Manual of AP-PLN

Table of Contents

System Requirements

Running Automatic pipeline to build phenotypic linkage network (AP-PLN) requires at least Python (Python version 2.7.6) and R (R version 3.1.2)

The following Python package must be installed: numpy (numpy version 1.9.1)

The following R package must be installed: earth (earth version 4.3.3, https://cran.r-project.org/web/packages/earth/index.html)

As no compilation is required, the pipeline can be used on any computer, where Python and R are installed and it is therefore available for Windows, Linux, and MAC OS machines.

License

AP-PLN is an open source and distributed under the GNU General Public License v3.0 (http://www.gnu.org).

Download and Installation

AP-PLN is available from http://csandorfr.github.io/AP-PLN/ Download the ZIP file or the TAR ball, unzip/extract the download, and save the AP-PLN.tar.gz or AP-PLN.zip file in your favorite directory for your applications. For the TAR ball: tar –zxvf AP-PLN.tar.gz

After unzip/extract procedure a directory AP-PLN is created in your favorite applications directory. You need to move next in the directory AP-PLN (cd AP-PLN). There is one subdirectory including all scripts: src.

You must then download and extract the following TAR ball file in your AP-PLN directory:

- data.tar.gz
- example.tar.gz
- example_t2d.tar.gz

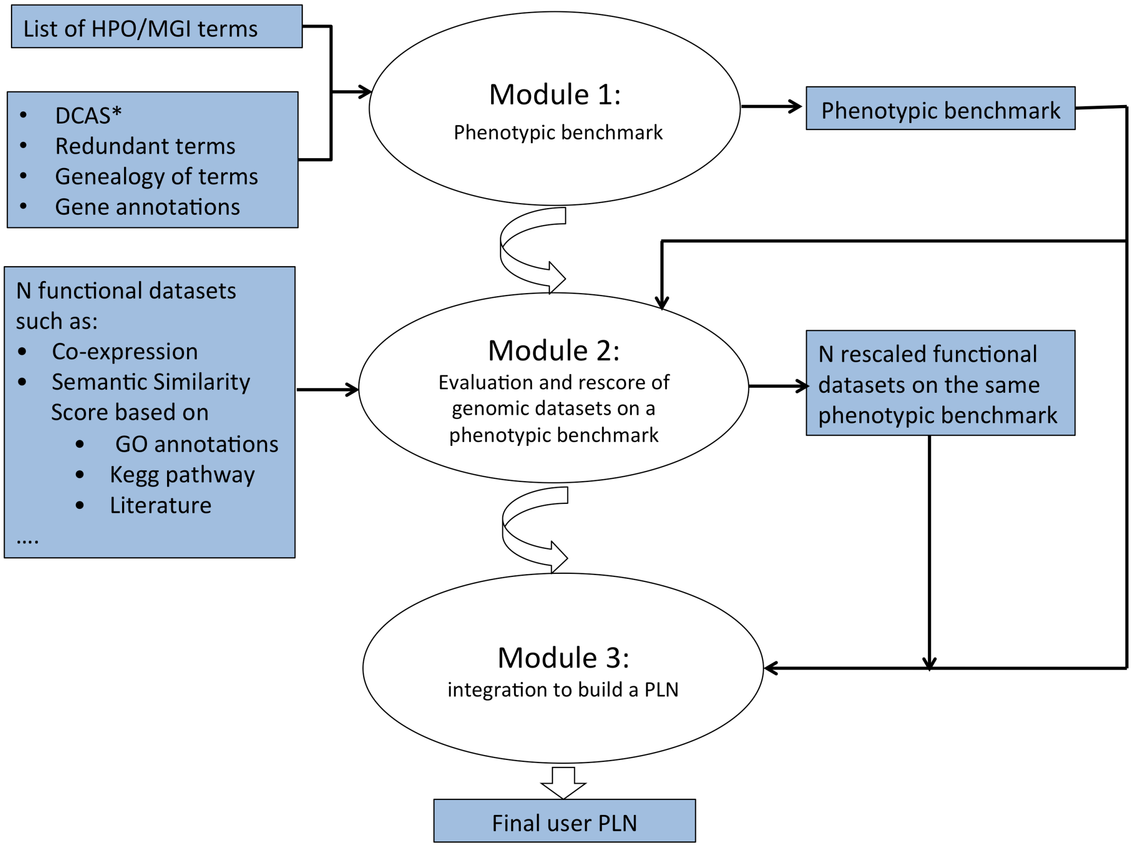
Finally, you must configure an environment variable, named $AP_PLN_HOME by this way:
export AP_PLN_HOME=directory of AP_PLN.

Overview

The pipeline AP-PLN is divided into three modules (**Figure 1**):

- The module 1 (pipeline_compute_phen_sem_sim.py) enables the user designing his own user-specific phenotypic benchmark by computing semantic similarity scores between gene pairs, from his list of Human Phenotype Ontology (HPO) [1] or Mouse Phenotype Ontology (MPO) [2] phenotype annotations.
- The module 2 (pipeline_evaluation_rescore.py) evaluates and re-scales multiple functional dataset on a phenotypic benchmark
- The module 3 (pipeline_integration.py) combines multiple functional re-scaled dataset into a final phenotypic linkage network (PLN).

Figure 1: Flowchart of pipeline to design automatically a phenotypic linkage network


List of HPO/MGI terms
DCAS*
Redundant terms
Genealogy of terms
Gene annotations
Module 1:
Phenotypic benchmark
Phenotypic benchmark
N functional datasets such as:
Co-expression
Semantic Similarity Score based on
GO annotations
Kegg pathway
Literature
....
Module 2:
Evaluation and rescore of genomic datasets on a phenotypic benchmark
N rescaled functional datasets on the same phenotypic benchmark
Module 3:
integration to build a PLN
Final user PLN

Quick Start

In your $AP_PLN_HOME/src/quick-start, you can run a python script allowing combining two functional datasets in the directory: $AP_PLN_HOME/example: coexpr_gse3594 (co-expression dataset based on the microarray transcriptional profile GSE5394) and sem_goabp by using the following command:

```
python run_all_pipeline.py $AP_PLN_HOME/example/semantic_sim_gene_mgi_all.scale
$AP_PLN_HOME/example/list_file_data $AP_PLN_HOME /example
$AP_PLN_HOME/example quick_example
```

To speed up this test, the phenotypic benchmark used to evaluate each dataset was pre-computed in: $AP_PLN_HOME/example: semantic_sim_gene_mgi_all.scale

Module1: design phenotypic benchmark

Description

The module 1, pipeline_compute_phen_sem_sim.py, in the directory, $AP_PLN_HOME /src/module1_phen_sem_sim enables the user designing his own phenotypic benchmark by computing semantic similarity scores between gene pairs, from his list of HPO or MPO reference phenotype annotations. The semantic similarity score is a measure of relatedness between two genes as assessed by the similarity in meaning of their annotations.

The module 1 pipeline_compute_phen_sem_sim.py consists:

- To reannotate the genes with only the MPO or HPO phenotype annotations and their children terms provided by the user
- To estimate a measure of information content as described by Honti et al. [3] for each phenotypic annotation reflecting the specificity of term
- To calculate the Resnik's similarity measure [4] between terms organized in a hierarchical ontology, defining the semantic similarity between any two terms t1 and t2 as the average IC of their disjunct common ancestor terms by using GraSM approach [5].
- To measure the functional relatedness of two genes, by comparing their annotations with the maximum (max) and best match average (bma) methods [6].

Options

1 *<File providing the list of MPO or HPO relevant phenotype annotations>*

2 *<Directory where useful MPO or HPO file>*

3 *<Directory where the different output files will be created (output directory)>*

4 *<0/1 Exclude or use the phenotype annotations provide by user>*

5 *<Suffix used for the name of output file created in the work directory>*

Example with MPO terms

```
python pipeline_compute_phen_sem_sim.py $AP_PLN_HOME
/example_t2d/list_t2d_mgi_term.txt $AP_PLN_HOM/data/mgi $AP_PLN_HOME/example_t2d
0 t2d_mgi
```

Example with HPO terms

```
python pipeline_compute_phen_sem_sim.py $AP_PLN_HOME
/example_t2d/list_t2d_hpo_term.txt $AP_PLN_HOM/data/hpo $AP_PLN_HOME/example_t2d
0 t2d_hpo
```

Computational performances

Table 1: Computational performances for two examples

	Number of	**Numbers of Pairs**	
Dataset	**Terms**	**Genes**	**Time**
MPO	3	5277664	48 mn
HPO	8	830591	45 mn

Tested on high specification computer, 3.6-GB RAM and two 3.16-GHz Intel Core2 Duo CPUs. HS: 148-GB RAM and 24 2.67-GHz Intel Xeon CPUs

Input files

- File providing the list of MPO or HPO relevant phenotype annotations. The user must provide the MPO/HPO phenotype annotations according to their MP/HP accession number (see http://www.informatics.jax.org/searches/MP_form.shtml and http://human-phenotype-ontology.github.io/tools.html).
- There are four useful MPO or HPO files:
    - <name>.own_format.obo genealogy information of HPO or MPO terms. There are four columns:

      Column1: accession number of HPO or MPO terms

Column2: generation term (a parental term has still a generation number smaller than its children)

Column 3: full name of HPO or MPO term

Column 4: list of accession numbers of parent's terms

- genes_ens_to_phenotype_no_red.txt phenotype annotations (MPO or HPO phenotypes annotations) for different genes. There are two columns:

  Column1: name of gene in the ensembl format

  Column2: list of phenotypes annotations under accession number format

- <name>_redundant.obo file providing a list of alias term for a specific term There are two columns

  Column1: phenotype annotations (accession number format) used here

  Column2: list of phenotype annotation alias (accession number format)

- dcas: file providing for a pair of term, the list of disjunct common ancestor terms. This file contain 3 columns:

  Column1: term 1 (format accession number)

  Column2: term 2 (format accession number)

  Column3: list of disjunct common ancestor terms for a terms pair.

The disjunct common ancestors between each term pairs don't need to be determine each time. To save time, the common disjunct ancestors between terms pairs are pre-computed and reported in the file dcas. However if the user wishes to determine again the dcas, he can use the python script: dcas.py in $AP_PLN_HOME/scripts_python/dcas) (see Additional Tool DCAS).

Output files

The semantic similarity score can be find in the work directory under this name: semantic_sim_gene_<suffixe given by the user>.scale

This file includes three columns:

Column1: Gene1

Column2: Gene 2

Column3: semantics similarity score

The semantic similarity score is scaled between zero and one and the gene pairs are sorted in ascending order according to the phenotypic semantic score.

Module 2: automatic evaluation and re-scale of multiple functional datasets on a phenotypic benchmark

Description

The module 2 evaluates individual functional dataset on a (phenotypic) benchmark and rescores then the part of each functional dataset informative according to parameters learned during the evaluation.

The module2 pipeline_evaluation_rescore.py consists:

- To order gene pairs according to by their functional relatedness measure between genes of genomic dataset evaluated (e.g. co-expression dataset). The ordered pairs are divided to bins of n pair of genes (e.g. n=500) and the mean (or the median) of the phenotypic semantic similarity scores is calculated for each bin. The degree of similarity expected from pairs of random genes, $val_{random}$is computed by using the mean (or the median) of MPO (or HPO) similarity.
- To determine, what is part of functional dataset is informative to predict phenotypic benchmark. We model with a multivariate adaptive regression splines (MARS) the relation between our functional relatedness measure of functional dataset (x) and phenotypic (MPO or HPO) semantic similarity score

(y) and we consider the threshold value x_{th} from which functional relatedness measure of functional dataset is informative by using x-intercept of our MARS model with $y=val_{random.}$

- To determine the re-score curve for the informative part to predict the phenotypic benchmark for an functional datastet. We model then the relation between functional relatedness measure of functional dataset (x) greater than the informativeness threshold x_{th} and phenotypic (MPO or HPO) semantic score (y) with a new MARS. We use then the prediction of this MARS models to re-score the functional similarity between genes of a dataset with a value > x_{th} in to phenotypic semantic score measure.

Options

1 *<File providing phenotypic similarity score between gene pairs (Module 1)>*

2 *<File providing the list of functional datasets name>*

3 *<Directory where are the different functional dataset file >*

4 *<Directory where the different output files will be created (output directory)>*

5 *<Number of gene pairs peer bin>*

6 *<0/1> Use either mean or median to estimate:*

- *the phenotypic similarity value associated with random gene pair*
- *to compute the phenotypic similarity score per bin*

7 *<Suffix of output file>*

Example

```
python pipeline_evaluation_rescore.py
$AP_PLN_HOME/example/semantic_sim_gene_mgi_all.scale
$AP_PLN_HOME/example/list_file_data  $AP_PLN_HOME/example  $AP_PLN_HOME
/example 500 0 mpi_all
```

Computational performances

Table 2: Computational performances for two functional datasets

Dataset	**Numbers of Pairs Genes**	**Time**
Semantic Similarity (GO BP)	5967478	2 mn 15 s
Co-expression (GSE3594)	2779725	1 mn 10 s

Tested on high specification computer, 3.6-GB RAM and two 3.16-GHz Intel Core2 Duo CPUs. HS: 148-GB RAM and 24 2.67-GHz Intel Xeon CPUs

Input files

The input files are phenotypic similarity score and the functional dataset that the user wishes to integrate in his PLN. They have the following same format:

Column1: Name of gene 1 in ensembl annotation

Column2: Name of gene 2 in ensembl annotation

Column3: Phenotypic similarity score or functional relatedness measure of genes pairs.

There are two prerequisites regarding the phenotypic similarity score or functional relatedness measure:

- Phenotypic similarity score or functional relatedness measure must be scaled between zero and one (e.g. by using the function scale of R).
- The gene pairs must be ordered descending according to their phenotypic semantic score or functional related score.

The file must including just one time a gene pair. Therefore, Gene1 Gene2 Score is equivalent to Gene2 Gene1 Score

Output files

- <*name of functional dataset*>.<*suffix output file*>.eval file is file where the ordered pairs were divided to bins of n pair of genes (e.g. n=500) and the mean (or the median) of the phenotypic semantic similarity scores is calculated for each bin.
- <*name of functional dataset*>.<*suffix output file*>.png graphic is a plot of eval file. the x-axis and y-axis correspond to functional relatedness measure between genes pairs and phenotypic similarity score respectively. The value of random pairs is showed by horizontal dotted gray line, while a vertical blue line represents the threshold value of informativness. All gene pairs with functional relatedness measure less than this threshold would be discarded. The curve represents the MARS model used to rescore the data (example **Figure 2**).
- <*name of functional dataset*>.<*suffix output file*>.rescore file correspond to functional dataset rescored in phenotypic semantic similarity score measure. The format is the same than both input files

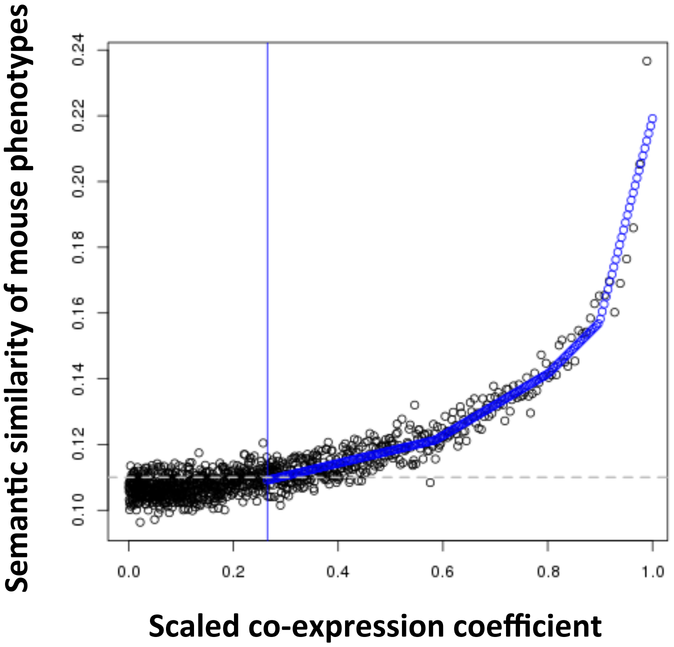


Figure 2: gse3594_11757_ordered_filter_0.4_sort.png

The data types were assessed and weighted according to the phenotypic benchmark which is a measure of semantic similarity estimated from gene phenotype annotations (see Module1) . The depicted co-expression data (GSE3594) were ordered by the Pearson's correlation coefficient and divided to bins of 500 gene pairs (black dots). The blue multivariate adaptive regression splines curve represents the (re)scoring function that associates co-expression correlation coefficients with the corresponding semantic similarity measured with mouse phenotype annotations. The median of the semantic similarity values of 500 gene pairs has been calculated for each bin.

Module 3: Integration of re-scaled functional dataset into single phenotypic linkage genes network

Description

The module3, pipeline_integration.py integrates multiple functional datasets re-scored on a phenotypic benchmark. In the case, where different genomic datasets suggest a functional link between the same gene pairs, the functional measures of different dataset are summed, by penalizing the less reliable data according to a formula proposed by Lee et al. [7]:

$$WS = L_o + \sum_{i=1}^{n} \frac{L_i}{D \times i}$$

, where L represents a re-scored functional measure from a single data set, L_0 being the largest functional measure among all the functional datasets between the given two genes, *i* is the index of the remaining links ordered by their weights for the gene pair and *D* is a free parameter. The value of *D* was optimized, by using the integrated dataset with a *D* parameter that is the best linear predictor a phenotypic semantic similarity measure.

Options

1 *< File providing phenotypic similarity score between gene pairs (Module 1) >*

2 *<File providing the list of name of file of re-scored functional datasets on a phenotypic benchmark>*

3 *<Data directory, directory where are re-scored functional datasets files>*

4 *<Directory where the different output files will be created (output directory)>*

5 *<Number of gene pairs peer bin>*

6 *<0/1> Use either mean or median to estimate to compute the phenotypic similarity score per bin*

7 *<Suffixe of output file>*

Example

```
python pipeline_integration.py $AP_PLN_HOME/example/semantic_sim_gene_mgi_all.scale
$AP_PLN_HOME/example/list_file_rescore.mgi_all $AP_PLN_HOME/example
$AP_PLN_HOME/example  500 0 mgi_allnet
```

Computational performances

Table 3: Computational performances to build PLN with two re-scaled datasets

Dataset	Numbers of Pairs Genes	Max D	Time
Semantic Similarity (GO BP)	5967478	6	5 mn 15 s
Co-expression (GSE3594)	2779725		

Tested on high specification computer, 3.6-GB RAM and two 3.16-GHz Intel Core2 Duo CPUs.

HS: 148-GB RAM and 24 2.67-GHz Intel Xeon CPUs

Input files

- Phenotypic similarity score (format see above)
- File providing the list of file of functional rescaled datasets. There is just one column, where for each line, there is the file name of functional rescaled dataset.
- .rescore file correspond to functional datasets rescored on a phenotypic semantic similarity score measure. The format is the described above

Output files

- wl_<*suffix of output file*>.scale.ord File of final phenotypic network. This file includes three columns:
  - Column1: Name of gene 1 in ensembl annotation
  - Column2: Name of gene 2 in ensembl annotation
  - Column3: functional weighted link measure

  The gene pairs are ordered descending according to their final functional weighted link measure. The weighted link measure
- best_parameter_d file reporting the *D* parameter used.
- best_parameter_d.png png graphic showing the evaluation of integrated measure with different *D* parameter.

Additional tools

Disjunct common ancestors

The disjunct common ancestors between two terms can be determined by using the python script dcas.py (directory: $AP_PLN_HOME/src/scripts_python/dcas). This script takes two arguments: (1) obo genealogy information file (format described above) (input file) (2) dcas file (format described above) (e.g. python dcas.py hp_own_format.obo dcas)

Scale

Functional relatedness measures for an individual dataset can be scaled between zero and one by using the python script, scale_dataset.py (directory: $AP_PLN_HOME /src/scripts_python/scale_dataset.py). This script takes two arguments: (1) input file of functional associations measure between gene pairs (2) output file of scaled functional associations measure between gene pairs

Sort gene pairs by functional associations measure

The gene pairs can be ordered descending according to their functional association measure by using the python script sort_pair_value.py. This script takes two arguments: (1) input file of functional associations measure between gene pairs (2) output file of sorted functional associations measure between gene pairs.

Comparison of accuracy and gene pairs coverage of multiple re-scaled functional datasets

The python script pipeline_evaluation.py (directory $AP_PLN_HOME/src/module4_comparison_dataset) allows comparing visually different functional datasets and final PLN revealing their relative accuracy to predict phenotypic benchmark and their coverage. This script takes 7 arguments: (1) phenotypic benchmark file (2) file of list of functional datasets re-scaled (3) data directory (4) work directory (5) Number of gene pairs by bin (6) use mean or median (0 or 1) (7) prefix of output file

Example: python pipeline_evaluation.py
$AP_PLN_HOME/example/semantic_sim_gene_MPO_all.scale your
$AP_PLN_HOME/example /list_file_rescore_wl.MPO_all $AP_PLN_HOME/example
$AP_PLN_HOME/example 200 0 MPO_allnet

The output file is graphic file under png format showed in the figure below:

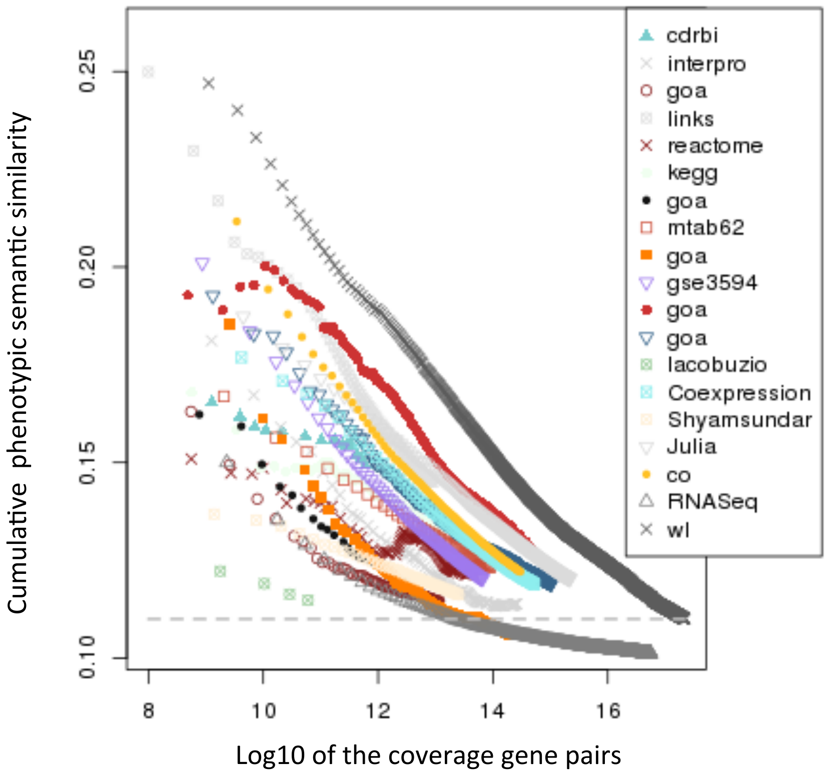


Figure 3: Comparison of information provided by different data types and by the final PLN.

Different data types provide information of characteristic accuracy over different sets of genes. The Y-axis gives the semantic similarity of the phenotypes from the pairwise mouse model comparisons given the number of gene-gene links covered on the X-axis.

References

1. Robinson PN, Kohler S, Bauer S, Seelow D, Horn D, Mundlos S. The Human Phenotype Ontology: a tool for annotating and analyzing human hereditary disease. Am J Hum Genet. 2008;83(5):610-5. doi: 10.1016/j.ajhg.2008.09.017. PubMed PMID: 18950739; PubMed Central PMCID: PMCPMC2668030.

2. Blake JA, Bult CJ, Eppig JT, Kadin JA, Richardson JE, Mouse Genome Database G. The Mouse Genome Database: integration of and access to knowledge about the laboratory mouse. Nucleic Acids Res. 2014;42(Database issue):D810-7. doi: 10.1093/nar/gkt1225. PubMed PMID: 24285300; PubMed Central PMCID: PMCPMC3964950.

3. Honti F, Meader S, Webber C. Unbiased functional clustering of gene variants with a phenotypic-linkage network. PLoS Comput Biol. 2014;10(8):e1003815. doi: 10.1371/journal.pcbi.1003815. PubMed PMID: 25166029; PubMed Central PMCID: PMCPMC4148192.

4. Philip R. Using information content to evaluate semantic similarity in a taxonomy. Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1 %@ 1-55860-363-8, 978-1-558-60363-9. Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc.; 1995. p. 448-53.

5. Couto FM, Silva MJ, Coutinho PM. Measuring semantic similarity between Gene Ontology terms. Data & Knowledge Engineering. 2007;61(1):137-52. doi: 10.1016/j.datak.2006.05.003.

6. Pesquita C, Faria D, Bastos H, Ferreira AE, Falcao AO, Couto FM. Metrics for GO based protein semantic similarity: a systematic evaluation. BMC Bioinformatics. 2008;9 Suppl 5:S4. doi: 10.1186/1471-2105-9-S5-S4. PubMed PMID: 18460186; PubMed Central PMCID: PMCPMC2367622.

7. Lee I, Blom UM, Wang PI, Shim JE, Marcotte EM. Prioritizing candidate disease genes by network-based boosting of genome-wide association data. Genome Res. 2011;21(7):1109-21. doi: 10.1101/gr.118992.110. PubMed PMID: 21536720; PubMed Central PMCID: PMCPMC3129253.