The PySML Interface Manual

Gaston K. Mazandu [1,2,3*] *et al.*

Type Library
Title An open Python library implementing Semantic Similarity Measures [Library]
Version 2.5.1

Contributors Gaston K. Mazandu[1,2,3*], Kenneth Opap[2], Funmilayo L. Makinda[2,3], Victoria Nembaware[1], Francis Agamah[1], Christian Bope[1], Emile R. Chimusa[1], Ambroise Wonkam[1] and Nicola J. Mulder[2]

[1] *Division of Human Genetics, Department of Pathology, Institute of Infectious Disease and Molecular Medicine, University of Cape Town (UCT), Medical School, Anzio Road, Observatory 7925, Cape Town, South Africa.*

[2] *Division of Computational Biology, Department of Integrative Biomedical Sciences, Institute of Infectious Disease and Molecular Medicine, University of Cape Town, Medical School, Anzio Road, Observatory 7925, Cape Town, South Africa.*

[3] *African Institute for Mathematical Sciences (AIMS), Melrose Road, Muzenberg 7945, South Africa.*

Maintainer Mazandu GK <gmazandu@{gmail.com, uct.ac.za}, kuzamunu@aims.ac.za>

General description

PySML is a portable and expandable Python library enabling the retrieval of Semantic Similarity (SS) scores for any ontology to overcome issues related to computation, reproducibility and reusability of SS scores for any ontology in any application. PySML implements a large collection of SS measures consisting of 9 information content (IC) approaches, yielding over 624 ontology concept and 4430 entity pair-wise SS measures.

This library provides a more general tool able to analyze even the newly developed ontology and annotated set of entities. PySML deals with any ontology, independently of the file format (OBO, OWL or RDF), whether the file is already stored on a local computer or server, or to be directly retrieved via an online source provided a URL. It reads different ontology file formats using the pronto python library and outputs different results using the tabulate python module. PySML also contains modules implementing common applications related to semantic similarity measures.

Depends	Python ($\geq$ 2.7)
requires	python-networkx, [python-scipy, python-matplotlib]
License	GLP (`https://www.gnu.org/licenses/gpl-3.0.en.html`)
URL	`http://web.cbio.uct.ac.za/ITGOM/post-analysis-tools/pysml-dev/` and `https://github.com/gkm-software-dev/post-analysis-tools`

Release date Friday 7th May, 2021–04:50

*To whom correspondence should be addressed. Tel: +27 21 650 3463; Email: gaston.mazandu@uct.ac.za

Contents

Appendix-1 PySML Administration and Usage

1. PySML administration

The main website for the PySML library is `http://web.cbio.uct.ac.za/ITGOM/post-analysis-tools/pysml-dev` where users can find essential information about obtaining PySML. It is freely downloadable under GNU General Public License (GPL), pre-compiled for Linux version and protected by copyright laws, a free software and comes with ABSOLUTELY NO WARRANTY. Users are free to copy, modify, merge, publish, distribute and display information contained in the package, provided that it is done with appropriate citation of the library and by including the permission notice in all copies or substantial portions of the module contained in this package.

The whole lbrary itself is relatively small with a total of 33MB, the Gene Ontology file to be used as default ontology taking 30MB. PySML contains one main module and one main folder containing modules required for retrieving SS scores, running some related common applications, reading ontology files in any format (OWL, OBO and RDF) and formatting results to be displayed on the screen or written into a file. It is currently maintained by one member of the core-development team, Gaston K. Mazandu <gmazandu@gmail.com, gaston.mazandu@uct.ac.za, kuzamunu@aims.ac.za>, who regularly updates the information available in this package and makes every effort to ensure the quality of this information.

2. PySML Environment Management

PySML system is composed of one main high level folder: PySML and one main python module, `procsemsim.py`, which serves as an interface running different SS measures and applications implemented. The tests folder contains an illustrative Python modules for testing the PySML interface.

The PySML folder includes three Python modules: `informationcontent.py`, `conceptsimilarity.py` and `entitysimilarity.py` inplementing classes, building Python object for retrieving IC, concept and entity semantic similarity scores, respectively, described in the following pages. It also contains two sub-folders: smlapps and imports. The smpapps sub-folder contains source codes common applications related to semantic similarity measures implemented under PySML. The imports folder containing imported modules for reading an ontology and outputting different results.

As a library, PySML can be imported in a Python module, however, a user can also directly retrieve SS scores or run embedded applications in two main steps: User interface and input processing via a simple single command-line terminal as described in **Figure 1** (see main document). SS scores produced are presented in a table format, displayed on the screen or directed into a file.

3. PySML usage

PySML v2.5.1 requires Linux operating system and Python ($\geq$ 2.7.x) and one package, python-networkx for any application implemnted. This needs to be installed prior to the use of PySML and for running common related applications to SS measures: Entity Fuzzy classification, Entity Fuzzy Identification and Concept Fuzzy Enrichment, additional python-scipy and python-matplotlib should be installed.

To use PySML, the user needs to download the 'tar.gz' file and extract all files as follows:

```
tar xzf pysml-tool.tar.gz
```

or alternatively, it can also be retrieved from the github public platform using git clone command line as follows.

```
git clone https://github.com/gkm-software-dev/post-analysis-tools.git
```

After downloading and/or uncompressing, move to the folder post-analysis-tools/pysml-dev/, which should be set as a working directory where PySML and related commands are executed using the following terminal command:

```
cd post-analysis-tools/pysml-dev/
```

4. PySML licence and version

As pointed out previously, the PySML library is free to use under GNU General Public License. You are free to copy, distribute and display information contained herein, provided that it is done with appropriate citation of the library. Thus, by using the PySML library, it is assumed that you have read and accepted the agreement provided and that you agreed to be bound to all terms and conditions of this agreement. Please, use the following command line to see the package licence:

```
python setup.py --licence
```

To check the current version of the PySML interface, use the following terminal command:

```
python setup.py --version
```

5. Running PySML

Any SS measures or common related applications implemented under PySML can be processed through one main python module, `procsemsim.py`, which serves as an interface. Get help on how to run PySML through this interface module using the following command:

```
python procsemsim.py -h
```

The above command should produce the following output:

```
usage: procsemsim.py [-h] [-a ANNOT] [-f ONTOLOGY] [-m [MODELS [MODELS ...]]] [-o FILE]
[-d str [str ...]] -t str [-s int]
```

with different tags explained below:

-h, --help	show this help message and exit
-t str, --type str	Type of SS: ic (IC), cs (Term SS), es (Entity SS) (default: None)
-m [MODELS [MODELS ...]], --models [MODELS [MODELS ...]]	SS models to be considered (default: None)
-p str, --parameter str	Other necessary parameters needed for the models considered (default: None)
-d str [str ...], --data str [str ...]	Full path to the file containing list of terms, term-term pairs, entity-entity (default: None)
-a ANNOT, --annotationfile ANNOT	Full path to the file appropriate Entity-term mapping (default: None)
-f ONTOLOGY, --ontologyfile ONTOLOGY	Full path to the file appropriate Entity-term mapping (default: None)
-n str, --namespace str	The name space of the ontology being used (default: biological_process)
-o FILE, --out FILE	Naming the SS scores output file (default: current working folder)
-s int, --stream int	Output on (1) screen or (0) file (default: 1)

As highlighted by the `help` option, PySML is run using the following one line command:

```
python procsemsim.py -t ss-model -m models -p parameters -d dataset -a annotationfile -f ontologyfile -n namespace -o outputfile -s value
```

1. `ss-models`: Semantic similarity type, which may be: ic for information content (IC), cs for concept semantic smilarity or es for entity semantic similarity model. This parameter is required and must be provided.

2. `models`: These represent different models to be run and if no model is provided, then the default model is executed. The default model is: universal for IC, Nunivers for concept SS and BMA for entity SS if pair-wise models and SimGIC for group-wise. PySML allows user to run several models and they are space separated. For colon (:), comma (,) or dash (-) separated model and sub-model when a model required a sub-model. For example, ABM requires a concept SS model, e.g., the Lin et al. model, also concept SS model may required a specific IC model, e.g., Zhang et al. model. This entity SS model is then abm:lin:seco or abm,lin,seco or abm-lin-seco. Note that these model-submodel separators ':', ',' and '-' may be mixed within a model. If required model or sub-model is not provided, then the default indicated above is run.

3. `dataset`: A file containing list of concepts (for 'ic' or 'cs'), concept or entity pairs (for cs or es). This file is space, comma or colon pairs and it is only required for retrieving concept SS score (cs). These concepts, concept or entity pairwise can also be provide as space separated with comma separated pairwise concepts or entities where applicable. In case this dataset is not provided for ic and es, then the IC scores are produced for all active concepts in the ontology for ic and, for es, the pairwise entities are built from the annotation file, described in point 5 below.

4. `parameters`: This is a string-like dictionary in which each string-key is an argument representing a given model parameter. Model-parameter and argument symbol mapping is provided in **Table S1**. Other keys in this string-like dictionary include 'app' associated with the

IC approach to be used where required, TermStats and TermIC associated to a dictinary, either mapping concept counts for annatation-based approach or pre-computed IC scores, respectively.

5. `annotationfile`: An entity-concept mapping file and only required for retrieving entity SS score (es). If the file is provided, then the file should be space separated entity-concepts with ';' or ',' separated concepts. One can also provide a string-like a dictionary, in case where the perhaps number of entities is very limited.

6. `ontologyfile`: The ontology file or active URL to the ontology file. It worth noting that PySML can retrieve the ontology online if the URL is provided using an adjusted python package initially written by 'Martin Larralde (martin.larralde@ens-paris-saclay.fr)' under the "MIT" license. If this ontology file or associated URL is not provided, then the Gene Ontology (GO) in the tests folder is used. with 'biological_process' namespace by default.

7. `namespace`: The name space of the ontology being used. This is the case, for example, for GO which has three main name space or sub-onfology, namely biological_process, molecular_function and cellular_component. In this context, if this namespace is not provided, then 'biological_process' namespace by default.

8. `outputfile`: The path to the folder where the outputs should be written. If not provided, the current working directory is used.

9. `value`: An indicator whether the output should be dislayed on screen (1) or written in a file (0) in the directory provided in point 6. If not provided, it uses the default value, which is 1, i.e., displaying the outputs on the screen. Note that use python module `tabulate.py` for `pretty-print tabular data`, which is borrowed from other authors, specifically written by 'Sergey Astanin (s.astanin@gmail.com)' and collaborators under the "BSD" license.

It is worth recalling that, every time a default model is used, BMA is called for a default entity pairwise SS measure or SimGIC for group-wise, Nunivers is used for concept SS scores and universal is used as a default IC approach.

6 Illustrating PySML usage

As pointed out previously, move to the pysml-dev folder and run following commands for illustration. Please type these commands manually using the computer keyboard, do not use `copy` and `paste`:

```
python procsemsim.py -t es -m bma kstats ui -a "{'Prot1': ['GO:0000022', 'GO:0051231', 'GO:1903047', 'GO:0000278', 'GO:0007052', 'GO:0000023', 'GO:0005984'], 'Prot2':['GO:0000022', 'GO:0051231', 'GO:1903047', 'GO:0000278', 'GO:0007052'], 'Prot3':['GO:1903047', 'GO:0000278', 'GO:0007052', 'GO:0000023', 'GO:0005984']}"
```

This produces a table of entity pairwise 'Prot1', 'Prot2' and 'Prot3' SS scores for models (BMA, Nunivers, Universal), non-ontology Kappa-Statistics (SimKPS) and Jaccard-like (UI-like) using the ontology GO biological process by default and displaying the result on the screen.

The second, third and fourth commands retrieving concept SS and IC scores are given below:

```
python procsemsim.py -t cs -d GO:0000022 GO:0051231 GO:1903047 GO:0000278 GO:0007052
GO:0000023 GO:0005984 -m nunivers-zhou resnik:zhang wang wang_edge lin,zanchez aic wu -p
'dict(sigma = 0.7)'
```

```
python procsemsim.py -t ic -m meng universal zanchez zhang wang seco -f tests/go-basic.obo -s
0
```

```
python procsemsim.py -t ic -m meng universal zanchez zhang wang seco -d GO:1900309
GO:1900308 GO:1900303 GO:1900302 GO:0019990
```

The second command processes the concept (term) SS scores for following models: Nunivers based on Zhou et al IC approach, Resnik based on Zhang et al. approach, node based Wang et al. approach, edge based Wang et al. approach, Lin based on Zanchez et al. approach, AIC based on the IC universal approach and Wu et al. approach. Any approach requiring an IC approach, which is not provided, uses the IC-universal based approach as default and the key 'sigma', provided in the string-like dictionary (-p) is the parameter related to the Zhou et al. approach set to 0.7. The third command will process IC scores for Meng et al., Universal, Zanchez et al. Zhang et al. Wang et al. and Seco et al. models using the ontology provided under the tests folder with biological_process as a default ontology namespace, writing all ontology concept IC scores in a file whose the name is printed on the screen and located in the current working directory (by default), which is the pysml-dev directory. The fourth command is similar to the the second, but it uses a default ontology, which is provided in the tests folder with biological_process as ontology namespace by default, displaying on the screen (-s 1) by default, IC scores only for the five concepts provided.

7 Running PySML as a Python Package

As any python library or package, PySML can be imported and used in another Python models. For accessing and learning about different classes of the three main classes under PySML, InformationContent, ConceptSimilarity and EntitySimilarity, with ConceptSimilarity inheriting directly from InformationContent and EntitySimilarity inhereting directly from ConceptSimilarity. Please access the python interpreter or the command shell for interactive computing (IPython) and run following commands:

```
>>> from PySML import *
>>> help(InformationContent)
>>> help(ConceptSimilarity)
>>> help(EntitySimilarity)
```

8 Important notes

- To efficiently use the PySML library and to maximally benefit from its use, make sure that you have carefully read this PDF package documentation file, which is provided in the library.
- In some cases, you may need or be required to provide the name of the file. Please make sure that the full path to the file target is provided.
- make use of the full screen mode when displaying results on it for a nice and more adapted visualization.

Table S1. Mapping Semantic Similarity model parameter (par) to PySML argument (arg).Different models requiring parameters, as well as asociated arguments in PySML are highlited. It is always optimal to use default values, in which case, they are not required to be provide. However, if, for some reasons, other parameter values are needed, then these values should be dictionary values asscociated to arguments in this table, each provided as dictionary string key.

Model	PySML arg	Model par	PySML arg	Range	Type	Default
Zhou et al.	zhou	`sigma` (σ)	`sigma`	[0, 1]	float	0.5
Zhong et al.	zhong	`k`	`zk`	≥ 2	integer	2
Al-Mubaid et al.	almubaid	`k`	`ak`	≥ 1	float	1.0
		`alpha` (α)	`aa`	> 0	float	1.0
		`beta` (β)	`ab`	> 0	float	1.0
Li et al	li_edge	`alpha` (α)	`alpha`	≥ 0	float	0.2
		`beta` (β)	`beta`	> 0	float	0.6
Correction factor	cf	`epsilon` (ϵ)	`cf`	0 if no correction 1 for Graph-based 2 for Relevance 3 for SimIC	integer	0
Graph-based indicator	gr	-	`gr`	0 for XGraSM 1 for EISI	integer	0
Jian&Conrath variant	jv	-	`jv`	0 if Resnik-based 1 for Couto-based 2 for Leacock-based 3 for Garla-based 4 for Rada-based 5 for Canonical-based	integer	0
SimALN	aln	`alpha` (α)	`aaln`	> 0	float	1.0

9 Contributors

Gaston K. Mazandu, Kenneth Opap, Funmilayo L. Makinda, Victoria Nembaware, Emile R. Chimusa, Ambroise Wonkam and Nicola J. Mulder

Maintainer: Mazandu GK <gmazandu@gmail.com, gaston.mazandu@uct.ac.za, kuzamunu@aims.ac.za>

10 Main references

1. Mazandu, G. K., Chimusa, E. R., and Mulder, N. J. (2016) Gene ontology semantic similarity tools: survey on features and challenges for biological knowledge discovery. *Brief Bioinform* **18(5)**, 886–901.

2. Mazandu, G. K., Chimusa, E. R., Mbiyavanga, M., and Mulder, N. J. (2016) A-DaGO-Fun: An adaptable Gene Ontology semantic similarity based functional analysis tool. *Bioinformatics* **32(3)**, 477–479.

3. Mazandu GK, Mulder NJ (2013) DaGO-Fun: Tool for Gene Ontology-based functional analysis using term information content measures. BMC Bioinformatics 14: 284.

4. Mazandu GK, Mulder NJ (2013) Information content-based Gene Ontology semantic similarity approaches: Toward a unified framework theory. BioMed Research International 2013: Article ID 292063, 11 pages.

5. Mazandu GK, Mulder NJ (2014) Information Content-Based Gene Ontology Functional Similarity Measures: Which One to Use for a Given Biological Data Type? PLoS ONE 9(12): e113859.

11 Questions, Comments and Report Bugs

The PySML team is striving to aggregate knowledge about scoring semantic similarity into an organized structure in order to ease the retrieval SS scores for any ontology, resolving issues related to computation, reproducibility and reusability of SS scores for any ontology in any application, and effectively producing scores in realistic timeframes. manipulation. However, PySML does not guarantee the quality or accuracy of different result outputs. Thus, if it happens that you find errors, please contact the primary source of data set used for more information. If you feel that the errors may be due to some systematic error in the PySML library, please contact the library maintainer at <gmazandu@gmail.com, gaston.mazandu@uct.ac.za, kuzamunu@aims.ac.za>.

12 PySML copyright and license

The PySML library is free to use under GNU General Public License (GLP: `https://www.gnu.org/licenses/gpl-3.0.en.html`). You are free to copy, distribute and display information contained herein, provided that it is done with appropriate citation of the tool.

13 Citing PySML.

The manuscript is being prepared for publication, before its publication you can cite the preliminary report:

"Mazandu GK, Opap K, Makinda FL, Nembaware V, Agamah F, Bope C, Chimusa ER, Wonkam A, Mulder NJ. (2020) PySML–An open library implementing semantic similarity measures and common related applications". Technical report 2018, H3ABioNet-AIMS node and SADaCC, AIMS & UCT, South Africa. `http://web.cbio.uct.ac.za/ITGOM/post-analysis-tools/pysml-dev/`.

14 Other information about PySML development

Please refer to `http://web.cbio.uct.ac.za/ITGOM/post-analysis-tools/pysml-dev/PKG-INFO` (See some other details about the PySML development)

or

go to the local pysml folder and type the following command line for the short description:

```
python setup.py --description
```

or alternatively,

```
python setup.py --long-description
```

for the long description.

Appendix-2 Semantic Similarity measures

This survey provides an overview of mathematical expressions of different term information content (IC) models, term semantic similarity approaches and functional similarity measures between proteins or genes that were introduced or updated for use in the context of the Gene Ontology (GO). It is assumed that IC models are partitioned into two families: annotation and topology-based IC families [1], the concept or term semantic similarity approaches are divided into three main categories: IC- or node-based, edge- or path-based and 'hybrid' categories, and that entity semantic similarity measures are classified into two main classes: Ontology- and non ontology-based measures (see **Figure 2** in the main document). Building upon this assumption, this document will be dynamic in nature in the way that any new semantic similarity measure discovered will be classified and added to this document in order to remain up to date. This aims to build an encyclopedia of mathematical expressions of all existing semantic similarity measures in the context of WordNet and GO. It is assumed that GO is divided into three separate ontologies, namely, Molecular Function (MF), Biological Process (BP) and Cellular Component (CC) with GO identifiers (IDs) GO:0003674, GO:0008150 and GO:0005575, respectively, as roots for these ontologies, located at level 0, the reference level. These are assumed to be biologically meaningless with the lowest term information content (IC) value. Unless specified explicitly, in the rest of this document, the left side object denotes the right side object description.

Object	Description
r	The root of a given ontology.
LCA	Lowest Common Ancestor.
$\text{LCA}(a, b)$	The set of LCAs between terms a and b.
MICA	Most Informative Common Ancestor.
$\text{MICA}(a, b)$	The set of MICAs between terms a and b.
c	A MICA or LCA between terms a and b, depending on the context.
$\mathcal{A}_s$	The subsumers of the term s, i.e., $\mathcal{A}_s = \mathcal{A} \cup \{s\}$ with $\mathcal{A}$ the set of ancestors of the term s.
D_s	The set of term hyponyms (descendants) to the term s.
$\lvert A \rvert$	The length (number of elements in) of the set A.
$\text{len}(a, b)$	The length (number of edges) of the longest path connecting terms a and b.
$\text{len}_s(a, b)$	The length of the longest path connecting terms a and b via the term s.
$\overline{\text{len}(r, a)}$	The average path length connecting from the root to the term a. Similarly, $\overline{\text{len}_s(r, a)}$ is the average path length linking the root to the term a via $s \in \mathcal{A}_a$.
$\mathcal{D}_{\text{sp}}(a, b)$	The shortest distance (i.e., minimum number of edges or links) connecting a and b, i.e., $\mathcal{D}_{\text{sp}}(a, b) = \min\left\{\mathcal{D}_{\text{sp}}(a, s) + \mathcal{D}_{\text{sp}}(b, s) : s \in \mathcal{A}_a \cap \mathcal{A}_b\right\}$.
$\mathcal{D}_x(u, v)$	The distance, $\mathcal{D}(u, v)$, between concepts u and v for the method x.
$\delta(s)$	The depth of the term s in the ontology, i.e., the length of the longest path from the root term r and corresponds to the level of the term s in the ontology.
$\delta_{\max}$	The maximum depth in the ontology or taxonomy.
T_p	The set of terms annotating the protein p.
$\mathcal{A}_p$	The set of terms annotating the protein p including ancestors of these terms, i.e., $\mathcal{A}_p = \bigcup_{s \in T_p} \mathcal{A}_s$.

1 Computing IC values

From their conception, term information content (IC) approaches can be divided into two families: annotation and topology-based IC families. While the topology-based family exploits only the intrinsic topology of the ontology directed acyclic graph (DAG), the annotation-based family requires the addition of annotation data for the corpus under consideration. With the exception of the topology-based model proposed by Wang et al. [2], all other approaches compute the IC of terms in a similar way (i.e., using log function) despite their conceptual differences. The IC value of the term is given by

$$\mathrm{IC}\left(x\right) = -\ln\left(p\left(x\right)\right) \tag{1}$$

1.1 Annotation-based IC model

In the case of annotation-based approaches, $p\left(x\right)$ is the relative frequency of the term x in the protein dataset under consideration, obtained from frequency $f\left(x\right)$ representing the number $\eta\left(x\right)$ of proteins annotated with the term x in the dataset considering the 'true-path rule' principle of the ontology DAG structure. Thus, this frequency $f\left(x\right)$ is given by

$$f\left(x\right) = \begin{cases} \eta\left(x\right) & \text{if } x \text{ is a leaf} \\ \eta\left(x\right) + \sum\limits_{z \in \mathcal{C}_h(x)} \eta\left(z\right) & \text{otherwise.} \end{cases}$$

where $\mathcal{C}_h\left(x\right)$ is the set of ontology concepts having x as a parent, and a leaf is a term that has no child.

1.2 GO-universal IC model

In the context of the GO-universal approach [3], $p(x)$ is called the topological position characteristic of x, recursively obtained using its parents gathered in the set $\mathcal{P}_x = \{t : (t, x) \in \mathcal{L}_{GO}\}$ where $\mathcal{L}_{GO}$ expresses the set of links in the ontology-DAG and $(t, x) \in \mathcal{L}_{GO}$ represents the link or association between a given parent t and its child x. This topological position characteristic, $p(x)$, is given by

$$p\left(x\right) = \begin{cases} 1 & \text{if } x \text{ is a root} \\ \prod\limits_{t \in \mathcal{P}_x} \frac{p\left(t\right)}{|\mathcal{C}_h(t)|} & \text{otherwise} \end{cases} \tag{2}$$

with $|\mathcal{C}_h(t)|$ the number of children with term t as parent.

1.3 Zhang et al. IC model

In the case of the topology-based approach introduced by Zhang et al. [4], $f\left(x\right)$ is called the count of the term x, it depends only on the children of a given ontology concept and is numerically equal to the sum of counts of all its children. $f\left(x\right)$ is calculated using a recursive formula starting from leaves in the hierarchical structure, and given by

$$f\left(x\right) = \begin{cases} 1 & \text{if } x \text{ is a leaf} \\ \sum\limits_{z \in \mathcal{C}_h(x)} f\left(z\right) & \text{otherwise.} \end{cases}$$

In order to better relate the Zhang et al. model to other topology-based IC models that have been suggested in the context of WordNet, such as Seco et al. [5] and Sánchez et al. [6], and which are being updated to be applied in the context of GO, it is worth mentioning that this count, $f(x)$, of the term x, represents the number of term hyponyms, i.e., the number of descendants of the term x in the ontology. The relative frequency $p\,(x)$, called the D-value in the case of the context of the Zhang et al. model, is then computed independently for each ontology and given by

$$p\,(x) = \frac{f\,(x)}{f\,(r)}$$

where $f\,(r)$ is the frequency (count) of the root term in the ontology under consideration. It is worth mentioning that the Zhang et al. model for computing the IC score follows the Seco et al. approach [5] in its conception.

1.4 Seco et al. IC model

The Seco et al. model is a 'Zhang et al.-like' normalized IC model, i.e., with IC values ranging between 0 and 1, and given by:

$$\mathrm{IC}_{\mathrm{Seco}}\,(x) = \frac{\log\left(\frac{f(x)+1}{f(r)+1}\right)}{\log\left(\frac{1}{f(r)+1}\right)} = 1 - \frac{\log\,(f(x)+1)}{\log\,(f(r)+1)} \tag{3}$$

It has been noted that the Zhang et al. model, which can be considered to be the Seco et al. model applied to GO, overestimates the IC values of terms and in many instances it may fail to effectively distinguish terms at different levels of specificity (i.e., one appears in an upper level of the structure with respect to the other) by assigning equal count score to these terms. This suggests that these terms are equally specific, ignoring that a child term is supposed to be more specific than the parent [3]. On the other hand, the fact that the Seco et al model includes the root of the ontology in its count value biases the IC values by overestimating or underestimating these scores. In attempting to solve the issue caused by Seco et al. related models, Zhou et al. [7] suggested adjusting these models using the relative depth of the concept in the taxonomy as shown below.

1.5 Zhou et al. IC model

The Zhou et al. IC model [7] is a hybrid model that weighs both 'Zhang et al.-like' normalized IC models and the depth of the term in order to correct their incapability to distinguish different level of specificity, and given by:

$$\mathrm{IC}_{\mathrm{Zhou}}\,(x) = \sigma * \mathrm{IC}_{\mathrm{Seco}}\,(x) + (1-\sigma) * \frac{\log\,(\delta(x))}{\log\,(\delta_{\max})} \tag{4}$$

where the depth of the root is set to 0, i.e., $\delta\,(r) = 0$ to avoid $\log(0)$ and σ is a tuning factor that adjusts the contribution of the two control values involved in the IC assessment. Initially σ was set to 0.5, but it should normally depend on the nature of the ontology and a trade-off between the control values is needed to optimally select the value of the tuning factor σ.

1.6 Seddiqui et al. IC model

Like the Zhou et al. model, the Seddiqui et al. model [8] is also a hybrid model and suggests the use of the number of relations of the term, i.e., the number of terms in the ontology that are connected to it, instead of the depth of the term, in which case, the IC value of term x is given by:

$$\mathrm{IC}_{\mathrm{Sed}}(x) = (1-\sigma) * \mathrm{IC}_{\mathrm{Seco}}(x) + \sigma * \frac{\log(\lambda(x)+1)}{\log(\lambda_{\mathrm{edge}}+1)} \tag{5}$$

where $\lambda(x)$ is the number of relations of x, λ_{edge} total number of relations (edges) in the ontology and the tuning factor σ given by:

$$\sigma = \frac{\log(\lambda_{\mathrm{edge}}+1)}{\log(\lambda_{\mathrm{edge}}) + \log(\lambda_{\mathrm{node}})} \tag{6}$$

with λ_{node} the number of terms in the ontology under consideration. It is clear that, in this case, the tuning factor depends on the topology of the ontology.

1.7 Sánchez et al. IC model

Sánchez et al. [6] suggested the IC model which is directly proportional to the number of ontological subsumers of the term for which the IC value is being computed and inversely proportional to the amount of leaves connected to the term. This should capture the concept of specificity in the ontological structure without relying on tuning factors and taking into account the depth and the overall ontological structure by computing the IC value as the ratio between its level of generality (expressed by the amount of taxonomical subsumers) with respect to its level of specificity (expressed by the number of leaves). Thus, the IC value is computed as follows:

$$\mathrm{IC}_{\mathrm{Zánchez}}(x) = -\log\left(\frac{\frac{\zeta_x}{|\mathcal{A}_x|}+1}{\zeta_{\mathrm{leaf}}+1}\right) \tag{7}$$

where ζ_x is the number of leaves connected to the term x and ζ_{leaf} the number of all leaves in the ontology, corresponding to the number of leaves connected to the root r of the ontology, i.e., $\zeta_{\mathrm{leaf}} = \zeta_r$. Note that $|\mathcal{A}_r| = 1$ to avoid $\log(0)$ when dealing with the root term.

Again the consideration of the root in the conception of an IC model can bias IC values computed and in the context of the Zánchez et al. IC model, the model can be redefined by assessing adding 1 or not as for Zhang et al. and Seco et al. IC models. Thus, the IC value of the term x can be redefined as follows:

$$\mathrm{IC}(x) = -\log\left(\frac{\zeta_x}{|\mathcal{A}_x| * \zeta_{\mathrm{leaf}}}\right) \tag{8}$$

but, this needs to be evaluated on experimental data for validation.

1.8 Meng et al. IC model

In attempting to fix the inability of a model to effectively capture the term specificity without depending on a tuning factor, Meng et al. [9] also introduced another model, in which the IC value

of a term x is given by:

$$\mathrm{IC}_{\mathrm{Meng}}(x) = \frac{\log(\delta(x))}{\log(\delta_{\max})} * \left(1 - \frac{\log\left(\sum_{t \in D_x} \frac{1}{\delta(t)} + 1\right)}{\log(\lambda_{\mathrm{node}})} \right) \tag{9}$$

where λ_{node} is the number of nodes (terms) in the ontology as defined in equation (6).

1.9 Wang et al. IC model

Wang et al. [2] introduced a topology-based semantic similarity measure in which the semantic value of a given term x is computed using an S-value S_x related to the term x, and given by

$$S_x(t) = \begin{cases} 1 & \text{if } t = x \\ \max\{\omega_e * S_x(t') : t' \in \mathcal{C}_h(t)\} & \text{otherwise} \end{cases} \tag{10}$$

with $\mathcal{C}_h(t)$ the set of children of the term t, and ω_e the semantic contribution factor for 'is_a' and 'part_a' relations set to 0.8 and 0.6, respectively. The information content or a semantic value of a term x is calculated as follows:

$$\mathrm{IC}(x) = \sum_{t \in \mathcal{A}_x} S_x(t) \tag{11}$$

In this case, the IC scores of the three roots are 1, the lowest term IC value.

2 Ontology concept semantic similarity approaches

Broadly speaking, there exist two or three main classes of ontology concept similarity approaches, namely edge- (or path-), information content-based (or node-based) and 'hybrid' edge-node based approaches. The next sections attempt to provide an exhaustive review of these different classes of term similarity measures. Note, as for IC models, most of the approaches used in the context of GO were suggested in WordNet and adapted to be used for GO.

2.1 IC- or node-based concept semantic similarity approaches

Several approaches have been proposed for computing term semantic similarity scores within the ontology DAG, especially in the context of annotation-based approaches. These approaches include Resnik [37], Lin [38], Nunivers, Jiang& Conrath [36] and several other corrections, such as Graph-based Similarity (Disjunct Common Ancestors [11], known as GraSM, Exclusively Inherited Shared Information, referred to as EISI [39], eXtended GraSM, denoted XGraSM [1, 10], and Aggregate Information Content, referred to as AIC [14]), relevance similarity by Schlicker et al. [19] and information coefficient similarity by Li et al. [13], have been proposed in order to improve existing ontology concept comparison approaches.

2.1.1 Resnik, Lin, Nunivers, FaITH and P&S approaches

– For Resnik [37], the similarity between two terms is the information content of their most informative common ancestor (MICA), given by the following formula:

$$\mathcal{S}_{\mathrm{r}}(a,b) = \mathrm{IC}(c) = \max\left\{\mathrm{IC}(x) : x \in \mathcal{A}_a \cap \mathcal{A}_b\right\} \tag{12}$$

where c is MICA between terms a and b.

– The Lin semantic similarity approach [38] takes MICA between terms being compared and normalized by the average of IC values of these terms. Thus, the similarity between two terms is given by:

$$\mathcal{S}_{\mathrm{l}}(a,b) = \frac{2 * \mathrm{IC}(c)}{\mathrm{IC}(a) + \mathrm{IC}(b)} \tag{13}$$

Note that the Lin approach produces scores ranging between 0 and 1, and satisfies the property that the semantic similarity score between a term and itself is 1, but that is not the case for the Resnik approach. So, two strategies were suggested to scale these scores between 0 and 1 [1], one using either the possible upper bound of IC values [21], referred to as the Nunif strategy, and another one using the highest IC score in the ontology under consideration [22], referred to as the Nmax strategy, given by:

$$\mathcal{S}_r(a,b) = \begin{cases} \dfrac{\mathrm{IC}(c)}{\log_2 N} & \text{for Nunif} \\ \\ \dfrac{\mathrm{IC}(c)}{\mathrm{IC}_{\max}} & \text{for Nmax} \end{cases} \tag{14}$$

where N is the number of annotated proteins in the corpus under consideration and $\mathrm{IC}_{\max}$ the highest IC score in the ontology considered. The DaGO-Fun tool uses the Nmax model, which showed better performance than the Nunif model, for the Resnik approach.

– The Nunivers approach [1] has been proposed to satisfy the requirement that the semantic similarity score between a term and itself should be 1 by normalizing the score by the maximum IC values of terms and given by:

$$\mathcal{S}_{\mathrm{n}}(a,b) = \frac{\mathrm{IC}(c)}{\max\{\mathrm{IC}(a), \mathrm{IC}(b)\}} \tag{15}$$

– The FaITH approach [40] computes the semantic similarity score, $\mathcal{S}_{\mathrm{FaITH}}(a,b)$, as follows:

$$\mathcal{S}_{\mathrm{FaITH}}(a,b) = \frac{\mathrm{IC}(c)}{\mathrm{IC}(a) + \mathrm{IC}(b) - \mathrm{IC}(c)} \tag{16}$$

This approach is just an adaptation of the edge-based semantic similarity score introduced by Stojanovic et al. [31] (see equation (36)).

– The P&S approach [41, 42] suggests the semantic similarity score, $\mathcal{S}_{\mathrm{P\&S}}(a,b)$, to be computed as

shown below[1]:

$$\mathcal{S}_{\mathrm{P\&S}}(a,b) = \begin{cases} 1 & \text{if } a = b \\ 3 * \mathrm{IC}(c) - \mathrm{IC}(a) - \mathrm{IC}(b) & \text{otherwise.} \end{cases} \tag{17}$$

2.1.2 Improving Scores: Relevance, SimIC, GraSM, EISI, XGraSM and AIC

Some correction factors were proposed to deal with the issue of biases observed when estimating similarity score in the context of annotation-based approaches. We quote the Relevance similarity measure introduced by Schlicker et al. [19] and the Information Coefficient, often referred to as SimIC, suggested by Li et al. [13] in the context of the Lin approach, and the Graph-based similarity measure, i.e., GraSM [11], EISI [39], XGraSM [1, 10] and AIC [14], which can be applied to any annotation-based term semantic similarity approaches. In this case, the similarity score $\mathcal{S}(a,b)$ between terms a and b is weighted by a correction factor ϵ, i.e., the corrected score $\mathcal{S}_f(a,b)$ is given by

$$\mathcal{S}_{\mathrm{f}}(a,b) = \epsilon * \mathcal{S}(a,b) \tag{18}$$

with

$$\epsilon = \begin{cases} 1 - \exp\big(-\mathrm{IC}(c)\big) & \text{for Relevance} \\ 1 - \big(1 + \mathrm{IC}(c)\big)^{-1} & \text{for SimIC} \\ \dfrac{1}{n}\left(1 + \displaystyle\sum_{j=1}^{n-1} \frac{\mathrm{IC}(t_j)}{\mathrm{IC}(c)}\right) & \text{for Graph-based: GraSM, EISI and XGraSM} \end{cases} \tag{19}$$

with n the number of disjunctive (for GraSM), exclusively inherited (for EISI) or all informative (for XGraSM) common ancestors between terms a and b, the n^{th} ancestor term being the most informative common ancestor (MICA) between a and b, i.e., the common ancestor with the highest IC value. For two given terms s and t, their set of disjunctive, exclusively inherited and all informative common ancestors, denoted $\mathrm{DCA}(s,t)$, $\mathrm{EICA}(s,t)$ and $\mathrm{ICA}(s,t)$, respectively, are defined as follows:

$$\begin{aligned} \mathrm{DCA}(s,t) &= \big\{a \in \mathcal{A}_s \cap \mathcal{A}_t : \forall c \in \mathcal{A}_s \cap \mathcal{A}_t \text{ and } \mathrm{IC}(a) < \mathrm{IC}(c) \Rightarrow (a,c) \in \mathrm{DA}(s) \cup \mathrm{DA}(t)\big\} \\ \mathrm{EICA}(s,t) &= \big\{a \in \mathcal{A}_s \cap \mathcal{A}_t : \mathcal{C}_h(a) \cap \big((\mathcal{A}_s \cup \mathcal{A}_t) - (\mathcal{A}_s \cap \mathcal{A}_t)\big) \neq \varnothing\big\} \\ \mathrm{ICA}(s,t) &= \big\{a \in \mathcal{A}_s \cap \mathcal{A}_t : \mathrm{IC}(a) > 0\big\} \end{aligned} \tag{20}$$

where $\mathcal{C}_h(x)$ is the set of ontology concepts having x as a parent and $\mathrm{DA}(x)$ the disjunctive ancestors of the term x with $\mathrm{DA}(x) = \big\{(a,b) \in \mathcal{A}_x \times \mathcal{A}_x : \exists p \in \mathrm{P}_{a-x}$ such that $b \notin S_p$ and $\exists p \in \mathrm{P}_{b-x}$ such that $a \notin S_p\big\}$, P_{d-x} being the set of paths from x to d and S_p the set of terms on the path p.

It is worth mentioning that XGraSM and EISI have been shown to outperform the GraSM approach [1] and finding the disjunctive common ancestors (DCA) between two ontology conceptd makes the original GraSM approach computationally unattractive. Unfortunately, this computational complexity is not proportional to the improvement in performance, and thus, this approach is

[1] In order to prevent the violation of the non-negativity property of semantic similarity measures, this P&S term semantic similarity formula can be defined as follows:

$$\mathcal{S}_{\mathrm{P\&S}}(a,b) = \begin{cases} 1 & \text{if } a = b \\ \max\big\{0, 3 * \mathrm{IC}(c) - \mathrm{IC}(a) - \mathrm{IC}(b)\big\} & \text{otherwise.} \end{cases}$$

not included in most of currently used ontology semantic similarity tools [10]. Note that a random walks enhancement [15] was also proposed to improve any of the existing similarity measures by modeling inherent uncertainty from the incomplete knowledge of gene annotations and ontology structure, and this enhancement is implemented in the GOssTo tool [58].

Recently, Aggregate Information Content (AIC) approach [14] was introduced and computes the semantic value SV (x) of the GO term x by aggregating semantic contribution of the term ancestors, which is given by:

$$\mathrm{SV}\,(x) = \sum_{t \in \mathcal{A}_x} S_\omega\,(t) \tag{21}$$

where $S_\omega\,(t)$ is the semantic weight of the ontology concept t and calculated as follows:

$$S_\omega\,(t) = \frac{1}{1 + \exp\left(-\frac{1}{\mathrm{IC}(t)}\right)} \tag{22}$$

Thus, the semantic similarity between ontology concepts a and b is defined based on aggregate information content scores of common ancestors between terms a and b, and given by:

$$\mathcal{S}_{\mathrm{AIC}}\,(a,b) = \sum_{t \in \mathcal{A}_a \cap \mathcal{A}_b} \frac{2 * S_\omega(t)}{\mathrm{SV}(a) + \mathrm{SV}(b)} \tag{23}$$

2.1.3 Wang, Zhang and GO-universal approaches

In the case of topology-based models, each approach was set with a specific term semantic similarity approach, except for the Zhang et al. approach, which is a context dependent method, often implemented with the Lin-like term semantic similarity approach [1, 10] as it has been shown to perform better with the Lin-approach [5]. Thus, for the Zhang et al approach, the semantic similarity between two terms is given by

$$\mathcal{S}_z\,(a,b) = \frac{2 * \mathrm{IC}(c)}{\mathrm{IC}(a) + \mathrm{IC}(b)} \tag{24}$$

The GO-universal approach uses the Nunivers normalization model and calculates the similarity score as follows:

$$\mathcal{S}_{\mathrm{u}}\,(a,b) = \frac{\mathrm{IC}(c)}{\max\{\mathrm{IC}(a), \mathrm{IC}(b)\}} \tag{25}$$

Finally, for the Wang et al approach, the semantic similarity between two terms is given by:

$$\mathcal{S}_w\,(a,b) = \sum_{t \in \mathcal{A}_a \cap \mathcal{A}_b} \frac{S_a(t) + S_b(t)}{\mathrm{IC}(a) + \mathrm{IC}(b)} \tag{26}$$

where S_x is the S-value related to the term x as defined previously.

2.2 Edge-based concept semantic similarity approaches

The edge-based approach is the oldest approach that was proposed for measuring similarity between terms in a hierarchical semantic structure. In this approach, semantic similarity between two terms is a function of the number of edges (or nodes) on a path between these terms. In this traditional approach, the shorter the length of the shortest path, the more semantically similar the two terms are.

2.2.1 Rada et al. based approach

This approach is based on the shortest distance [20] in terms of number of edges or links between the two terms under consideration. Therefore, the semantic similarity, $\mathcal{S}_{\text{sp}}(a,b)$, between a and b, is actually the inverse multiplicative of the length of the shortest path, $\mathcal{D}_{\text{sp}}(a,b)$, between terms increased by 1 to prevent a division by 0. Thus, $\mathcal{S}_{\text{rada}}(a,b)$ is quantified as:

$$\mathcal{S}_{\text{rada}}(a,b) = \frac{1}{1+\mathcal{D}_{\text{sp}}(a,b)} \tag{27}$$

2.2.2 Resnik edge-based approach

In this case, the shortest distance, $D_{sp}(a,b)$, between a and b is converted to a semantic similarity $\mathcal{S}_{\text{re}}(a,b)$ by simply subtracting it from the maximum possible path length [24] in the ontology, i.e.,

$$\mathcal{S}_{\text{re}}(a,b) = 2*\delta_{\max} - D_{sp}(a,b) \tag{28}$$

This normalized version of the Resnik edge-based approach can be obtained by dividing each term on the right side of the equation (28) by the possible maximum value, which is $2*\delta_{\max}$. Thus, this normalized Resnik edge-based semantic similarity approach is given by:

$$\mathcal{S}_{\text{nre}}(a,b) = 1 - \frac{\mathcal{D}_{sp}(a,b)}{2*\delta_{\max}} \tag{29}$$

2.2.3 Leacock & Chodorow approach

Similarly to the Resnik edge-based approach, Leacock and Chodorow [25] rather suggested the use of a non-linear function 'log' to convert the shortest distance, $\mathcal{D}_{sp}(a,b)$, to semantic similarity score $\mathcal{S}_{\text{lc}}(a,b)$. Thus, the similarity score between two terms is the negative logarithm of the ratio between the length of the shortest path and twice the maximum depth of the hierarchy under consideration, i.e., $\mathcal{S}_{\text{lc}}(a,b)$ is given by:

$$\mathcal{S}_{\text{lc}}(a,b) = -\log\left(\frac{\mathcal{D}_{\text{sp}}(a,b)}{2*\delta_{\max}}\right) = \log(2*\delta_{\max}) - \log(\mathcal{D}_{\text{sp}}(a,b)) \tag{30}$$

As for the normalized Resnik edge-based approach in the expression (29), scaling the score to the unit interval by dividing each operand in the equation above by $\log(2*\delta_{\max})$, we obtain:

$$\mathcal{S}_{\text{nlc}}(a,b) = 1 - \frac{\log(\mathcal{D}_{\text{sp}}(a,b))}{\log(2*\delta_{\max})} \tag{31}$$

which represents a normalized version of the Leacock & Chodorow approach, i.e., with scores ranging between 0 and 1.

2.2.4 Wu & Palmer approach

Equation (32) describes the Wu & Palmer [23] term semantic similarity score suggested in the context of edge- or path-based approaches. This is given by

$$\begin{aligned}\mathcal{S}_{\text{wp}}(a,b) &= \frac{2*\delta(c)}{\text{len}(a,c)+\text{len}(b,c)+2*\delta(c)} = \frac{2*\delta(c)}{\text{len}_c(a,b)+2*\delta(c)} \\ &= \frac{2*\delta(c)}{\delta(a)+\delta(b)}\end{aligned} \tag{32}$$

where c is the LCA of terms a and b that yields the largest value of $\mathcal{S}_{\text{wp}}(a,b)$.

2.2.5 Slimani et al. and Shenoy et al. approaches

In order to correct biases produced by the Wu & Palmer approach due to the fact that it does not consistently consider the semantic relation between terms in the ontology, leading to inadequate and unrealistic scores (overestimation) in the context of semantic information retrieval, Slimani et al. [27] and Shenoy et al. [28] introduce correction factors or weights in order to penalize different scores produced. This correction factor, $\text{CF}(a,b)$, depends on the shortest length between terms, a and b, and depth of the ontology for Shenoy et al. or on parameters related to terms, such as depth of terms and that of their LCA, for Slimani et al. Thus, the semantic similarity score between two terms a and b is calculated as follows:

$$\mathcal{S}_{\text{sli}}(a,b) = \text{CF}(a,b) * \mathcal{S}_{\text{wp}}(a,b) \tag{33}$$

with $\mathcal{S}_{\text{wp}}(a,b)$ the Wu & Palmer semantic similarity score in the expression (32) and the correction factor, $\text{CF}(a,b)$, is given by:

$$\text{CF}(a,b) = \begin{cases} (1-\lambda) * \left(\min\left\{\delta(a),\delta(b)\right\} - \delta(c)\right) + \lambda * \left(\left|\delta(a)-\delta(b)\right|+1\right)^{-1} & (1) \\ \exp\left(-\frac{\lambda * D_{\text{sp}}(a,b)}{\delta_{\max}}\right) & (2) \end{cases} \tag{34}$$

(1) is the correction factor that has been suggested by Slimani et al., and (2) is that suggested by Shenoy et al. The switching parameter λ is a Boolean value, i.e., $\lambda = 0$ or 1, with 0 for two terms in the same hierarchy and 1 for neighborhood two terms, respectively.

2.2.6 Pekar & Staab approach

Another path-based concept semantic similarity score between a and b as used in [29, 30], known as the Pekar & Staab approach, is captured by equation (35) below:

$$\mathcal{S}_{\text{ps}}(a,b) = \frac{\delta(c)}{\delta(c)+\text{len}(a,c)+\text{len}(b,c)} = \frac{\delta(c)}{\delta(c)+\text{len}_c(a,b)} \tag{35}$$

2.2.7 Stojanovic et al. approach

The Stojanovic approach [31] is a path-based term semantic similarity score between a and b given by the following formula:

$$\mathcal{S}_{\text{ps}}(a,b) = \frac{\delta(c)}{\delta(a)+\delta(b)-\delta(c)} \tag{36}$$

It is worth noting that this approach is just a mathematical simplification of the Pekar and Staab approach in equation (35), so there is no difference between the two approaches.

2.2.8 Wang edge-based approach

Wang et al. [26] also introduced the semantic similarity approach that averages path lengths from the root of the ontology to all lowest common ancestors. So, the semantic similarity between two terms a and b is obtained as follows:

$$\mathcal{S}_{\text{we}}(a,b) = \frac{1}{|\text{LCA}(a,b)|} \sum_{c \in \text{LCA}(a,b)} \frac{\overline{\text{len}(r,c)}^2}{\overline{\text{len}_c(r,a)} * \overline{\text{len}_c(r,b)}} \tag{37}$$

2.2.9 Zhong et al. approach

For the Zhong et al. approach [32], the semantic distance between terms a and b is computed between different concepts from their positions and that of their LCA in the hierarchy, given by their mile-stone values in the hierarchy defined as:

$$m(x) = \frac{1/2}{k^{\delta(x)}} \tag{38}$$

for a term x, with $k > 1$ a contribution factor controlling the effect of the term in the ontology. In most cases, $k = 2$. The distance between two terms a and b with c as their LCA is given by:

$$\begin{aligned} \mathcal{D}_{\text{zh}}(a,b) &= \mathcal{D}_{\text{zh}}(a,c) + \mathcal{D}_{\text{zh}}(b,c) = \big(m(c) - m(a)\big) + \big(m(c) - m(b)\big) \\ &= 2 * m(c) - m(a) - m(b) \\ &= \frac{1}{k^{\delta(c)}} - \frac{1/2}{k^{\delta(a)}} - \frac{1/2}{k^{\delta(b)}} \end{aligned} \tag{39}$$

Thus, the semantic similarity score between two terms a and b is given by:

$$\mathcal{S}_{\text{zh}} = 1 - \mathcal{D}_{\text{zh}}(a,b) \tag{40}$$

2.2.10 Al-Mubaid & Nguyen approach

This approach computes the semantic distance score, $\mathcal{D}_{\text{na}}(a,b)$ between a and b by combining the shortest distance, $\mathcal{D}_{sp}(a,b)$, the depth of the ontology, $\delta_{\max}$, and that of their lowest common ancestor, $\delta(c)$, in a non-linear log fashion [17] as follows:

$$\mathcal{D}_{\text{aln}}(a,b) = \log\Big(k + (\mathcal{D}_{sp}(a,b) - 1)^{\alpha} * (\delta_{\max} - \delta(c))^{\beta}\Big) \tag{41}$$

where $\alpha > 0$ and $\beta > 0$ are contribution factors controlling the importance of path length and common specificity features, and k is a constant. Nguyen & Al-Mubaid showed that $\mathcal{D}_{\text{na}}(a,b)$ is positive if $k \geq 1$ and in their experiment, they set $k = 1$ and assigned an equal weight of 1 to the contribution factors α and β, i.e., $\alpha = \beta = 1$.

2.2.11 Li et al. edge-based approach

This approach [18] also combines the shortest path and the depth of terms using a non-linear function, 'tanh', under the assumption that information sources are infinite and are being transformed to a bounded interval between completely similar and nothing similar. Thus, this transformation is non-linear and the semantic similarity score, $\mathcal{S}_{\text{le}}(a,b)$, is given by:

$$\mathcal{S}_{\text{le}}(a,b) = \exp\left(-\alpha * \mathcal{D}_{sp}(a,b)\right) * \tanh\left(\beta * \delta(c)\right) \tag{42}$$

where $\alpha \geq 0$ and $\beta > 0$ are are shortest path length and depth contribution scaling parameters, respectively. The authors suggested setting $\alpha = 0.2$ and $\beta = 0.6$, as these values provided good performance for an empirical finding in a specific setting. However, it lacks a theoretical basis, cannot be generalized, and more importantly it is not clear for which values of these semantic factors the semantic similarity measure yields the optimal value of biological content of terms.

2.3 "Hybrid" concept semantic similarity approaches

Edge-based term semantic similarity approaches are limited to edge counting and fail to take into account the positions of terms expressing their specificity in the hierarchy. In order to attenuate this shortcoming, some researchers weighted edges by assigning lower weight to edges at the lower level (close to the root) compared to edges at a higher level in the hierarchy. However, terms at the same depth do not necessarily have the same specificity, and edges at the same level do not necessarily represent the same semantic distance [33]. These approaches are being used in the context of categories of term semantic similarity approaches refer to as "hybrid" approaches. These approaches combine several structural characteristics (such as path length, depth, etc.) and assign weights to terms along these paths, very often using their IC values and other correction factors in order to balance the contribution of different components to the final similarity score. Note that even though these approaches have shown better performance for a concrete scenario or specific case than more basic edge-based measures, this performance often depends on the empirical tuning of weights and correction factors according to the ontology and input terms.

2.3.1 Relative Specificity Similarity (RSS) approach

The relative specificity similarity (RSS) score [34], $\mathcal{S}_{\text{rss}}$ between the two terms a and b of a given ontology, is quantified as follows:

$$\mathcal{S}_{\text{rss}}(a,b) = \frac{\delta_{\max}}{\delta_{\max} + \mathcal{D}_{\text{sp}}(a,b)} * \frac{\alpha}{\alpha + \beta} \tag{43}$$

where α and β are tuning parameters with α weighing the specificity of the lowest common ancestor, c, of terms a and b, which should obviously depend on $\delta(c)$, and β weighing the generality of terms a and b, defined as the minimum path length from the term under consideration to the leaf terms connected to it.

2.3.2 Hybrid Relative Specificity Similarity (HRSS) approach

The hybrid relative specificity similarity (HRSS) approach [34] is similar to the RSS approach, but it weighs the specificity and the generality of a term using the IC concept. This approach computes

the term semantic similarity score, $\mathcal{S}_{\text{hrss}}$ between the two terms a and b of a given ontology as follows:

$$\mathcal{S}_{\text{rss}}(a,b) = \frac{1}{1+\mathcal{D}_{\text{ic}}(a,b)} * \frac{\alpha_{\text{ic}}}{\alpha_{\text{ic}}+\beta_{\text{ic}}} \tag{44}$$

where $\mathcal{D}_{\text{ic}}(a,b) = \mathcal{D}_{\text{ic}}(a,c) + \mathcal{D}_{\text{ic}}(b,c)$ with c the MICA between terms a and b and the IC-distance $\mathcal{D}_{\text{ic}}(x,c)$ between a term x and its ancestor c is given by[2]:

$$\mathcal{D}_{\text{ic}}(c,x) = |\text{IC}(x) - \text{IC}(c)| \tag{45}$$

and thus, α_{ic} expressing the specificity of the MICA, c, between a and b is given by:

$$\alpha_{\text{ic}} = \mathcal{D}_{\text{ic}}(r,c) = |\text{IC}(c) - \text{IC}(r)| = \text{IC}(c) \tag{46}$$

where r is the root of the ontology. The IC-based or semantic generality of a term is defined as the $\mathcal{D}_{\text{ic}}$ value between the term and the most informative leaf term connected to it. For this specific approach, β_{ic}, which weighs the generality of terms a and b, was set as the average between IC-based generality values of these two terms, i.e.,

$$\beta_{\text{ic}} = \frac{\mathcal{D}_{\text{ic}}(a,\ell_a) + \mathcal{D}_{\text{ic}}(b,\ell_b)}{2} \tag{47}$$

where ℓ_x is the most informative leaf term connected to the term x.

2.3.3 Shen et al. approach

The Shen et al. approach [35] computes the semantic distance between terms by summing IC term weights along the shortest path connecting each term to their MICA. So, the distance between two terms a and b is calculated as follows[3]:

$$\mathcal{D}_{\text{shen}}(a,b) = \frac{\arctan\left(\sum_{x\in S_{a-c}} \frac{1}{\text{IC}(x)} + \sum_{x\in S_{b-c}-\{c\}} \frac{1}{\text{IC}(x)}\right)}{\pi/2} \tag{48}$$

where c is the MICA between terms a and b, and S_{x-z} a set of terms along shortest paths connecting x to its ancestors z in terms of sum of multiplicative inverse term IC values. Since this distance

[2]In the corresponding paper, this semantic distance is defined as follows:

$$\mathcal{D}_{\text{ic}}(x,y) = \text{IC}(y) - \text{IC}(x)$$

However, as a distance it must be positive definite and symmetric. So, we have fixed the formula by redefining this distance as follows:

$$\mathcal{D}_{\text{ic}}(x,y) = |\text{IC}(y) - \text{IC}(x)|$$

This ensures that the measure is symmetric and positively defined.

[3]In the original manuscript, the equation defining this semantic distance is given by:

$$\mathcal{D}_{\text{shen}}(a,b) = \frac{\arctan\left(\sum_{x\in S_{a-c}} \frac{1}{\text{IC}(x)} + \sum_{x\in S_{b-c}} \frac{1}{\text{IC}(x)}\right)}{\pi/2}$$

As such, we assume that this formula raises some concerns, including (1) MICA of two terms in the formula will be considered twice and (2) the shortest path score (number of hops, edge weights, node weights, etc.) was not clearly or explicitly defined. So, we have rectified the formula defining this distance and defined the shortest path score.

is normalized, i.e., ranges between 0 and 1, the semantic similarity score between terms a and b is given by:

$$\mathcal{S}_{\text{shen}}(a,b) = 1 - \mathcal{D}_{\text{shen}}(a,b) \tag{49}$$

2.3.4 Shortest semantic differentiation distance (SSDD) approach

Instead of selecting a shortest path in terms of number of edges, Xu et al. [12] select a path with minimum sum weights among all paths connecting two terms under consideration via their lowest common ancestors and suggested the shortest semantic differentiation distance (SSDD) approach to compute the semantic or IC-based distance as follows[4]:

$$\mathcal{D}_{\text{ssdd}}(a,b) = \frac{\arctan\left(\min\limits_{p\in \text{SP}_{a-b}} \sum\limits_{c\in S_p} T(c)\right)}{\pi/2} \tag{50}$$

where S_p is a set of terms on the shortest path connecting the terms a and b via their lowest common ancestors in terms of number of hops or edges and SP_{a-b} is the set of all such paths. The function T is known as a T-value function and quantifies the semantic value that a term inherited from its ancestors and distributed to its descendants, computed as:

$$T(x) = \begin{cases} 1 & \text{if } x \text{ is a root} \\ \frac{1}{|\mathcal{P}_x|} \sum\limits_{t\in\mathcal{P}_x} \big(\omega * T(t)\big) & \text{otherwise} \end{cases} \tag{51}$$

The variable ω is the semantic differentiation factor for the edge linking term t with its parent x in a set of $\mathcal{P}_x$ of all parents of the term t and given by:

$$\omega = \frac{\left|D_{\text{t}}^{+}\right|}{\left|D_{\text{x}}^{+}\right|} \tag{52}$$

where $D_{\text{s}}^{+} = D_{\text{s}} \cup \{s\}$, with D_{s} the set of term hyponyms (descendants) to the term s.

2.3.5 General version of Jiang and Conrath approach

Jiang & Conrath [36] introduced a term semantic similarity approach in which density, depth, strength of connotation and information content of classes are taken into account. In this approach, the strength of association between a term s and its parent t, which represents the overall edge weight (wt), is defined as follows:

$$wt(s,t) = \left(\beta + (1-\beta) * \frac{\overline{C}}{\left|\mathcal{C}_h(t)\right|}\right) * \left(\frac{\delta(t)+1}{\delta(t)}\right)^{\alpha} * \big(\text{IC}(s) - \text{IC}(t)\big) * T(s,t) \tag{53}$$

[4] In the original paper, the equation defining this semantic distance is given by:

$$\mathcal{D}_{\text{ssdd}}(a,b) = \frac{\arctan\left(\min\left\{\sum\limits_{c\in S_p} T(c)\right\}\right)}{\pi/2}$$

As such, the min operator is redundant and can be removed. However, removing this min operator will make this distance ambiguous and not well defined as two terms in the ontology DAG can share several LCAs and there may be several shortest paths between two terms.

where $\left|\mathcal{C}_h\left(t\right)\right|$ is the number of children with term t as parent, which represents the local density, $\overline{\mathcal{C}}$ is the average density in the whole ontology structure, $T\left(s,t\right)$ the link type factor, and $\left(\mathrm{IC}\left(s\right)-\mathrm{IC}\left(t\right)\right)$ the link strength. The contribution factors α, with $\alpha \geq 0$ and $\beta \in [0,1]$ adjust the effect of term depth and density on the edge weight. The overall distance between two terms a and b is the summation of edge weights along the shortest path linking the two terms, as shown below:

$$\mathcal{D}_{\mathrm{JC}}\left(a,b\right) = \sum_{s \in S_{a-b}-\{c\}} wt\left(s, s_p\right) \tag{54}$$

where S_{a-b} is the set that contains all of the terms in the shortest path from a to b passing through c their LCA and s_p is the parent of s along the shortest path under consideration. The most used case, in which $\alpha = 0$, $\beta = 1 = T\left(s, s_p\right)$ for all $s \in S_{a-b} - \{c\}$, produces the simplified version of the Jiang and Conrath formula, which is an IC- or node-based approach, given by:

$$\mathcal{D}_{\mathrm{JC}}\left(a,b\right) = \mathrm{IC}\left(a\right) + \mathrm{IC}\left(b\right) - 2 * \mathrm{IC}\left(c\right) \tag{55}$$

– Using the Resnik edge-based version in the expression (28), one can transform $\mathcal{D}_{\mathrm{JC}}\left(a,b\right)$ to the semantic similarity score $\mathcal{S}_{\text{jc-re}}\left(a,b\right)$ as follows:

$$\mathcal{S}_{\text{jc-re}}\left(a,b\right) = 2 * \mathrm{IC}_{\max} - \mathcal{D}_{\text{jc}}\left(a,b\right) \tag{56}$$

with $\mathrm{IC}_{\max}$ the largest IC value in the ontology under consideration. Note that one can use the possible upper bound of IC values [21] in the ontology under consideration [22, 43] as $\mathrm{IC}_{\max}$, which is given by

$$\mathrm{IC}_{\max} = \log_2 N \tag{57}$$

with N the number of annotated proteins in the corpus under consideration.

Following the expression (29), the normalized Resnik edge-based Jiang & Conrath, is expressed as follows:

$$\mathcal{S}_{\text{jc-re}}\left(a,b\right) = 1 - \frac{\mathcal{D}_{\mathrm{JC}}\left(a,b\right)}{2 * \mathrm{IC}_{\max}} \tag{58}$$

which indicates that the normalized Jiang & Conrath distance, $d_{\mathrm{JC}}\left(a,b\right)$, between terms a and b is given by:

$$d_{\mathrm{JC}}\left(a,b\right) = \frac{\mathcal{D}_{\mathrm{JC}}\left(a,b\right)}{2 * \mathrm{IC}_{\max}} = \frac{\mathrm{IC}_n\left(a\right) + \mathrm{IC}_n\left(b\right)}{2} - \mathrm{IC}_n\left(c\right) \tag{59}$$

corresponding to the normalized Jiang & Conrath distance model suggested by Pesquita et al [21], where $IC_n\left(x\right)$ is the normalized IC score of x, given by

$$IC_n\left(x\right) = \frac{\mathrm{IC}\left(x\right)}{\mathrm{IC}_{\max}} \tag{60}$$

In the context of GO, another Jiang & Conrath normalization scheme was suggested by Couto et al [43] and defined as follows:

$$d_{\mathrm{JC}}\left(a,b\right) = \min\left\{1, \frac{\mathcal{D}_{\mathrm{JC}}\left(a,b\right)}{\mathrm{IC}_{\max}}\right\} \tag{61}$$

It is worth noting that by using canonical normalization of this distance, where the original distance is divided by the maximum possible distance between terms, which is $\mathrm{IC}\left(a\right) + \mathrm{IC}\left(b\right)$, leads to the Lin term semantic similarity approach, i.e.,

$$\mathcal{S}_L\left(a,b\right) = 1 - \frac{\mathcal{D}_{\mathrm{JC}}\left(a,b\right)}{\mathrm{IC}\left(a\right) + \mathrm{IC}\left(b\right)} \tag{62}$$

In other words, $\mathcal{D}_{JC}$ is simply the non-normalized distance derived from the Lin semantic similarity approach. The other normalization schemes were unable to improve the performance of the term semantic similarity approach inferred from the Jiang & Conrath distance [21]. This is why we are not referring to Jiang & Conrath, as the best semantic similarity measure inferred from this distance is Lin's approach.

– Following a similar view, Garla & Brandt [44] suggested the use of the Leacock & Chodorow normalization model in the relation (31) and defined the IC-based Leacock & Chodorow normalization model described below:

$$\mathcal{S}_{\text{jc-lc}}(a,b) = 1 - \frac{\log\left(D_{\text{JC}}(a,b)+1\right)}{\log\left(2 * \text{IC}_{\max}\right)} \tag{63}$$

in which 1 is added to $D_{\text{jc}}(a,b)$ to avoid having the logarithm of 0. In addition, to avoid negative semantic similarity score, Garla & Brandt suggested adding 1 to $2 * \text{IC}_{\max}$ within the log function in the denominator to get the following formula:

$$\mathcal{S}_{\text{jc-gb}}(a,b) = 1 - \frac{\log\left(D_{\text{JC}}(a,b)+1\right)}{\log\left(2 * \text{IC}_{\max}+1\right)} \tag{64}$$

It is important to know that one can use the canonical normalization scheme by using $\text{IC}(a)+\text{IC}(b)$ instead of $2 * \text{IC}_{\max}$ following the expression (62), which leads to the new term semantic similarity score, expressed as follows:

$$\mathcal{S}_{\text{jc-gbl}}(a,b) = 1 - \frac{\log\left(D_{\text{JC}}(a,b)+1\right)}{\log\left(\text{IC}(a)+\text{IC}(b)+1\right)} \tag{65}$$

– Finally, using the Rada et al. transformation of a distance to a similarity score, $D_{\text{JC}}(a,b)$ can be converted to semantic similarity score $\mathcal{S}_{\text{jc-ra}}(a,b)$ as follows:

$$\mathcal{S}_{\text{jc-ra}}(a,b) = \frac{1}{1+D_{\text{JC}}(a,b)} \tag{66}$$

with 1 added to avoid dividing by 0.

3 Entity semantic similarity measures

An entity is a set of ontology concepts essential in a domain, e.g., protein or gene which may be annotated by a set of GO terms since it can perform more than one biological function and be involved in several processes. Thus, a semantic similarity can be measured between sets of concepts associated with entities, for example, by using the IC values of their ontology concepts directly, referred to as group-wise measures, also known as direct term score based measures. Alternatively one can use concept semantic similarity scores of between concepts pair-wise, known as pairwise or term semantic-based or indirect term score based measures. These two types of measures obviously use the topology of the ontology and the concept of IC scores. Another type of measures exist and do not use the concept of IC values, referred to as edge-like based measures, these include SimALN, SimINT, SimLP and SimYE, described in the following sections. As these functional similarity measures require knowledge of the ontology under consideration, they are referred to as ontology-based measures. Finally, the last type of measures are those relying solely on the annotation dataset under consideration and do not consider the structure of the ontology or the

concept of IC scores. They are referred to as non ontology-based measures, and include Cho et al., Ali & Diane and Kappa-Statistics measures described in the following sections. In summary, we have categorized functional similarity measures into two main classes: Ontology-based and non ontology-based measures. The ontology-based measures are composed of two types of measures. The first type relies on the term IC values and the second uses edge counting approaches with each type being divided into two models, namely group-wise or direct term score based models and pairwise models, also known as term semantic-based or indirect term score based measures.

3.1 Pairwise concept semantic-based measures: Avg, Max, BMA, BMM, ABM and HDF

Pairwise measures combine the semantic similarity scores between concepts associated with entities or sets of ontology concepts using basic statistical measures of closeness (mean, max, min, etc.). These include Best-Match Average (BMA) [3, 21], Best Match Maximum (BMM) [19], Average Best-Matches (ABM) [2, 22], Average (Avg) [45], Maximum (Max) [46] and the topological clustering semantic similarity (TCSS). These measures are also known as term semantic-based (non direct) or pairwise (indirect) measures. In this category of measures, special measures derived from term distance scores from the Hausdorff (HDF) distance [47, 48, 49] have been suggested, used [50, 51] and implemented in several semantic similarity tools [52, 54, 55].

The average and maximum measures are computed as follows:

$$\mathrm{Avg}\,(p,q) = \frac{1}{n*m} \sum_{s\in T_p,\, t\in T_q} \mathcal{S}\,(s,t) \tag{67}$$

and

$$\mathrm{Max}\,(p,q) = \max\left\{\mathcal{S}\,(s,t) : s \in T_p \,\mathrm{and}\, t \in T_q\right\} \tag{68}$$

where T_r is a set of concepts in a given ontology, which can be, in the context of GO, the molecular function (MF), biological process (BP) or cellular component (CC) ontology annotating a given protein r and $n = |T_p|$ and $m = |T_q|$ are the number of GO terms in these sets. $\mathcal{S}\,(s,t)$ is the semantic similarity score.

The Topological Clustering Semantic Similarity (TCSS) measure is the particular case of the Max measure with a specific term semantic similarity score, $\mathcal{S}\,(s,t)$, calculated as follows:

$$\mathcal{S}\,(s,t) = \begin{cases} \mathrm{IC}^*_S\,(\mathrm{LCA}(s,t)) & \text{if } \mathcal{A}_s \subseteq \mathcal{A}_t \text{ or } \mathcal{A}_t \subseteq \mathcal{A}_s \\ \mathrm{IC}^*_M\,(\mathrm{LCA}(s,t)) & \text{otherwise} \end{cases} \tag{69}$$

where $\mathrm{IC}^*_X\,(t)$ represents the normalized IC score, $\mathrm{IC}_X\,(t)$, computed using the graph as provided from the GO database ($X = S$) or its collapsed version ($X = M$), referred to as a meta-graph, obtained by removing transitive term relationships [22]. This normalized IC score is given by:

$$\mathrm{IC}^*_X\,(t) = \frac{\mathrm{IC}_X(t)}{\max\limits_{s\in G_X} \mathrm{IC}_X(s)} \tag{70}$$

The BMA [3, 10] for entities p and q is the mean of the following two values: average of best matches of ontology concepts related to an entity p against those related to an entity q, and average of best

matches of concepts or terms soociated with entity q against those associated with entity p, given by the following formula:

$$\mathrm{BMA}\left(p,q\right)=\frac{1}{2}\left(\frac{1}{n}\sum_{s\in T_p}\mathcal{S}\left(s,T_q\right)+\frac{1}{m}\sum_{s\in T_q}\mathcal{S}\left(s,T_p\right)\right) \tag{71}$$

with $\mathcal{S}\left(s,T_r\right)=\max\left\{\mathcal{S}\left(s,t\right):t\in T_r\right\}$. It is important to note that the BMM measure, also known as RCMax (RowScore and ColumnScore Maximum) measure [57] implemented in the GOSemSim R package, takes the maximum values between them instead of the mean of these two values and is given by

$$\mathrm{BMM}\left(p,q\right)=\max\left\{\frac{1}{n}\sum_{s\in T_p}\mathcal{S}\left(s,T_q\right),\frac{1}{m}\sum_{s\in T_q}\mathcal{S}\left(s,T_p\right)\right\} \tag{72}$$

However, the performance of this measure has never been assessed and it is rarely used.

The ABM [10] for entities provided their associated concepts is the mean of best matches of concepts of each entity against the other, given by the following formula:

$$\mathrm{ABM}\left(p,q\right)=\frac{1}{n+m}\left(\sum_{s\in T_p}\mathcal{S}\left(s,T_q\right)+\sum_{s\in T_q}\mathcal{S}\left(s,T_p\right)\right) \tag{73}$$

Note ABM and BMA measures produce different scores and they are equal only when $n=m$, which is not often the case in a set of annotated genes or proteins.

A class of functional similarity measures was derived from the Hausdorff distance and used in the context of GO. The initial Hausdorff distance between proteins p and q is given by:

$$\mathrm{HDF}\left(p,q\right)=\max\left\{\max_{s\in T_p}\mathcal{D}\left(s,T_q\right),\max_{s\in T_q}\mathcal{D}\left(s,T_p\right)\right\} \tag{74}$$

where $\mathcal{D}\left(s,T_p\right)=\min\left\{\mathcal{D}\left(s,t\right):t\in T_p\right\}$, with $\mathcal{D}\left(s,t\right)$ the distance between terms s and t. It is clear that if the distance $\mathcal{D}\left(s,t\right)$ are normalized (ranging between 0 and 1), then the $\mathrm{HDF}\left(p,q\right)$ score also ranges between 0 and 1, and emphasizes the semantics closeness between entities p and q through shared terms between these two entities. If the two entities p and q share highly similar terms, in which case similarity scores $\mathcal{S}\left(s,t\right)$ are high for any $s\in T_p$ and $t\in T_q$, then distance scores $\mathcal{D}\left(s,t\right)=1-\mathcal{S}\left(s,t\right)$ will be low or close to 0 and consequently the distance score $\mathrm{HDF}\left(p,q\right)$ between p and q will also be low or close to 0. In this case, the functional similarity score between p and q, given by:

$$\mathcal{S}\left(p,q\right)=1-\mathrm{HDF}\left(p,q\right) \tag{75}$$

is high or close to 1. Because we are using semantic similarity scores rather than distances, to ease the computation of distance score between proteins we need to express $\mathcal{D}\left(t,T_p\right)$ in terms of semantic similarity scores. Thus,

$$\begin{aligned}\mathcal{D}\left(s,T_p\right)&=\min\left\{\mathcal{D}\left(s,t\right):t\in T_p\right\}\\&=\min\left\{1-\mathcal{S}\left(s,t\right):t\in T_p\right\}\\&=1-\max\left\{\mathcal{S}\left(s,t\right):t\in T_p\right\}\\&=1-\mathcal{S}\left(s,T_p\right)\end{aligned} \tag{76}$$

It follows that

$$\begin{aligned} \mathrm{HDF}\left(p,q\right) &= \max\left\{\max_{s\in T_p}\mathcal{D}\left(s,T_q\right),\max_{s\in T_q}\mathcal{D}\left(s,T_p\right)\right\} \\ &= \max\left\{\max_{s\in T_p}\left(1-\mathcal{S}\left(s,T_q\right)\right),\max_{s\in T_q}\left(1-\mathcal{S}\left(s,T_p\right)\right)\right\} \end{aligned} \tag{77}$$

Finally,

$$\mathrm{HDF}\left(p,q\right) = \max\left\{1-\min_{s\in T_p}\mathcal{S}\left(s,T_q\right),1-\min_{s\in T_q}\mathcal{S}\left(s,T_p\right)\right\} \tag{78}$$

It was noted that 24 different measures for object matching can be derived from the Hausdorff distance. Based on their behavior in the presence of noise, the best measure, called the modified Hausdorff distance (MHDF) for object matching, shown to be more robust to outliers [47], is given by:

$$\mathrm{MHDF}\left(p,q\right) = \max\left\{\frac{1}{n}\sum_{s\in T_p}\mathcal{D}\left(s,T_q\right),\frac{1}{m}\sum_{s\in T_q}\mathcal{D}\left(s,T_p\right)\right\} \tag{79}$$

In terms of semantic similarity scores, we can write:

$$\mathrm{MHDF}\left(p,q\right) = \max\left\{\frac{1}{n}\sum_{s\in T_p}\Big[1-\mathcal{S}\left(s,T_q\right)\Big],\frac{1}{m}\sum_{s\in T_q}\Big[1-\mathcal{S}\left(s,T_p\right)\Big]\right\} \tag{80}$$

Interestingly, the functional similarity derived from MHDF corresponds to the BMM measure defined above, eliciting the need for further assessment of this measure in the context of GO. Note that the BMA and ABM measures also match some variants of the HDF metric [47].

Another variant of the HDF distance, denoted VHDF, refers to a measure suggested by Lerman and Shakhnovich [50], and computes scores as follows:

$$\mathrm{VHDF}\left(p,q\right) = \frac{1}{2}\left(\sqrt{\frac{1}{n}\sum_{s\in T_p}\mathcal{D}^2\left(s,T_q\right)}+\sqrt{\frac{1}{m}\sum_{s\in T_q}\mathcal{D}^2\left(s,T_p\right)}\right) \tag{81}$$

It is worth mentioning that MHDF and VHDF measures do not define a metric or distance since they violate the triangle inequality property of a metric. For converting these distance scores to similarity scores, some research projects used exponential weight [54] and direct substitution [55] models, respectively, and given by:

$$\mathcal{S}\left(p,q\right) = \exp\big(-\mathrm{HDF}\left(p,q\right)\big) \text{ and } \mathcal{S}\left(p,q\right) = \frac{1}{2}\left(\sqrt{\frac{1}{n}\sum_{s\in T_p}\mathcal{S}^2\left(s,T_q\right)}+\sqrt{\frac{1}{m}\sum_{s\in T_q}\mathcal{S}^2\left(s,T_p\right)}\right) \tag{82}$$

The negative exponential weight model is used possibly to guarantee that semantic similarity axioms are conserved and to ensure the convergence, while the use of the direct substitution is possibly motivated by the fact that BMM, BMA and ABM measures directly match some variant of HDF based similarity measures [56], even though this is not the case for the specific variant it uses. However, it is recommended to normalize the distance scores, if not normalized, and use the linear transformation between distance $\mathcal{D}\left(p,q\right)$ and similarity $\mathcal{S}\left(p,q\right)$ scores given by [3]:

$$\mathcal{S}\left(p,q\right) = 1-\mathcal{D}\left(p,q\right) \tag{83}$$

to convert distance scores to semantic similarity scores.

3.2 Pairwise Edge-like measures: Al-Mubaid & Nagar, IntelliGO and spgk measures

– The Al-Mubaid & Nagar entity semantic similarity measure computes semantic similarity score, $\mathrm{SimALN}(p,q)$, between two entities, p and q, using the concept of shortest path lengths between pairwise ontology terms associated with the two entities under consideration [59, 60]. $\mathrm{SimALN}(p,q)$ is an exponential function of the additive inverse of mean shortest path lengths between terms aassoiated with entities p and q, given by:

$$\mathrm{SimALN}(p,q) = \exp\left(-\alpha * \mathcal{D}_{\mathrm{avg}}(p,q)\right) \tag{84}$$

where α is a contribution scaling factor of the average shortest path lengths, $\mathcal{D}_{\mathrm{avg}}(p,q)$, into the entity semantic similarity score, calculated as follows:

$$\mathcal{D}_{\mathrm{avg}}(p,q) = \frac{1}{n*m} \sum_{s\in T_p,\, t\in T_q} \mathcal{D}_{\mathrm{sp}}(s,t) \tag{85}$$

with $\mathcal{D}_{\mathrm{sp}}(s,t)$ the shortest distance between terms s and t.

– The IntelliGO measure, introduced by Benabderrahmane et al. [61] integrates several complementary properties in the vector space model with a non orthogonal basis $\{e_1, e_2, \ldots, e_n\}$, i.e., $e_\imath * e_\jmath$ is not always equal to 0 for $\imath \neq \jmath$, for $\imath, \jmath = 1, \ldots, n$, where the dimension n of the space is the number of terms occurring in the entity associted concept dataset under consideration. The IntelliGO semantic similarity score, $\mathrm{SimINT}(p,q)$, between the two entities p and q, is calculated using the Cosine similarity scheme under the usual normalization model and defined as:

$$\mathrm{SimINT}(p,q) = \frac{\langle \overline{p}, \overline{q} \rangle}{\|\overline{p}\| \|\overline{q}\|} \tag{86}$$

where $\overline{\omega}$ is the vectorial representation of the protein ω, given by:

$$\overline{\omega} = \sum_{\imath=1}^{n} \omega_\imath * e_\imath \tag{87}$$

with $\omega_\imath = \sigma(\omega, s_\imath) * \Gamma(s_\imath)$ the weight coefficient associated with the term $s_\imath$ for the entity ω, $\sigma(\omega, s_\imath)$ representing the weight component assigned to the evidence code s_i as a concept associated with the entity ω. $\Gamma(s_\imath)$ is the inverse annotation frequency (IAF) of the term s_i based on its occurrence frequency in the annotation dataset under consideration and computed as follows [5]:

$$\Gamma(t) = \log\left(\frac{\gamma_n}{\gamma(t)}\right) \tag{88}$$

[5] We think that this formula of IAF written as initially defined in the entity semantic similarity score is mathematically inconsistent, not realistic and violates axioms of a semantic similarity measure. To test this, one can just consider a annotation dataset with only two proteins annotated by one and the same term. Instead of getting the score 1 as similarity between these two entities, using IAF as originally set, one gets the entity semantic similarity score of 0, which is not correct. Thus, this IAF should rather be the relative annotation frequency, i.e.,

$$\Gamma(t) = \frac{\gamma(t)}{\gamma_n}$$

Even intuitively, this holds because if a term associated with many entities, it is more likely to contribute to the similarity shared between entities in the dataset depending on its evidence code for the associated entity in its weight coefficient.

with $\gamma(t)$ the number of entities related to the term t and γ_n the total number of entities in the dataset. The dot product $\langle \overline{p}, \overline{q} \rangle$ is given by:

$$\langle \overline{p}, \overline{q} \rangle = \sum_{\imath=1}^{n} \sum_{\jmath=1}^{n} (p_\imath * q_\jmath) * (e_\imath * e_\jmath) \tag{89}$$

where $e_\imath * e_\jmath = \mathcal{S}_{\text{wp}}(s_\imath, s_\jmath)$ is the Wu & Palmer term semantic similarity score between terms $s_\imath$ and $s_\jmath$. Note that $\|\overline{\omega}\| = \sqrt{\langle \overline{w}, \overline{w} \rangle}$. In the context of the PySML interface, these weight coefficients are simplified and set to 1 or 0, depending on whether the concepts is associed with the entity or not.

– The Shortest Path Graph Kernel (spgk) measure is an adaptation of a measure from [62] in the context of GO by Alvarez et al. [63] and uses the shortest path graphs G^p_{sp} and G^q_{sp} of the two proteins p and q under consideration to compute the functional similarity score between proteins. For a given protein ω, $G^\omega_{\text{sp}} = (\mathcal{A}_\omega, E_\omega)$ with the length of an edge $(s,t) \in E_\omega$ corresponding to the path shortest distance $\mathcal{D}_{\text{sh}}(s,t)$ from a term t to its ancestor s in the GO DAG, and the entity semantic similarity score $\text{SimSPGK}(p,q)$ between p and q is calculated as follows:

$$\text{SimSPGK}(p,q) = \sum_{e \in E_p, f \in E_q} k_{\text{walk}}(e,f) \tag{90}$$

where k_{walk} is a positive definite kernel for comparing two paths (walks) in the GO DAG and given by:

$$k_{\text{walk}}(e,f) = k_{\text{node}}(s_1, s_2) * k_{\text{edge}}(e,f) * k_{\text{node}}(t_1, t_2) \tag{91}$$

with $e = (s_1, t_1)$ and $f = (s_2, t_2)$ edges (or shortest paths) connecting term $t_\imath$ to its ancestor $s_\imath$, $\imath = 1, 2$, in the shortest path graphs G^p_{sp} and G^q_{sp} of the two proteins p and q (or in the GO DAG), respectively, k_{node} a kernel function comparing two terms a and b, which is a term-indicator function, i.e.,

$$k_{\text{node}}(a,b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases} \tag{92}$$

and k_{edge} called Brownian bridge kernel, given by:

$$k_{\text{edge}}(e,f) = \max\{0, c - |\text{len}(e) - \text{len}(f)|\} \tag{93}$$

returning the largest value when two edges have identical length, and 0 when the edges differ in length more than a constant c. This constant c is as initially set, i.e., $c = 2$, and note that scores produced by this measure are not normalized and for a given edge $d = (s,t)$ in a shortest path graph, $\text{len}(d) = \mathcal{D}_{\text{sh}}(s,t)$ in the GO DAG.

3.3 Group-wise Concept-based measures: SimGIC, SimDIC, SimUIC and Cosine

In general, these statistical measures of closeness are known to be sensitive to scores that lie at abnormal distances from the majority of scores, or outliers. This means that these measures may produce biases which affect functional similarity scores [3]. Thus, other functional similarity measures, such as SimGIC [21], SimDIC, SimUIC [3, 10, 64] and Cosine [33, 65], which use the IC of terms directly to compute functional similarity scores from their GO annotations, were introduced. SimGIC, SimDIC and SimUIC use the Jaccard index [67], but the Cosine measure uses a normalized dot product to estimate functional similarity scores.

The SimGIC measure computes the semantic similarity score between two entities p and q as follows:

$$\text{SimGIC}(p,q) = \frac{\sum\limits_{x \in \mathcal{A}_p \cap \mathcal{A}_q} \text{IC}(x)}{\sum\limits_{x \in \mathcal{A}_p \cup \mathcal{A}_q} \text{IC}(x)} \tag{94}$$

where IC(x) is the information content value of the term x [1] and $\mathcal{A}_r$ a set of GO terms together with their informative ancestors in a given ontology (MF, BP or CC for GO) associated with a given protein r.

Two other entity semantic similarity measures [3, 10], using Dice (Czekanowski or Lin like measure) and universal indexes, referred to as SimDIC and SimUIC, respectively, are given by the following formulae:

$$\text{SimDIC}(p,q) = \frac{2 * \sum\limits_{x \in \mathcal{A}_p \cap \mathcal{A}_q} \text{IC}(x)}{\sum\limits_{x \in \mathcal{A}_p} \text{IC}(x) + \sum\limits_{x \in \mathcal{A}_q} \text{IC}(x)} \tag{95}$$

and

$$\text{SimUIC}(p,q) = \frac{\sum\limits_{x \in \mathcal{A}_p \cap \mathcal{A}_q} \text{IC}(x)}{\max\left\{\sum\limits_{x \in \mathcal{A}_p} \text{IC}(x), \sum\limits_{x \in \mathcal{A}_q} \text{IC}(x)\right\}} \tag{96}$$

Finally, the SimUI approach [68], which refers to the union-intersection entity semantic similarity measure, is a particular case of SimGIC assigning equal IC value to all terms in the ontology DAG [3]. Even though this assumption is not realistic in the context of the GO DAG, the SimUI measure can still be used as an alternative measure in practice as it showed relatively good performance when applied to different biological data [64]. This measure does not depend on concept IC values and is given by

$$\text{SimUI}(p,q) = \frac{|\mathcal{A}_p \cap \mathcal{A}_q|}{|\mathcal{A}_p \cup \mathcal{A}_q|} \tag{97}$$

Similarly, one can define particular cases based on SimDIC (Dice) [65] and SimUIC (Universal), denoted by SimDB and SimUB, respectively, and given by

$$\text{SimDB}(p,q) = \frac{2 * |\mathcal{A}_p \cap \mathcal{A}_q|}{|\mathcal{A}_p| + |\mathcal{A}_q|} \quad \text{and} \quad \text{SimUB}(p,q) = \frac{|\mathcal{A}_p \cap \mathcal{A}_q|}{\max\{|\mathcal{A}_p|, |\mathcal{A}_q|\}} \tag{98}$$

A variant of SimUB was suggested, known as normalized term overlap (SimNTO) [66], and defined as follows:

$$\text{SimNTO}(p,q) = \frac{|\mathcal{A}_p \cap \mathcal{A}_q|}{\min\{|\mathcal{A}_p|, |\mathcal{A}_q|\}} \tag{99}$$

These four measures are the normalized measure of the term overlap (TO) with scores computed as $\text{TO}(p,q) = |\mathcal{A}_p \cap \mathcal{A}_q|$.

In the case of the Cosine measure, the functional similarity score between two proteins p and q is calculated using a dot product and normalized using either a usual [65] or Tanimoto coefficient [33] scheme. Using the usual normalization model, this similarity score is given by:

$$\text{SimCOU}\,(p,q) = \frac{\langle \overline{p}, \overline{q} \rangle}{\|\overline{p}\| \|\overline{q}\|} \tag{100}$$

where $\langle \overline{p}, \overline{q} \rangle$ is the dot product between the two feature protein vectors $\overline{p}$ and $\overline{q}$ of proteins p and q, respectively. The feature protein vector of a protein $\omega = p$ or q is a vector $\overline{\omega} = (\omega_1, \ldots, \omega_m)$ of length $m = |\mathcal{A}_p \cup \mathcal{A}_q|$ in which each component $\omega_\imath$ for $\imath = 1, \ldots, m$, is associated with a term $t_\imath \in \mathcal{A}_p \cup \mathcal{A}_q$, indicating the absence (0) or presence (1) of term $t_\imath$ in the set of terms annotating the protein under consideration and weighted by its IC value. Thus, the component $\omega_\imath$ is given by:

$$\omega_\imath = \begin{cases} \text{IC}\,(t_\imath) & \text{if } t_\imath \in \mathcal{A}_\omega \\ 0 & \text{otherwise} \end{cases} \tag{101}$$

Unlike in the case of the IntelliGO measure in which the space basis is not orthogonal, here the space basis is orthonormal and thus, the dot product is computed as $\langle \overline{p}, \overline{q} \rangle = \sum_{\imath=1}^{m} (p_\imath * q_\imath)$ and the norm of $\overline{\omega}$ as $\|\overline{\omega}\| = \sqrt{\langle \overline{w}, \overline{w} \rangle} = \sqrt{\sum_{\imath=1}^{m} w_\imath^2}$. Another specialized normalization model is the Tanimoto coefficient calculated as follows:

$$\text{SimCOT}\,(p,q) = \frac{\langle \overline{p}, \overline{q} \rangle}{\|\overline{p}\|^2 + \|\overline{q}\|^2 - \langle \overline{p}, \overline{q} \rangle} \tag{102}$$

It is obvious that the SimUI, SimDB, SimUB and SimNTO measures are equivalent and the only difference between them is the normalization scheme used by each of these measures and more importantly $\text{TO}\,(p,q) = |\mathcal{A}_p \cap \mathcal{A}_q| = \langle \overline{p}, \overline{q} \rangle$ with term IC value set to 1. In this case, the Tanimoto coefficient normalization scheme is exactly SimUI and the usual normalization scheme can lead to another measure, referred to as SimCB, and equivalent to SimDB, SimUB and SimNTO.

3.4 Group-wise Edge-like measures: SimLP and Ye et al. measures

– The **SimLP** measure suggested by Gentleman [68] is computed as the longest path length in the intersection graph produced by the two sub-graphs derived from ontology concepts associated with the two entities under consideration, i.e.,

$$\text{SimLP}\,(p,q) = \max\left\{\delta\,(t) : t \in \mathcal{A}_p \cap \mathcal{A}_q\right\} \tag{103}$$

– **Ye et al.** [69] suggested a normalized version of the SimLP measure that considers the minimum and maximum path lengths within the intersection graph produced by the two sub-graphs derived from concepts associated with the two entities under consideration. In this case, the semantic similarity similarity score between two entities p and q is given by:

$$\text{SimYE}\,(p,q) = \max\left\{\frac{\delta\,(t) - \delta_{\min}}{\delta_{\max} - \delta_{\min}} : t \in \mathcal{A}_p \cap \mathcal{A}_q\right\} \tag{104}$$

3.5 Non ontology-based measures: Cho et al., Ali & Diane, Kappa-stats and others

Given a dataset with its set D of associated entities, the set S_D of all ontology concepts associated with entities in the dataset is given by

$$S_D = \underset{p \in D}{\cup} T_p \tag{105}$$

where T_p is the set of concepts associated with the entity p. Let σ be a function mapping an ontology concept with the number of entities associated with this concept, i.e., the number of times the concept occurs in the dataset.

– For the **Cho et al.** measure [70], the semantic similarity score, $\text{SimCHO}\,(p, q)$, between two entities p and q is calculated as follows:

$$\text{SimCHO}\,(p, q) = \frac{\log\left(\frac{C_{pq}}{C_{max}}\right)}{\log\left(\frac{C_{\min}}{C_{max}}\right)} \tag{106}$$

where $C_{pq} = \min\{\sigma\,(t) : t \in T_p \cap T_q\}$, $C_{\min} = \min\{\sigma\,(t) : t \in T_D\}$ and $C_{\max} = \max\{\sigma\,(t) : t \in T_D\}$.

– **Ali and Diane** [71] suggested computing the semantic similarity score, $\text{SimALD}\,(p, q)$, between two entities p and q as described below:

$$\text{SimALD}\,(p, q) = \max\left\{1 - \frac{\sigma\,(t)}{\sum\limits_{s \in S_D} \sigma\,(s)} : t \in T_p \cap T_q\right\} \tag{107}$$

– In the context of the **Kappa-statistics** measure [72], the entity semantic similarity score, $\text{SimKPS}\,(p, q)$, is given by:

$$\text{SimKPS}\,(p, q) = \frac{\sigma_{pq} - \alpha_{pq}}{1 - \alpha_{pq}} \tag{108}$$

where σ_{pq} is the observed relative frequency of co-occurrence locations between profiles of entities under consideration. The profile of a given entity ω is a binary vector $\overline{\omega} = (\omega_1, \ldots, \omega_n)$ of length n the number of terms occurring in the dataset D, i.e., $n = |T_D|$, with each component $\omega_\imath$, for $\imath = 1, \ldots, n$, associated with the concept $t_\imath \in T_D$ and given by:

$$\omega_\imath = \begin{cases} 1 & \text{if } t_\imath \in T_\omega, \text{ i.e., } \omega \text{ is associated with } t_\imath \\ 0 & \text{otherwise} \end{cases} \tag{109}$$

Thus,

$$\sigma_{pq} = \frac{|\gamma_{pq}|}{n} \tag{110}$$

where $\gamma_{pq} = \{\imath : p_\imath = q_\imath \text{ for } \imath = 1, \ldots, n\}$, $(p_1, \ldots, p_n)$ and $(q_1, \ldots, q_n)$ are profiles of entities p and q, respectively.
α_{pq} is the likelihood of observing profiles of entities p and q in the dataset under consideration, it is computed as:

$$\alpha_{pq} = \frac{1}{n * n} \sum_{\jmath=0}^{1} \sigma_{\jmath p} \sigma_{\jmath q} \tag{111}$$

where $\sigma_{\jmath\omega} = |\Gamma_{\jmath\omega}|$ with $\Gamma_{\jmath\omega} = \{\imath : \omega_\imath = \jmath \text{ for } \imath = 1, \ldots, n\}$ for a given protein ω with the profile $\overline{\omega} = (\omega_1, \ldots, \omega_n)$, and $\jmath = 0, 1$.

– Another type of this category of measures can be derived from the SimUI, SimUB, SimDB and SimNTO when used with only concepts associated with the two entities as they occur, without considering terms in the graph produced by the two sub-graphs derived from concepts associated with these proteins. One such measure is the term overlap (TO) like functional similarity measure, which is denoted TO-like and was introduced by Lee et al. [73]. Here, the semantic similarity of a entity-pairwise is scored simply by the number of concepts shared by entities and assigning a score of zero when there is no term shared by entities under consideration. The normalized version of this TO-like measure, denoted NTO-, UI-, UB- and DB-like measures, respectively, given by:

$$\begin{aligned} &\text{NTO-like}\,(p,q) = \frac{|T_p \cap T_q|}{\min\{|T_p|, |T_q|\}},\ \text{UI-like}\,(p,q) = \frac{|T_p \cap T_q|}{|T_p \cup T_q|} \\ &\text{UB-like}\,(p,q) = \frac{|T_p \cap T_q|}{\max\{|T_p|, |T_q|\}} \quad \text{and} \quad \text{DB-like}\,(p,q) = \frac{2 * |T_p \cap T_q|}{|T_p| + |T_q|} \end{aligned} \tag{112}$$

The UB-like measure computes the average number of matched concepts between entity pairwise and there exists a variant of the TO-like measure which assigns a score 1 if the two proteins share at least one term and 0 otherwise.

References

[1] Mazandu, G. K. and Mulder, N. J. (2013) Information content-based Gene Ontology semantic similarity approaches: Toward a unified framework theory. *BioMed Research International* **2013**, Article ID 292063, 11 pages.

[2] Wang, J. Z., Du, Z., Payattakool, R., Yu, P. S., and Chen, C. F. (2007) A new method to measure the semantic similarity of GO terms. *Bioinformatics* **23(10)**, 1274–1281.

[3] Mazandu, G. K. and Mulder, N. J. (2012) A topology-based metric for measuring term similarity in the Gene Ontology. *Adv Bioinformatics* **2012**, Article ID 975783, 17 pages.

[4] Zhang, P., Jinghui, Z., Huitao, S., Russo, J., Osborne, B., and Buetow, K. (2006) Gene functional similarity search tool (GFSST). *BMC Bioinformatics* **7**, 135.

[5] Seco, N., Veale, T., and Hayes, J. (2004) An intrinsic information content metric for semantic similarity in wordnet : 16th Eureopean Conference on Artificial Intelligence, ECAI 2004, IOS Press, pp. 1089–1090.

[6] Sánchez, D., Batet, M., and Isern, D. (2011) Ontology-based information content computation. *Knowledge-Based Systems* **24**, 297–303.

[7] Zhou, Z., Wang, Y., and Gu, J. (2008) A new model of information content for semantic similarity in wordnet : Second International Conference on Future Generation Communication and Networking Symposia, FGCNS 2008, IEEE Computer pp. 85–89.

[8] Seddiqui, M. H., and Aono, M. (2010) Metric of intrinsic information content for measuring semantic similarity in an ontology : Proceedings of 7th Asia-Pacific Conference on Conceptual Modeling.

[9] Meng, L., Gu, J., and Zhou, Z. (2012) A New Model of Information Content Based on Concept's Topology for measuring Semantic Similarity in WordNet. *International Journal of Grid and Distributed Computing* **5(3)**, 81–94.

[10] Mazandu, G. K. and Mulder, N. J. (2013) DaGO-Fun: Tool for Gene Ontology-based functional analysis using term. information content measures *BMC Bioinformatics* **14**, 284.

[11] Couto, F., Silva, M., and Coutinho, P. (2005) Semantic similarity over the gene ontology: Family correlation and selecting disjunctive ancestors : ACM CIKM - Conference in Information and Knowledge Management.

[12] Xu, Y., Guo, M., Shi, W., Liu, X., and Wang, C. (2013) A novel insight into gene ontology semantic similarity. *Genomics* **101**, 368–375.

[13] Li, B., Luo, F., Wang, J. Z., Feltus, F. A., and Zhou, J. (2010) Effectively integrating information content and structural relationship to improve the go-based similarity measure between proteins : BIOCOMP'10, pp. 166–172.

[14] Song, X., Li, L., Srimani, P. K., Yu, P. S., and Wang, J. Z. (2014) Measure the Semantic Similarity of GO Terms Using Aggregate Information Content : IEEE/ACM - Transactions on Computational Biology and Bioinformatics (TCBB), vol. **11(3)**, pp. 468–476.

[15] Yang, H., Nepusz, T., Paccanaro, A. (2012) Improving GO semantic similarity measures by exploring the ontology beneath the terms and modelling uncertainty. *Bioinformatics* **28(10)**, 1383-1387.

[16] Caniza, H., Romero, A. E., Heron, S., Yang, H., Devoto, A., Frasca, M., Valentini, G., and Paccanaro, A. GOssTo: a user-friendly stand-alone and web tool for calculating semantic similarities on the Gene Ontology. Bioinformatics (2014) Advance Access published.

[17] Al-Mubaid, H. and Nguyen, H. A. (2006) A cluster-based approach for semantic similarity in the biomedical domain : 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2006. New York, USA: IEEE Computer Society, pp. 2713–2717.

[18] Li, Y., Bandar, Z. A., and McLean, D. (2003) An approach for measuring semantic similarity between words using multiple information sources. : IEEE Transactions on Knowledge and Data Engineering, vol. **15(Issue 4)**, pp. 871–882.

[19] Schlicker, A., Domingues, F. S., Rahnenfuhrer, J., and Lengauer, T. (2006) A new measure for functional similarity of gene products based on Gene Ontology *BMC Bioinformatics* **7**, 302.

[20] Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989) Development and application of a metric on semantic nets : IEEE Transaction on Systems, Man, and Cybernetics, vol. **19(1)**, pp. 17–30.

[21] Pesquita, C., Faria, D., Bastos, H., Ferreira, A., Falcão, A. O., and Couto, F. M. (2008) Metrics for GO based protein semantic similarity: a systematic evaluation. *BMC Bioinformatics* **9(Suppl 5)**, S4.

[22] Jain, S. and Bader, G. D. (2010) An improved method for scoring protein-protein interactions using semantic similarity within the gene ontology. *BMC Bioinformatics* **11**, 562.

[23] Wu, Z. and Palmer, M. (1994) Verbs semantics and lexical selection : In Proceedings of the 32nd annual meeting on Association for Computational Linguistics–Association for Computational Linguistics, pp. 133–138.

[24] Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E. G. M., and Milios, E. E. (2005) Semantic similarity methods in WordNet and their application to information retrieval on the web : Proceedings of the 7th annual ACM international workshop on Web information and data management.

[25] Leacock, C. and Chodorow, M. (1998) Combining local context with WordNet similarity for word sense identification : In WordNet–A Lexical Reference System and its Application.

[26] Wang, J., Xie, D., Lin, H., Yang, Z., and Zhang, Y. (2012) Filtering Gene Ontology semantic similarity for identifying protein complexes in large protein interaction networks. *Proteome Science* **10(Suppl 1)**, S18.

[27] Slimani, T., Yaghlane, B. B., and Mellouli, K. (2006) A New Similarity Measure based on Edge Counting : In World academy of science, engineering and technology, pp. 34–38.

[28] Shenoy, K. M., Shet, K. C., and Acharya, U. D. (2012) A New Similarity Measure for Taxonomy Based on Edge Counting International Journal of Web & Semantic Technology (IJWesT) **3(4)**, DOI : 10.5121/ijwest.2012.3403.

[29] Pekar, V. and Staab, S. (2002) Taxonomy learning: factoring the structure of a taxonomy into a semantic classification decision : Proceedings of the 19th international conference on Computational linguistics–Association for Computational Linguistics, vol. **1**, pp. 1–7.

[30] Yu, H., Gao, L., Tu, K., and Guo, Z. (2005) Broadly predicting specific gene functions with expression similarity and taxonomy similarity. *Gene* **352**, 75–81.

[31] Stojanovic, N., Maedche, A., Staab, S., Studer, R., and Sure, Y. (2001) Seal: a framework for developing semantic portals : In Proc. Int. Conf. Knowledge Capture, pp. 155–162.

[32] Zhong, J., Zhu, H., Li, J., and Yu, Y. (2002) Conceptual Graph Matching for Semantic Search : In ICCS'02 Proceedings of the 10th International Conference on Conceptual Structures–Integration and Interfaces, Springer-Verlag, pp. 92–196.

[33] Pesquita, C., Faria, D., Falcão, A. O., Lord, P., and Couto, F. M. (2009) Semantic Similarity in Biomedical Ontologies. *PLoS Comput Biol* **5(7)**, e1000443.

[34] Wu, X., Pang, E., Lin, K., and Pei, Z.-M. Improving the Measurement of Semantic Similarity between Gene Ontology Terms and Gene Products: Insights from an Edge- and IC-Based Hybrid Method. *PLoS ONE* **8(5)**, e66745.

[35] Shen, Y., Zhang, S., and Wong, H.-S. (2010) A new method for measuring the semantic similarity on gene ontology : 2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 533–538.

[36] Jiang, J. J. and Conrath, D. W. (1997) Semantic similarity based on corpus statistics and lexical taxonomy : Proceedings of the 10th International Conference on Research in Computational Linguistics, pp. 19–33.

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

[38] Lin, D. (1998) An information-theoretic definition of similarity : Proceedings of the Fifteenth International Conference on Machine Learning, pp. 296–304.

[39] Zhang, S. B. and Lai, J. H. (2015) Semantic similarity measurement between gene ontology terms based on exclusively inherited shared information *Gene* **558(1)**, 108–117.

[40] Pirró, G., and Euzenat, J. (2010) A Feature and Information Theoretic Framework for Semantic Similarity and Relatedness : Proceedings of the 9th International Semantic Web Conference (ISWC), Springer, pp. 615–630.

[41] Pirró, G. (2009) A semantic similarity metric combining features and intrinsic information content. *Data & Knowledge Engineering*, **68(11)**, 1289–1308.

[42] Pirró, G., and Seco, N. (2008) Design , Implementation and Evaluation of a New Semantic Similarity Metric Combining Features and Intrinsic Information Content. Lecture Notes in Computer Science, R. Meersman & Z. Tari eds., vol. **5332**, pp.1271–1288.

[43] Couto, F. M., Silva, M. J., and Coutinho, P. M. (2003) Implementation of a Functional Semantic Similarity Measure between Gene-Products, `https://docs.di.fc.ul.pt/jspui/handle/10455/2935/1/03-29.pdf`

[44] Garla, V. N., and Brandt, C. (2012) Semantic similarity in the biomedical domain: an evaluation across knowledge sources. *BMC bioinformatics*, **13(1)**, 261.

[45] Lord, P. W., Stevens, P. W., Brass, A., and Goble, C. A. (2003) Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics* **19(10)**, 1275–1283.

[46] Sevilla, J. L., Segura, V., Podhorski, A., Guruceaga, E., Mato, J. M., Martinez-Cruz, L. A., and Corrales, F. J. (2005) Correlation between gene expression and GO semantic similarity : IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) archive, vol. **2(4)**, 330–338.

[47] Dubuisson, M. P. and Jain, A. K. (1994) A modified hausdorff distance for object matching : In ICPR94, pp. A:566–568.

[48] Memoli, F. and Sapiro, G. (2005) Theoretical and computational framework for isometry invariant recognition of point cloud data. *J Foundations Comp Math* **5**, 313–347.

[49] Bronstein, A. M., Bronstein, M. M., Mahmoudi, M., Kimmel, R., and Sapiro, G. (2010) A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *Int J Computer Vision* **89**, 266–286.

[50] Lerman, G. and Shakhnovich, B. E. (2007) Defining functional distance using manifold embeddings of gene ontology annotations. *Proc Natl Acad Sci* **104(27)**, 11334–11339.

[51] del Pozo, A., Pazos, F., and Valencia, A. (2008) Defining functional distances over gene ontology. *BMC Bioinformatics* **9**, 50.

[52] Mazandu, G. K., Chimusa, E. R., Mbiyavanga, M., and Mulder, N. J. (2016) A-DaGO-Fun: An adaptable Gene Ontology semantic similarity based functional analysis tool. *Bioinformatics* **32(3)**, 477–479.

[53] Mazandu, G. K., Chimusa, E. R., and Mulder, N. J. (2016) Gene ontology semantic similarity tools: survey on features and challenges for biological knowledge discovery. *Brief Bioinform* **18(5)**, 886–901.

[54] Fröhlich, H., Speer, N., Poustka, A., and Beißbarth, T. (2007) GOSim–an R-package for computation of information theoretic GO similarities between terms and gene products. *BMC Bioinformatics* **8**, 166.

[55] Jeong, J. C. and Chen, X. W. (2014) A new semantic functional similarity over gene ontology : IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), vol. **12(2)**, 322–334.

[56] Mazandu, G. K., Chimusa, E. R., Mbiyavanga, M., and Mulder, N. J. (2015) The 'A-DaGO-Fun' Package. Supplementary File, *Bioinformatics* **pii**, btv590.

[57] Yu, G., Li, F., Qin, Y., Bo, X., Wu, Y., and Wang, S. (2010) GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. *Bioinformatics* **26(7)**, 976–978.

[58] Caniza, H., Romero, A. E., Heron, S., et al. (2014) GOssTo: a user-friendly stand-alone and web tool for calculating semantic similarities on the Gene Ontology. *Bioinformatics*, **30**, 2235–2236.

[59] Al-Mubaid, H. and Nagar, A. (2008) Comparison of four similarity measures based on GO annotations for gene clustering : IEEE Symposium on Computers and Communications, 6–9 July 2008, Morocco: Marrakech, Report no. 3.

[60] Al-Mubaid, H. and Nagar, A. (2008) A New Path Length Measure Based on GO for Gene Similarity with Evaluation Using SGD Pathways : 21st IEEE International Symposium on Computer-Based Medical Systems, CBMS '08, pp. 590–595.

[61] Benabderrahmane, S., Smail-Tabbone, M., Poch, O., Napoli, A., and Devignes, M. D. (2010) IntelliGO: a new vector-based semantic similarity measure including annotation origin. *BMC Bioinformatics* **11**, 588.

[62] Borgwardt, K. M., Ong, C. S., Schonauer, S., Vishwanathan, S. V. N., Smola, A., J., and Kriegel, H. -P. (2005) Protein function prediction via graph kernels. *Bioinformatics*, **21**, i47–i56.

[63] Alvarez, M. A., Qi, X., and Yan, C. (2011) A shortest-path graph kernel for estimating gene product semantic similarity. *J Biomed Semant* **2**, 3.

[64] Mazandu, G. K. and Mulder, N. J. (2014) Information Content-Based Gene Ontology Functional Similarity Measures: Which One to Use for a Given Biological Data Type? *PLoS ONE* **9(12)**, e113859.

[65] Ovaska, K., Laakso, M., and Hautaniemi, S. (2008) Fast gene ontology based clustering for microarray experiments. *BioData Mining* **1**, 11.

[66] Mistry, M. and Pavlidis, P. (2008) Gene Ontology term overlap as a measure of gene functional similarity. *BMC Bioinformatics* **9**, 327.

[67] Tversky, A. (1977) Features of similarity *Psychological Review* **84(4)**, 327–352.

[68] Gentleman, R. Visualizing and Distances Using GO. `http://bioconductor.org/packages/2.6/bioc/vignettes/GOstats/inst/doc/GOvis.pdf` (2005).

[69] Ye, P., Peyser, B. D., Pan, X., Boeke, J. D., Spencer, F. A., and Bader, J. S. (2005) Gene function prediction from congruent synthetic lethal interactions in yeast. *Molecular Systems Biology* **2005**, 1–12.

[70] Cho, Y. R., Hwang, W., Ramanathan, M., and Zhang, A. (2007) Semantic integration to identify overlapping functional modules in protein interaction networks. *BMC Bioinformatics* **8**, 265.

[71] Ali, W. and Deane, C. M. (2009) Functionally guided alignment of protein interaction networks for module detection. *Bioinformatics* **25**, 3166–3173.

[72] Huang, D. W., Sherman, B. T., Tan, Q., Collins, J. R., Alvord, W. G., Roayaei, J., Stephens, R., Baseler, M. W., Lane, H. C., and Lempicki, R. A. (2007) The DAVID Gene Functional Classification Tool: a novel biological module-centric algorithm to functionally analyze large gene lists. *Genome Biology* **8**, R183.

[73] Lee, H. K., Hsu, A. K., Sajdak, J., Qin, J., and Pavlidis, P. (2004) Coexpression analysis of human genes across many microarray data sets. *Genome Biology* **14(6)**, 1085–1094.