NF, Shah NH, Alexander PR, Nyulas C, Tudorache T, et al. BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. Nucleic Acids Res. 2011;39:W541–5.
5. Ong E, Xiang Z, Zhao B, Liu Y, Lin Y, Zheng J, et al. Ontobee: A linked ontology data server to support ontology term dereferencing, linkage, query and integration. Nucleic Acids Res. 2016;45:D347–52.
6. Zhao Y, Wang J, Chen J, Zhang X, Guo M, Yu G. A Literature Review of Gene Function Prediction by Modeling Gene Ontology. Front Genet [Internet]. 2020 [cited 2023 Nov 22];11. Available from: https://www.frontiersin.org/articles/10.3389/fgene.2020.00400
7. Gu Z, Hübschmann D. simplifyEnrichment: A Bioconductor Package for Clustering and Visualizing Functional Enrichment Results. Genomics Proteomics Bioinformatics. 2023;21:190–202.
8. Guo X, Liu R, Shriver CD, Hu H, Liebman MN. Assessing semantic similarity measures for the characterization of human regulatory pathways. Bioinformatics. 2006;22:967–73.
9. Yu G, Luo W, Fu G, Wang J. Interspecies gene function prediction using semantic similarity. BMC Syst Biol. 2016;10:121.
10. Garla VN, Brandt C. Semantic similarity in the biomedical domain: an evaluation across knowledge sources. BMC Bioinformatics. 2012;13:261.
11. Miller GA. WordNet: a lexical database for English. Commun ACM. 1995;38:39–41.
12. Mazandu GK, Chimusa ER, Mulder NJ. Gene Ontology semantic similarity tools: survey on features and challenges for biological knowledge discovery. Brief Bioinform. 2017;18:886–901.
13. Greene D, Richardson S, Turro E. ontologyX: a suite of R packages for working with ontological data. Bioinformatics. 2017;33:1104–6.
14. Yu G, Li F, Qin Y, Bo X, Wu Y, Wang S. GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. Bioinformatics. 2010;26:976–8.
15. Fröhlich H, Speer N, Poustka A, Beißbarth T. GOSim – an R-package for computation of information theoretic GO similarities between terms and gene products. BMC Bioinformatics. 2007;8:166.
16. Zhao C, Wang Z. GOGO: An improved algorithm to measure the semantic similarity between gene ontology terms. Sci Rep. 2018;8:15107.
17. Lin D. An Information-Theoretic Definition of Similarity. Proc Fifteenth Int Conf Mach Learn. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1998. p. 296–304.
18. Wu Z, Palmer M. Verbs semantics and lexical selection. Proc 32nd Annu Meet Assoc Comput Linguist [Internet]. USA: Association for Computational Linguistics; 1994 [cited 2023 Nov 13]. p. 133–8. Available from: https://dl.acm.org/doi/10.3115/981732.981751
19. Mazandu GK, Mulder NJ. A Topology-Based Metric for Measuring Term Similarity in the Gene Ontology. Adv Bioinforma. 2012;2012:e975783.
20. Nagar A, Al-Mubaid H. A New Path Length Measure Based on GO for Gene Similarity with Evaluation using SGD Pathways. 2008 21st IEEE Int Symp Comput-Based Med Syst [Internet]. 2008 [cited 2023 Nov 13]. p. 590–5. Available from: https://ieeexplore.ieee.org/document/4562064
21. Jackson RC, Balhoff JP, Douglass E, Harris NL, Mungall CJ, Overton JA. ROBOT: A Tool for Automating Ontology Workflows. BMC Bioinformatics. 2019;20:407.
22. Schulz H-J. Treevis.net: A Tree Visualization Reference. IEEE Comput Graph Appl. 2011;31:11–5.
23. Resnik P. Using Information Content to Evaluate Semantic Similarity in a Taxonomy [Internet]. arXiv; 1995 [cited 2023 Oct 25]. Available from: http://arxiv.org/abs/cmp-lg/9511007
24. Schlicker A, Domingues FS, Rahnenführer J, Lengauer T. A new measure for functional similarity of gene products based on Gene Ontology. BMC Bioinformatics. 2006;7:302.
25. Song X, Li L, Srimani PK, Yu PS, Wang JZ. Measure the Semantic Similarity of GO Terms Using Aggregate Information Content. IEEE/ACM Trans Comput Biol Bioinform. 2014;11:468–76.
26. Carey VJ. Ontology concepts and tools for statistical genomics. J Multivar Anal. 2004;90:213–28.
27. Jain S, Bader GD. An improved method for scoring protein-protein interactions using semantic similarity within the gene ontology. BMC Bioinformatics. 2010;11:562.
28. Caniza H, Romero AE, Heron S, Yang H, Devoto A, Frasca M, et al. GOssTo: a stand-alone application and a web tool for calculating semantic similarities on the Gene Ontology. Bioinformatics. 2014;30:2235–6.
29. Mazandu GK, Chimusa ER, Mbiyavanga M, Mulder NJ. A-DaGO-Fun: an adaptable Gene

Ontology semantic similarity-based functional analysis tool. Bioinformatics. 2016;32:477–9.
30. Schoch CL, Ciufo S, Domrachev M, Hotton CL, Kannan S, Khovanskaya R, et al. NCBI Taxonomy: a comprehensive update on curation, resources and tools. Database. 2020;2020:baaa062.
31. Hastings J, Owen G, Dekker A, Ennis M, Kale N, Muthukrishnan V, et al. ChEBI in 2016: Improved services and an expanding collection of metabolites. Nucleic Acids Res. 2016;44:D1214–9.
32. He Y, Liu Y, Zhao B. OGG: a Biological Ontology for Representing Genes and Genomes in Specific Organisms. CEUR Workshop Proc. 2014;1327:13–20.
33. He Y, Cowell L, Diehl A, Mobley H, Peters B, Ruttenberg A, et al. VO: Vaccine Ontology. Nat Preced. 2009;1–1.

Gene Ontology, Biological Process, 27942 terms



Top terms

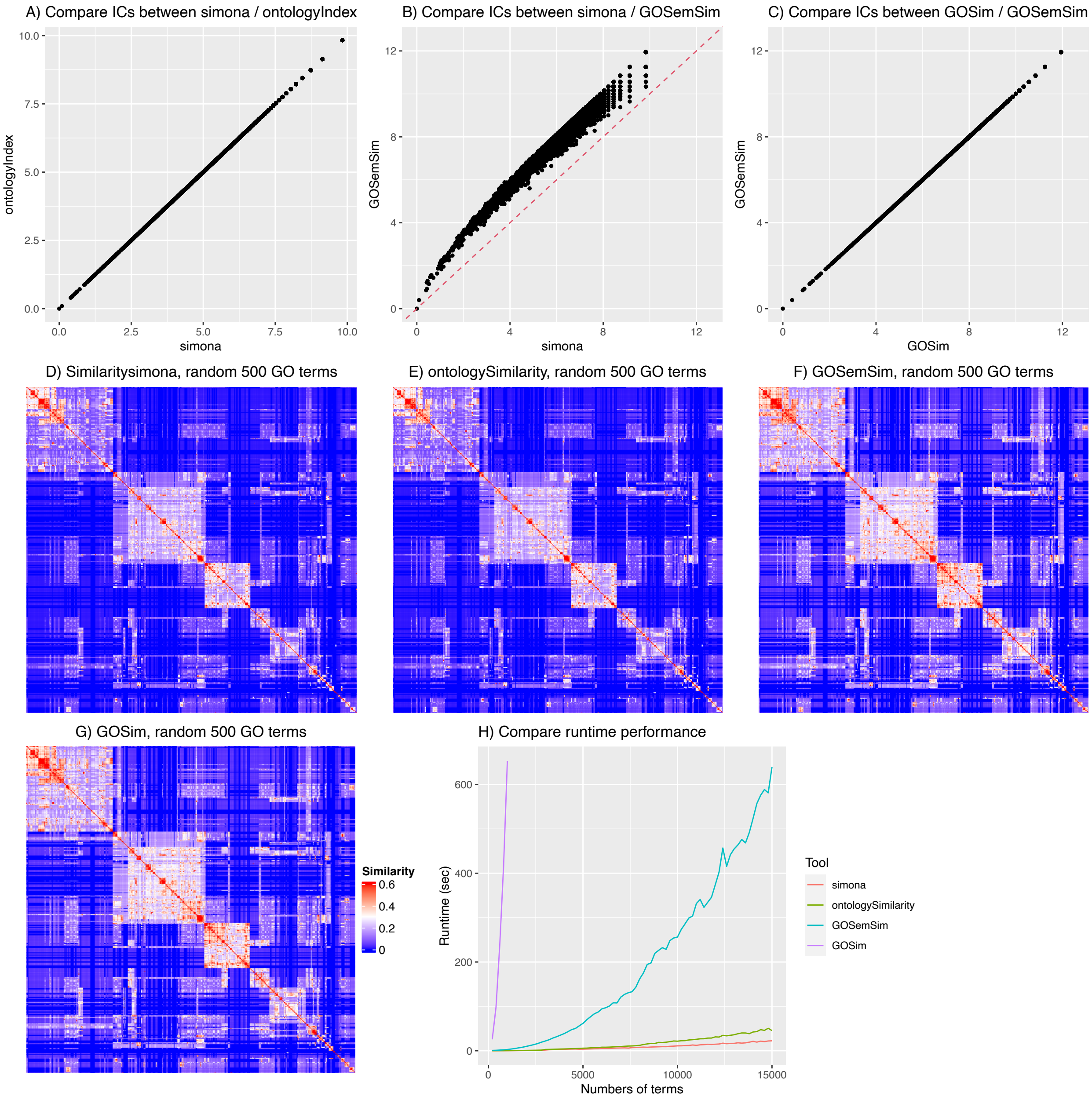
- metabolic process
- cellular process
- viral process
- multicellular organismal process
- developmental process
- growth
- biological process involved in interspecies interaction between organisms
- rhythmic process
- response to stimulus
- localization
- regulation of biological process
- regulation of biological quality
- regulation of molecular function

Relations

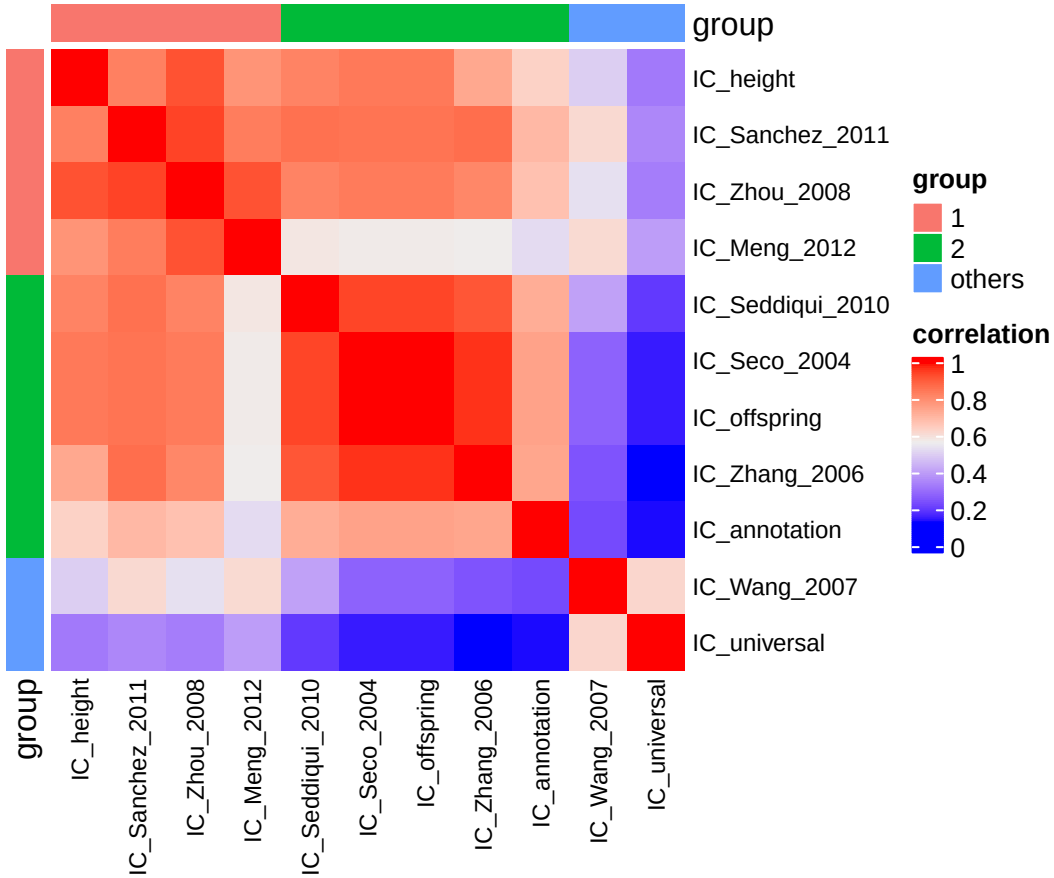
- is_a
- part_of

p.adjust

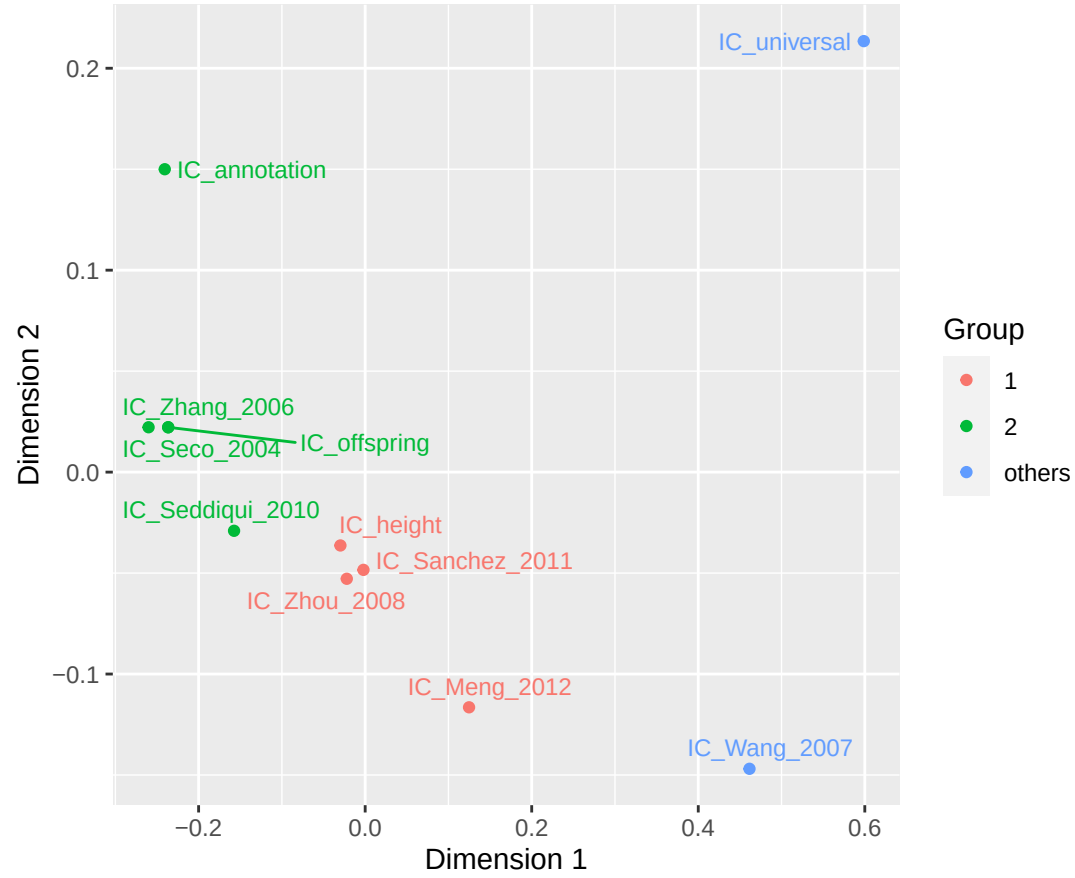
- 0.01
- 0.001
- 0.0001


A) Compare ICs between simona / ontologyIndex
ontologyIndex
simona
B) Compare ICs between simona / GOSemSim
GOSemSim
simona
C) Compare ICs between GOSim / GOSemSim
GOSemSim
GOSim
D) Similaritysimona, random 500 GO terms
E) ontologySimilarity, random 500 GO terms
F) GOSemSim, random 500 GO terms
G) GOSim, random 500 GO terms
Similarity
0.6
0.4
0.2
0
H) Compare runtime performance
Runtime (sec)
Numbers of terms
Tool
simona
ontologySimilarity
GOSemSim
GOSim

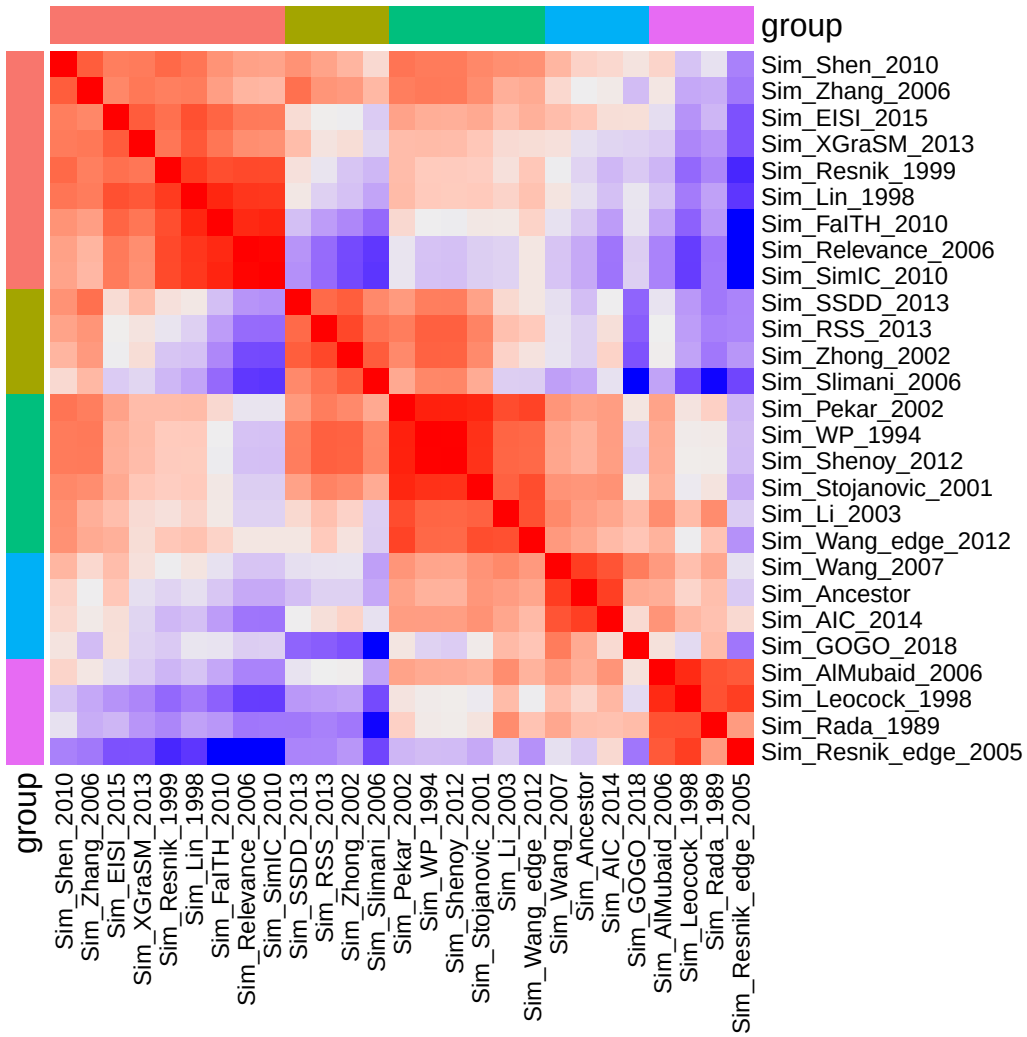
A) IC correlation, GO BP



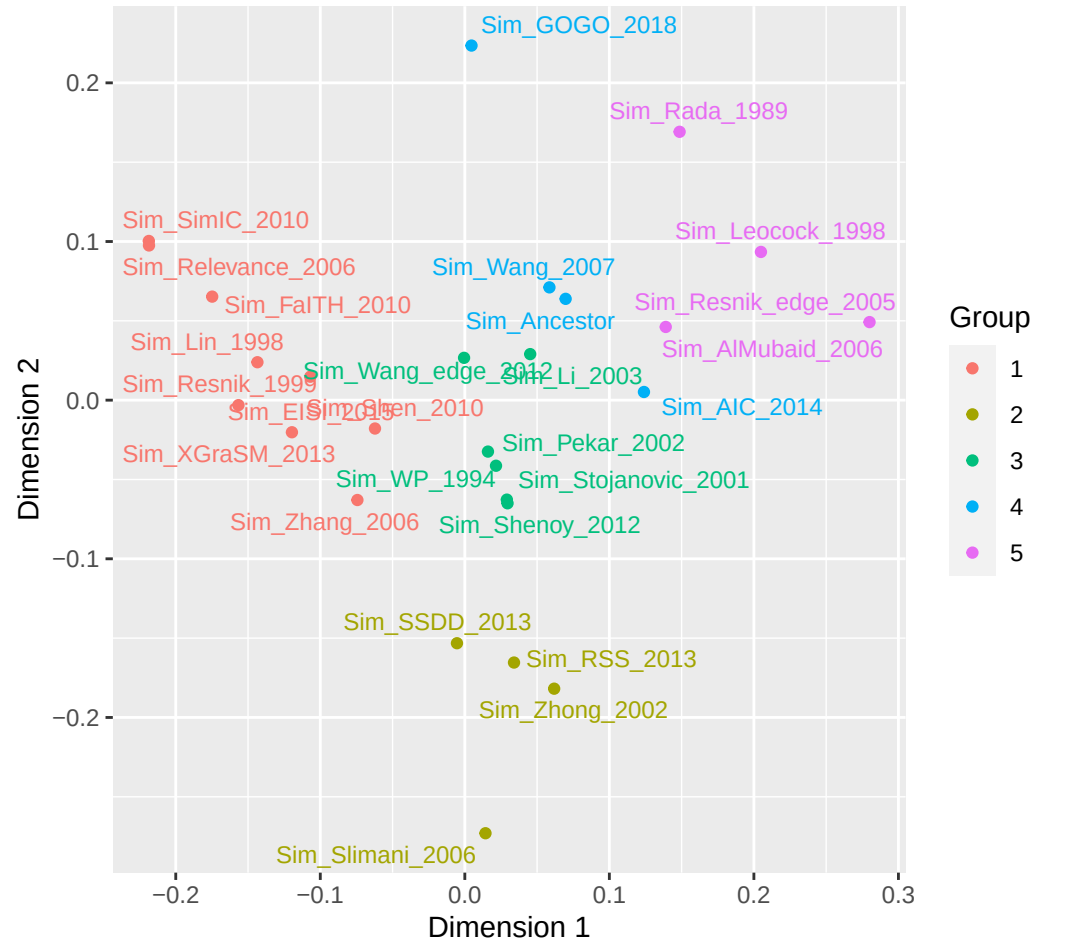
B) MDS based on IC correlations



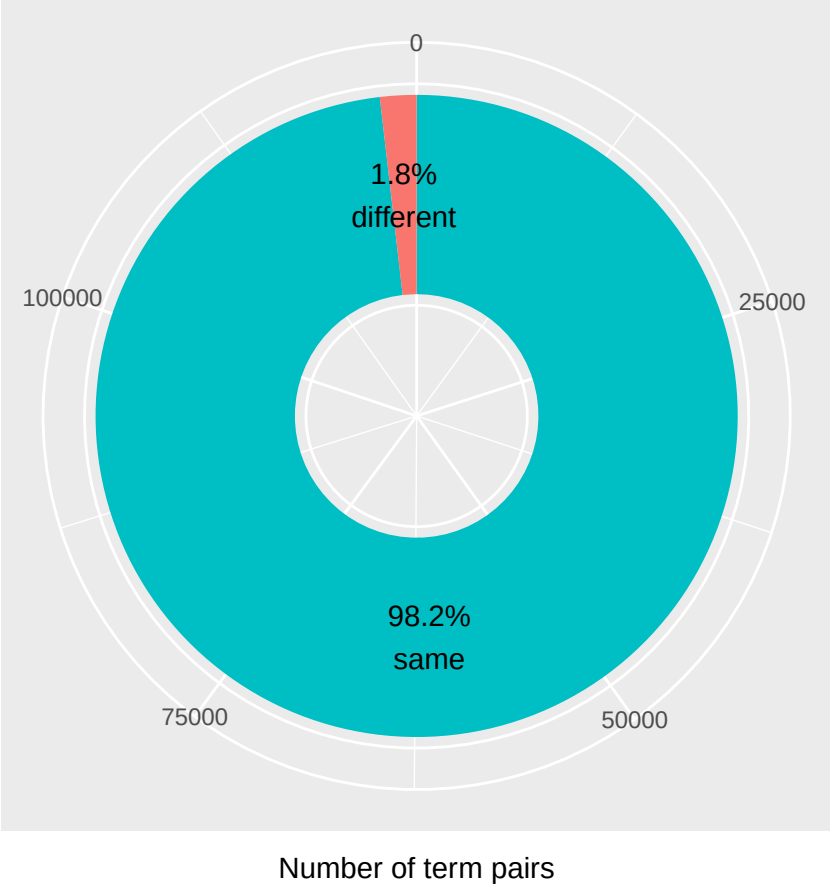
C) Semantic similarity correlation, GO BP



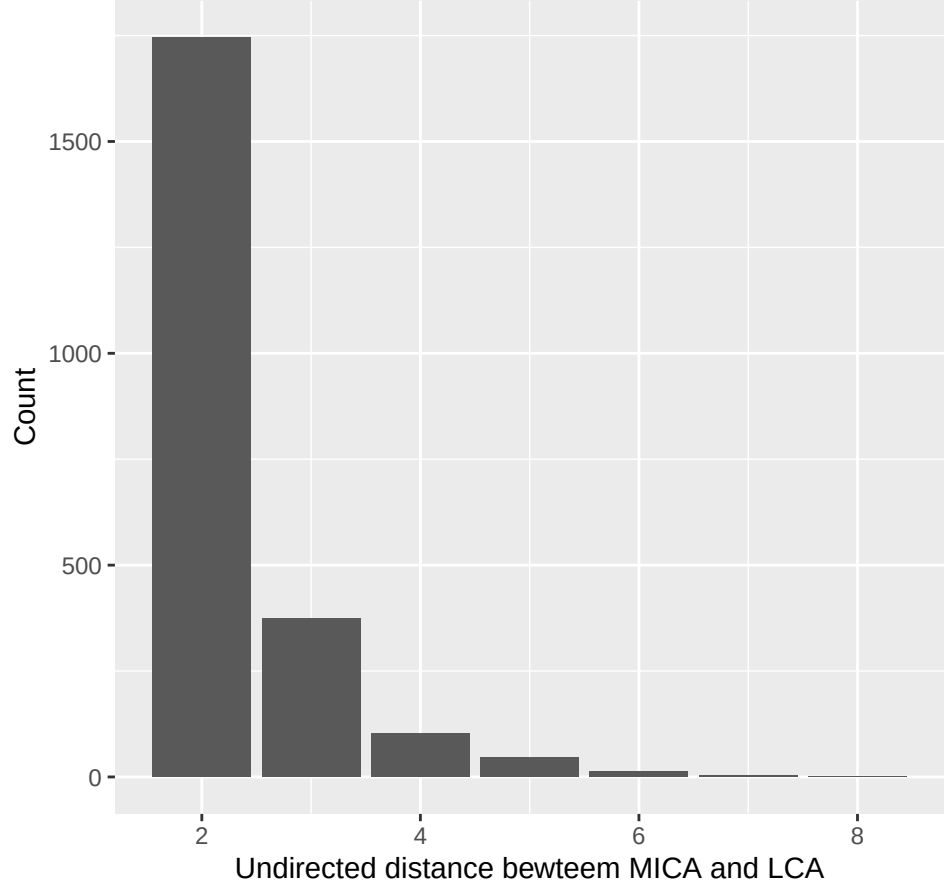
D) MDS based on similarity correlations



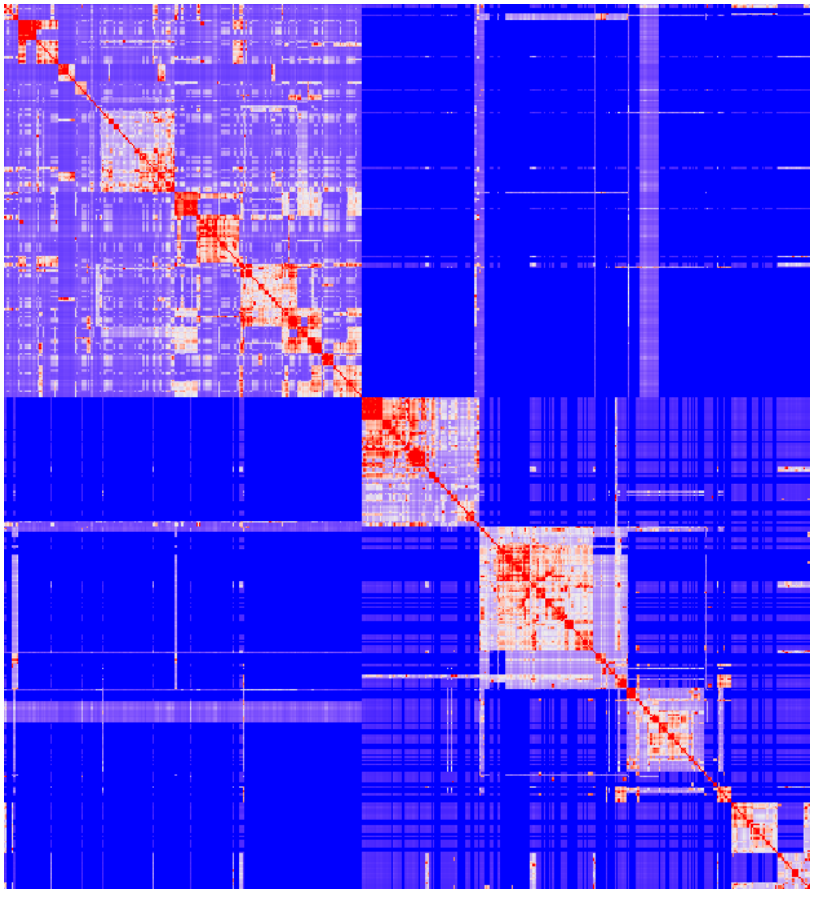
A) Compare MICA and LCA



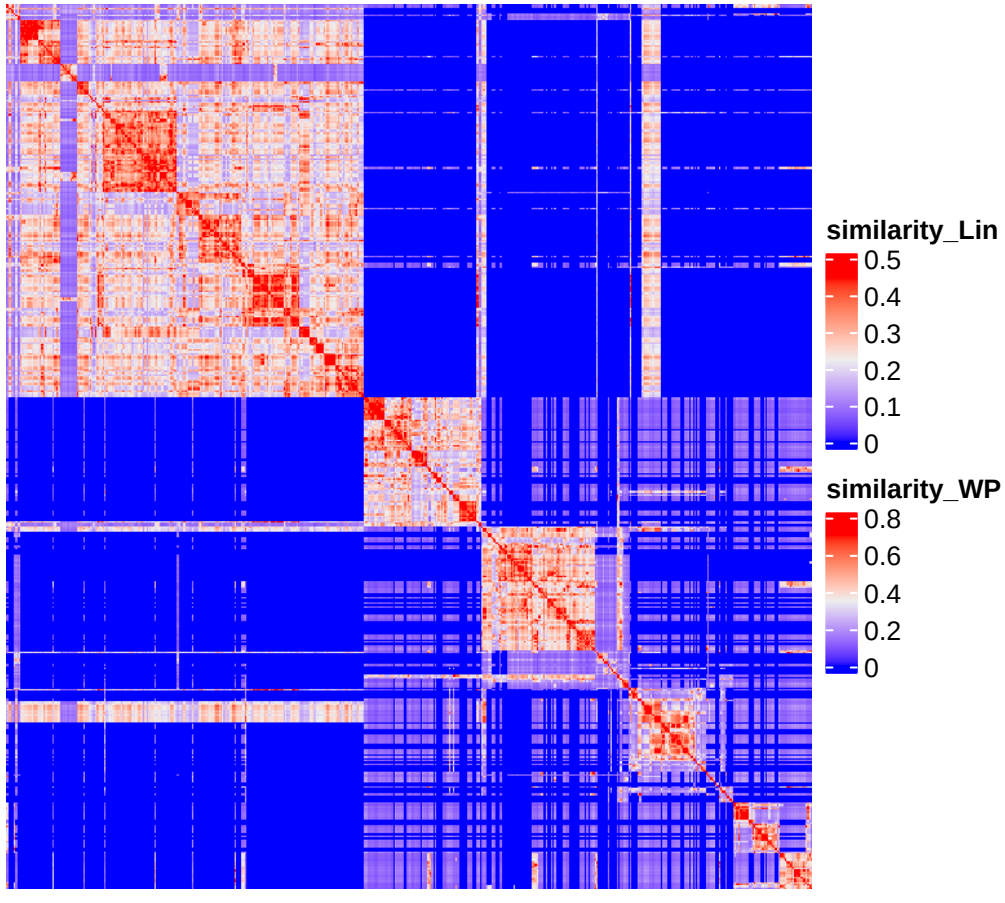
B) On term pairs with different MICA and LCA



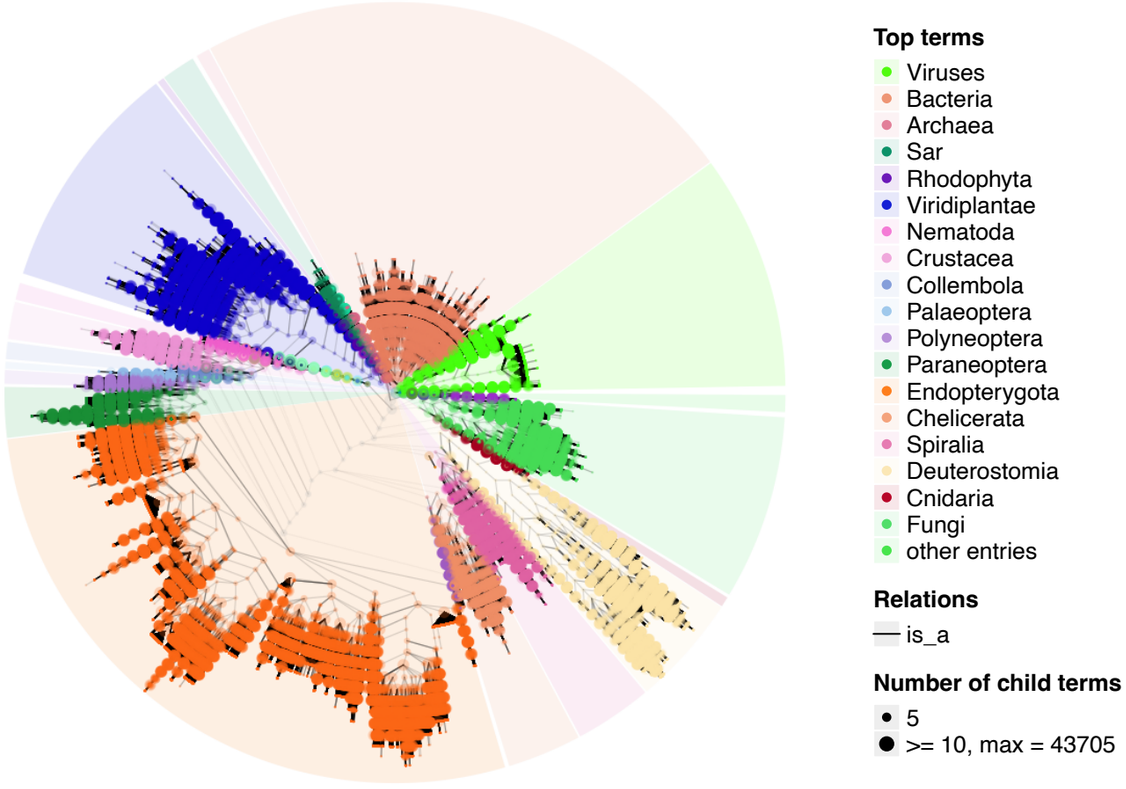
C) Sim_Lin_1998, based on MICA/IC_annotation



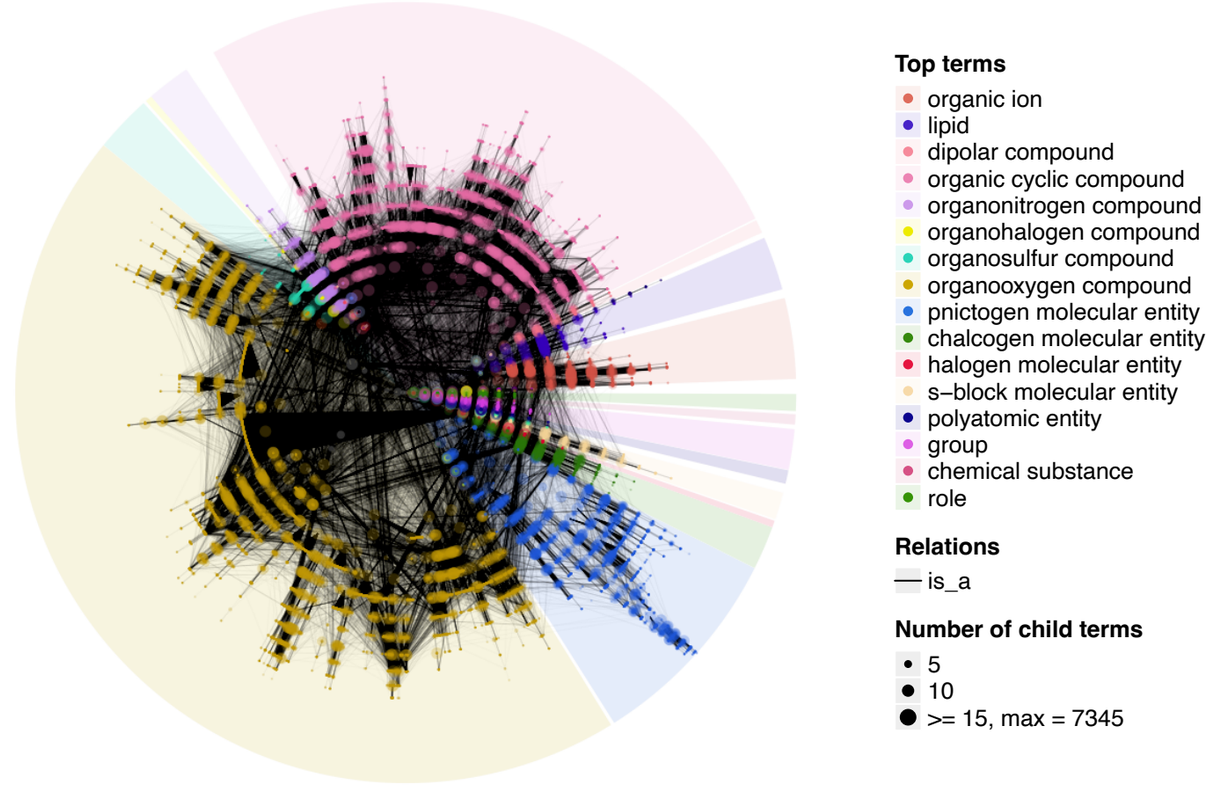
D) Sim_WP_1994, based on LCA/depth



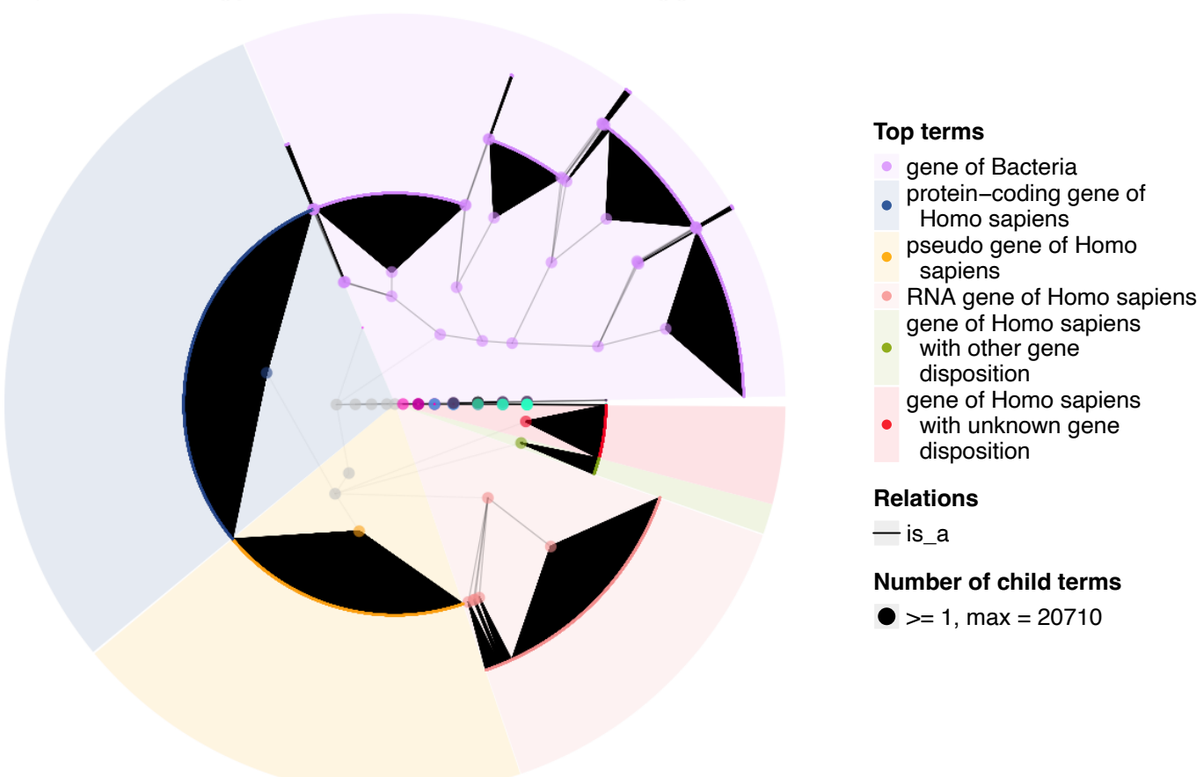
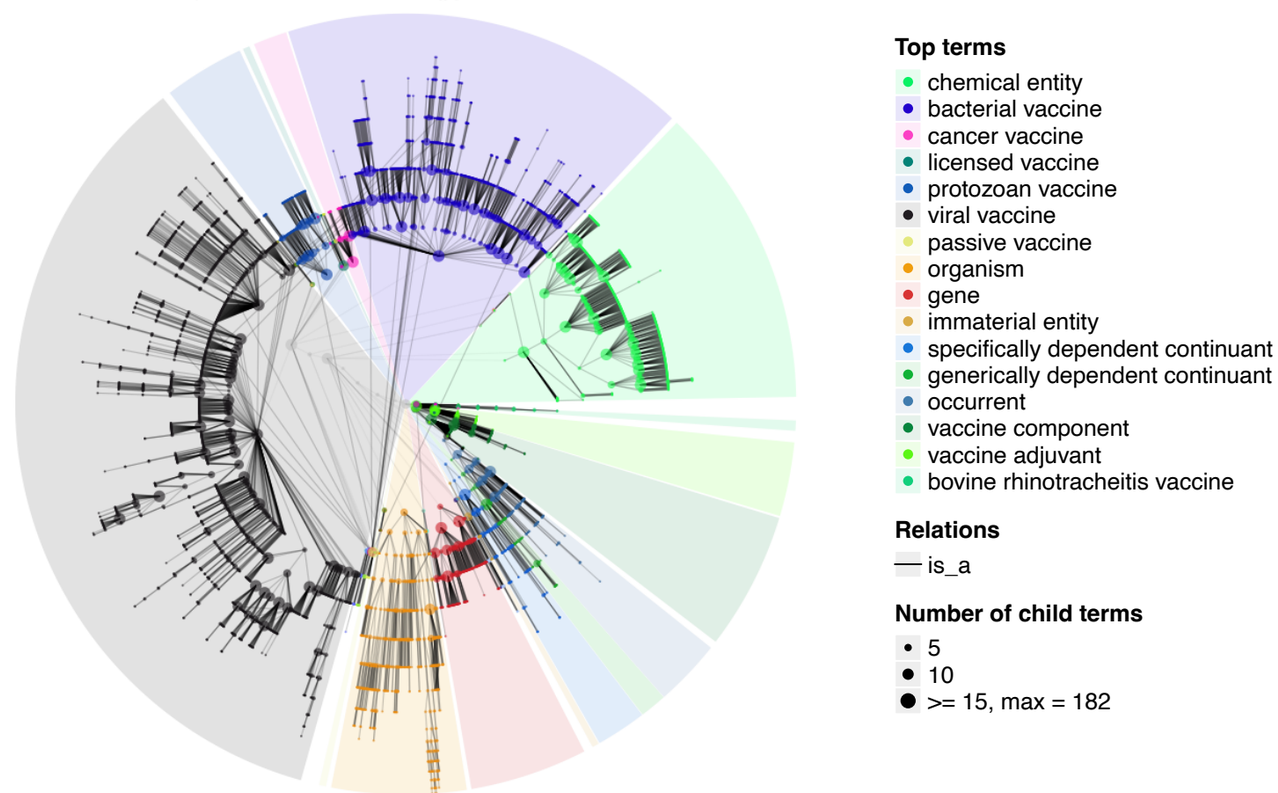
A) NCBI Taxonomy / ncbitaxon, 2511801 terms



B) Chemical Entities of Biological Interest / chebi, 164807 terms



C) The Ontology of Genes and Genomes / ogg, 69688 terms



D) Vaccine Ontology / vo, 7342 terms