Manual of AP-PLN

Table of Contents

System Requirements

Running Automatic pipeline to build phenotypic linkage network (AP-PLN) requires at least a Python version > 2.7.6 and an R version > 3.1.2. The following Python package must be installed: numpy (numpy version 1.9.1). The following R packages must be installed: earth (> 4.3.3), igraph (1.1.2) and biomaRt (2.30.0). 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 file, unzip/extract the download, and save it in your favorite directory for your applications. To uncompress the TAR ball file, type the following command:

tar xvzf 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 your AP_PLN directory

Overview

A basic analysis in AP-PLN consists of three steps (**Figure 1**):

- The user designs his own phenotypic benchmark, by computing semantic similarity scores between gene pairs, from a specified list of Human Phenotype Ontology (HPO) [1]or Mouse Phenotype Ontology (MPO) [2] phenotype annotations (**module 1 pipeline_compute_phen_sem_sim.py**).
- Different functional datasets considered by the user to build the final PLN are re-scaled on a unique phenotypic benchmark (step1) (**module 2 pipeline_evaluation_rescore.py**)
- The different re-scaled functional dataset are combined to build the final phenotypic linkage network (PLN) (**module 3 pipeline_integration.py**).

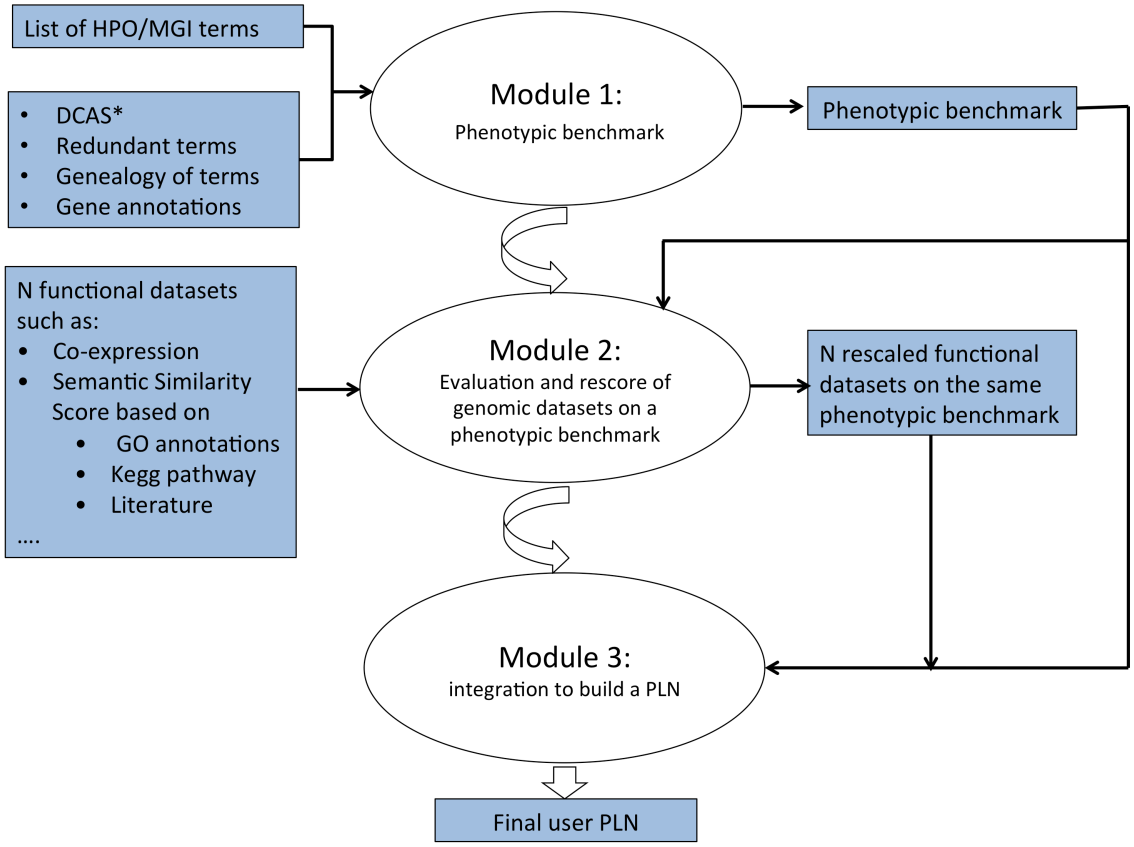

Figure 1: Flowchart of pipeline to design automatically a phenotypic linkage network

Quick Start

In your $AP_PLN_HOME/src/quick-start, you can run a python script allowing combining two functional datasets:

- coexpr_gse3594 (Co-expression dataset based on the microarray expression profile (GSE5394))
- sem_goabp (Semantic similarity score based on Gene annotations of human and mouse genes in the Biological Process)

, in the directory: $AP_PLN_HOME/example, 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 re-scale each functional dataset was pre-computed in: $AP_PLN_HOME/example: semantic_sim_gene_mgi_all.scale

Module1: design phenotypic benchmark

Python script

pipeline_compute_phen_sem_sim.py

Directory

$AP_PLN_HOME /src/module1_phen_sem_sim

Description

This python script 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.

Details

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 MGI or HPO relevant phenotype annotations* >

2 < *Directory of output files (output directory)* >

3 < *Suffix of output file* >

Example with MPO terms
python pipeline_compute_phen_sem_sim.py $AP_PLN_HOME
/example_t2d/list_t2d_mgi_term.txt $AP_PLN_HOME/example_t2d 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_HOME/example_t2d t2d_hpo

Computational performances

Table 1: Computational performances for two examples

Dataset	Number of Terms	Numbers of Pairs 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).

Output files

- <*m1_suffix output file*>.log a log file that summarizes the parameters used for the specific analysis run, and list the different steps
- < *suffix output file*>.scale, tab delimited file of the phenotypic similarity score, with following columns: (1) gene1, (2) gene 2 and (3) semantic 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: evaluation and re-score of each individual dataset on a phenotypic benchmark

Python script

pipeline_evaluation_rescore.py

Directory

$AP_PLN_HOME /src/module2_evaluation_rescore

Description

The module2 pipeline_evaluation_rescore.py consists:

- To evaluate each individual dataset on a phenotypic benchmark by estimating the ability of an individual functional dataset to identify genes more likely to influence the same phenotype. The relation between a score derived from the functional dataset and the semantic phenotypic similarity measure computed with the module is examined. The gene pairs are sorted according to data-specific scores (supposed proportional to strength of functional information) in descending order. The ordered pairs are divided to bins of x gene pairs and the median (or mean) of the phenotypic semantic similarity scores between gene pairs is calculated and plotted for each bin. The degree of phenotypic similarity expected for random gene pairs is equal to median of all gene pairs (yrandom). This plot is used to determine what strength of functional relation (linkage) within each functional dataset is informative to predict phenotypic similarity and to rescale the informative part of this last on the phenotypic benchmark.

- To Re-score of each individual dataset on a phenotypic benchmark: from (1), a threshold from which low uninformative linkages derived from a given dataset are identified and informative bins are defined as those above the overall median of the semantic phenotypic similarity measure. To determine this threshold, a multivariate adaptive regression splines (MARS) model the relation between the score derived from the functional dataset (x) and the phenotypic semantic similarity score (y) and the x-intercept of this MARS with yrandom. Above this threshold, we fit a novel MARS curves in order to re-score the links so that any data-specific scores characterising the gene pairs are replaced with the semantic similarity of phenotypes that they correspond to according to our MARS function.

Options

1 *< Phenotypic benchmark (Module 1) >*

2 *< File with the functional datasets to be evaluated &rescaled >*

3 *< Directory of each functional dataset file >*

4 *< Directory of output files (output directory) >*

5 *< Bin size: Number of gene pairs peer bin >*

6 *< 0/1 > Use either the mean or the 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: (1) phenotypic similarity score (phenotypic benchmark) and each functional datasets. They are tab-delimited file; each row referring to an unique genes pair; Columns:(1) and (2) gene pairs and (3) the phenotypic similarity score or score derived from the functional dataset. We used here the ensemble gene ID. Furthermore, there are two prerequisites regarding the phenotypic similarity score or score derived from the functional dataset: (1) the scores must be scaled between zero and one (e.g. by using the function scale of R). (2) The gene pairs are descending ordered according to their phenotypic semantic score or functional related score.

Output files

All output files will be printed to output directory provides by the user (option 4).
The different output files are:

- *< m2_suffix output file.log >* a log file that summarizes the parameters used for the specific analysis run, and list the different steps
- *< 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.
- *< functional dataset. suffix output file.png >* is a plot) used to determine what strength of functional relation (linkage) within each functional dataset is informative to predict phenotypic similarity and to rescale the informative part of this last on the phenotypic benchmark. The x-axis and y-axis correspond to the score derived from the functional dataset and the phenotypic similarity score respectively. The phenotypic semantic similarity score of random gene pairs is showed by horizontal dotted gray line, while a vertical blue line

represents the threshold from which low uninformative linkages derived from a given dataset are identified. The curve represents the MARS model used to rescore the data (example **Figure 2**).

- < *functional dataset. suffix output file.rescore* > file with functional dataset rescored on the phenotypic benchmark. The format is identical to the input files

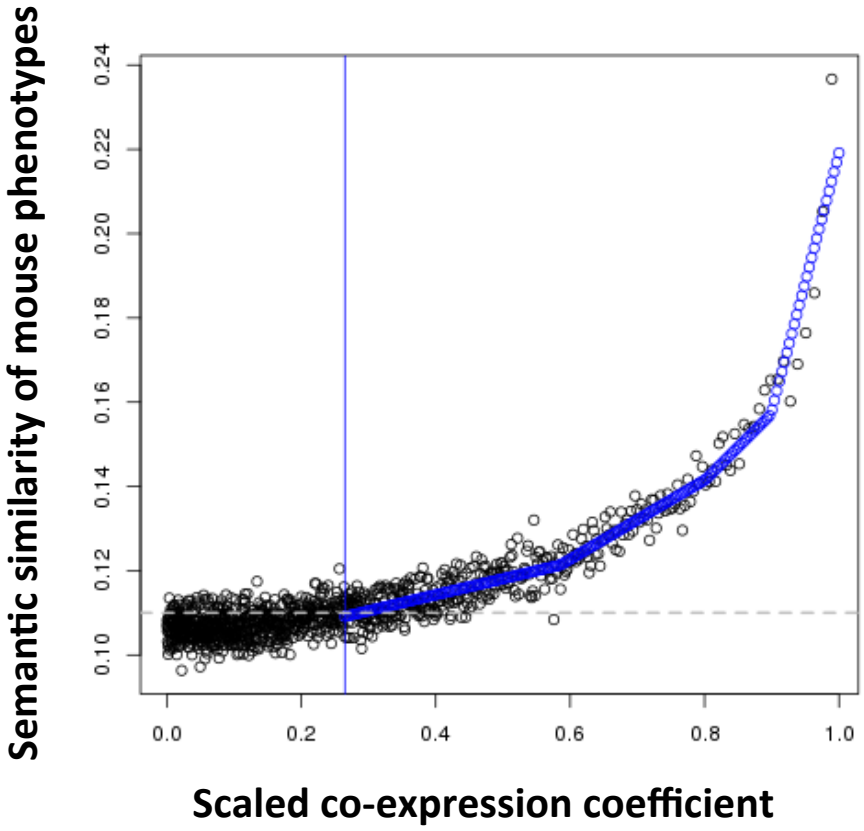

Figure2: Example with a co-expression dataset how each individual functional dataset are evaluated and rescored on a phenotypic benchmark. The depicted co-expression data (GSE3594) were ordered by the Pearson's correlation coefficient and divided to bins of 500 gene pairs (black dots). The blue multivariate 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-scored functional dataset into single phenotypic linkage genes network

Python script

pipeline_integration.py

Directory

$AP_PLN_HOME /src/ module3_integration

Description

The module3, pipeline_integration.py combines 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 *< Phenotypic benchmark (Module 1)>*

2 *< List of functional datasets rescaled on a phenotypic benchmark (Module 2) >*

3 *< Directory of each rescaled functional dataset file >*

4 *< Directory of output files (output directory) >*

5 *< Bin size: 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 *<Suffix 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 rescored functional 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

The input files are: phenotypic similarity score (phenotypic benchmark) and each functional datasets rescored on a phenotypic benchmark. They are tab-delimited file; each row referring to a unique genes pair; Columns:(1) and (2) gene pairs and (3) the phenotypic similarity score or score derived from the functional dataset. We used here the ensemble gene ID. Furthermore, there are two prerequisites regarding the phenotypic similarity score or score derived from the functional dataset: (1) the scores must be scaled between zero and one (e.g. by using the function scale of R). (2) The gene pairs are descending ordered according to their phenotypic semantic score or functional related score.

Output files

- *< m3_suffix output file.log >* a log file that summarizes the parameters used for the specific analysis run, and list the different steps
- *<wl_suffix of output file.scale.ord>* Final phenotypic network. It is tab-delimited file; each row referring to a unique genes pair; Columns 1 & 2 gene pairs and column 3 the 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* reports the *D* parameter used.
- *best_parameter_d.png* plot showing the evaluation of integrated measure with different *D* parameter.

Additional tools

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

The python $AP_PLN_HOME/src/additional_tools/comparison_dataset/pipeline_evaluation.py compares visually different functional rescaled datasets in term of relative accuracy to predict phenotypic benchmark and gene pairs coverage. This script takes seven 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

The output file is plot showing the figure below:

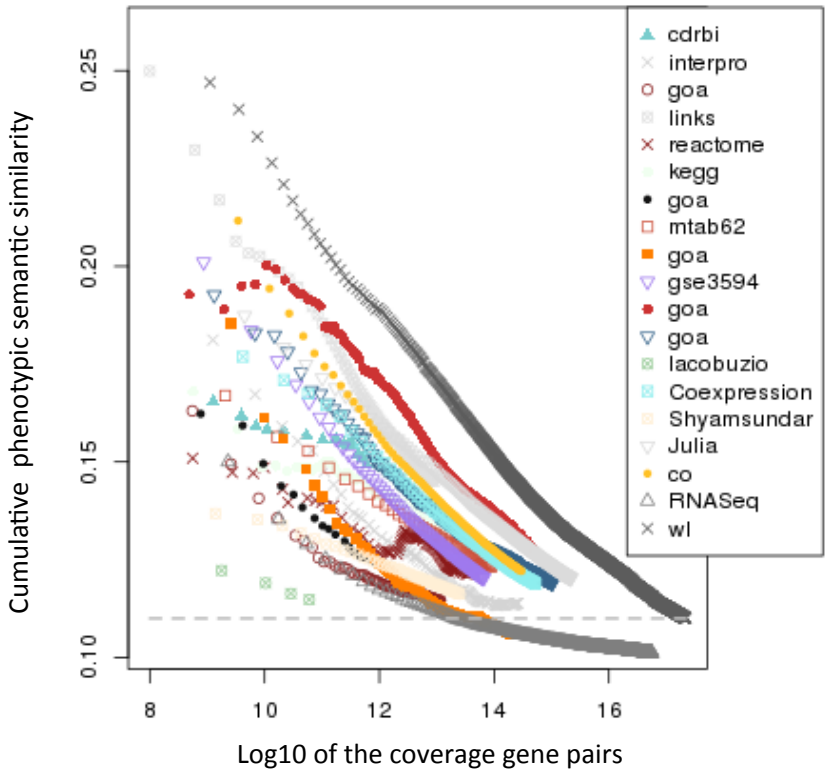

Figure 3: Comparison of information provided by different data types and by the final PLN.
Different data types provide information of characteristic accuracy over different sets of genes. The Y-axis gives the semantic similarity of the phenotypes from the pairwise mouse model comparisons given the number of gene-gene links covered on the X-axis.

Network representation for a set of gene

The R script *$AP_PLN_HOME/src/additional_tools/draw_network/make_network.R* represents the functional associations within PLN for a set of gene. This script takes two arguments: (1) the list of gene (gene symbol annotation) and (2) phenotypic linkage network. The output file are: (1) network representation: network_representation.pdf (**Figure 4**) (2) a igraph object: net_igraph.Rdata

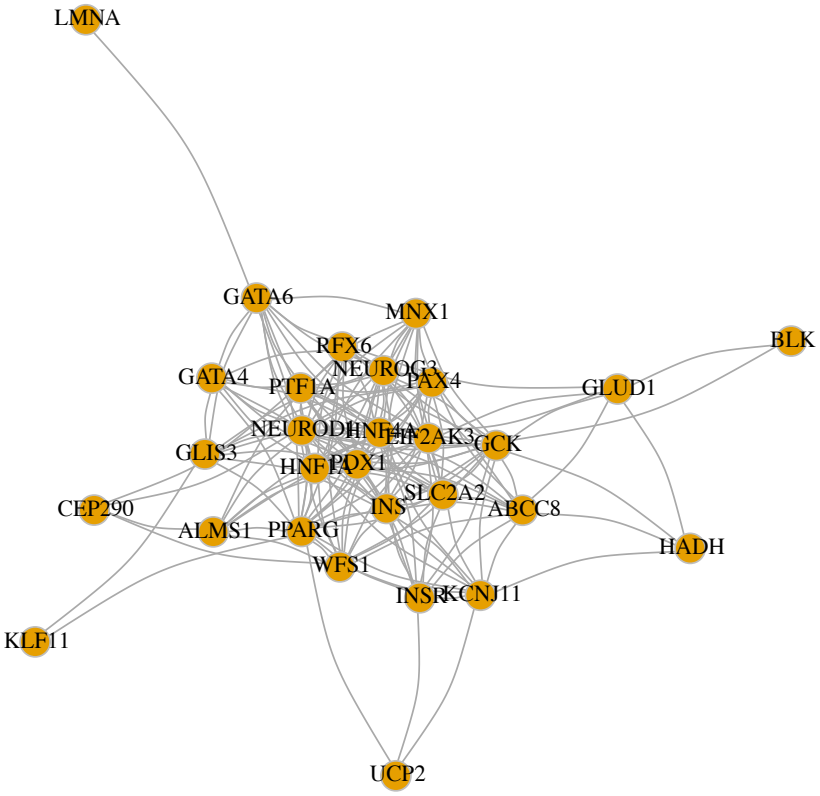

Figure 4: Example of network representation Network represention by using a generic PLN (http://wwwfgu.anat.ox.ac.uk/downloads/compbio_projects/CW003_SANDOR_T2D/pln.gz) and genes associated with a list of monogenic and syndromic diabetes genes (S3 Table of Sandor et al. (2017) {Sandor, 2017 #112})

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.