Manual of AP-PLN

Table of Contents

SYSTEM REQUIREMENTS **3**

DOWNLOAD AND INSTALLATION **3**

OVERVIEW **4**

QUICK START **5**

MODULE1: DESIGN PHENOTYPIC BENCHMARK **5**

DESCRIPTION 5

OPTIONS 6

EXAMPLE 6

INPUT FILES 6

OUTPUT FILES 8

MODULE 2: AUTOMATIC EVALUATION AND RE-SCORE OF A GENOMIC DATASET ON A PHENOTYPIC BENCHMARK **8**

DESCRIPTION 8

OPTIONS 9

EXAMPLE 10

INPUT FILES 10

OUTPUT FILES 10

MODULE 3: INTEGRATION OF RE-SCALED DATASET INTO SINGLE PHENOTYPIC LINKAGE GENES NETWORK **12**

DESCRIPTION 12

OPTIONS 13

EXAMPLE 13

INPUT FILES 13

OUTPUT FILES 13

ADDITIONAL TOOLS 14

DISJUNCT COMMON ANCESTORS 14

SCALE 14

SORT GENE PAIRS BY FUNCTIONAL ASSOCIATIONS MEASURE 14

COMPARISON OF ACCURACY AND GENE PAIRS COVERAGE OF MULTIPLE RE-SCALED FUNCTIONAL DATASET 14

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 are available (for Linux) 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 directory in your favorite directory for applications. You need to move next in the directory AP-PLN (cd AP-PLN).

There are one subdirectory including all scripts: src.

There are 3 tar.gz including data.tar.gz, example.tar.gz and example_tar.tar.gz and data.tar.gz. To extract *.tar.gz (tar –zxvf data.tar.gz).

Finally, you must configure a environment variable, named $AP_PLN_HOME by this way:

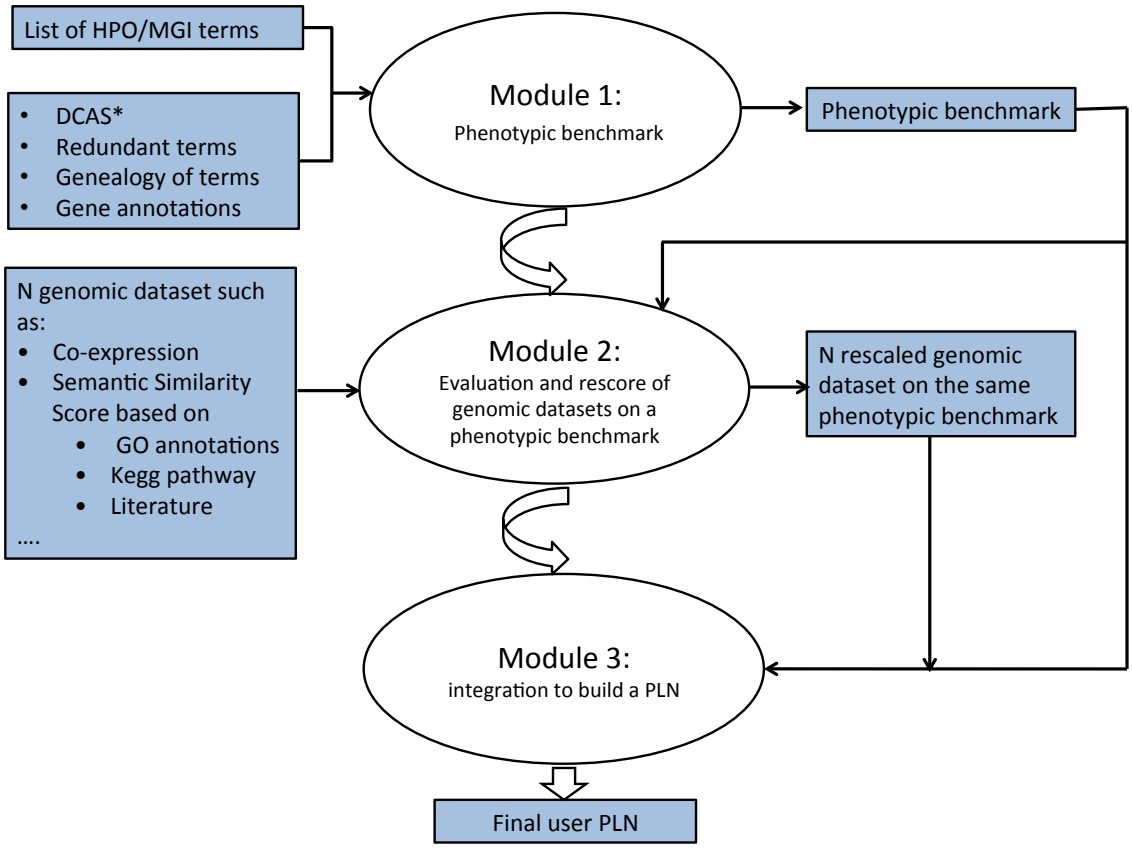
export AP_PLN_HOME=directory of AP_PLN

Overview

The pipeline to design a specific-user network is divided into three following modules (**Figure 1**):

- Module 1 (pipeline_compute_phen_sem_sim.py) allows designing a user-specific phenotypic benchmark by computing semantic similarity scores between gene pairs, for a list of HPO/MGI phenotype annotations provided by user.
- Module 2 (pipeline_evaluation_rescore.py) evaluates and re-scales a genomic dataset on a phenotypic benchmark
- Module 3 (pipeline_integration.py) integrates multiple functional genomic dataset re-scaled on a unique phenotypic benchmark.

Figure 1: Flowchart of pipeline to design user-specific phenotypic linkage network



Quick Start

In your $AP_PLN_HOME/src/quick-start, you can run a python script allowing combining two functional dataset 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_data $AP_PLN_HOME/CW016_SANDOR_AP_PLN/example $AP_PLN_HOME /CW016_SANDOR_AP_PLN/example mgi_all

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, your $AP_PLN_HOME /src/module1_phen_sem_sim allows designing a user-specific phenotypic benchmark by computing semantic similarity scores between gene pairs, for a list of HPO/MGI reference phenotype annotations provided by user. The semantic similarity score is a measure of relatedness between two genes assessed by their similarity of their phenotypic annotations pairs for a type of phenotype annotation (e.g. HPO or MGI) and for a list of phenotype annotations given by user. The module 1 pipeline_compute_phen_sem_sim.py consists:

- To reannotate the genes with only the MGI 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. [1] for each phenotypic annotation reflecting the specificity of term

- To calculate the Resnik's similarity measure (ref) 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 (ref).
- To measure the functional relatedness of two genes, by comparing their annotations with the maximum (max) and best match average (bma) methods (ref).

Options

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

2 *<Directory where useful MGI 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 output directory>*

Example

python pipeline_compute_phen_sem_sim.py your $AP_PLN_HOME /example_t2d/list_t2d_mgi_term.txt $AP_PLN_HOME /data/mgi $AP_PLN_HOME /example 0 t2d_mgi

Input files

- File providing the list of MGI or HPO relevant phenotype annotations. The user must provide the MGI/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).
- Useful MGI or HPO file. There are 4 files:

- <name>.own_format.obo genealogy information on hpo/mgi term. There are four columns:

    Column1: accession number of hpo/mgi term

    Column2: generation term (parental term has still a generation number < their children)

    Column 3: full name of hpo/mgi term

    Column4: list of accession numbers of parent's terms

- genes_ens_to_phenotype_no_red.txt phenotype annotations (MGI/HPO) for gene. There are two columns:

    Column1: name of gene in the ensembl format

    Column2: list of phenotypes annotation under format of type accession numbers

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

    Column1: phenotype annotations (format accession number) used here

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

- 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 term1 and term2

The disjunct common ancestors between each term pairs don't need to be determine each time. In order 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 (directory: scripts_python/dcas) (see Additional Tool DCAS) .

Output files

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

There are 3 columns in this file:

Column1: Gene1

Column2: Gene 2

Column3: semantics similarity score

The gene pairs are sorted in ascending order according to the phenotypic semantic score of gene pairs were ascending sorted and scaled between 0 and 1.

Module 2: automatic evaluation and re-score of a genomic dataset on a phenotypic benchmark

Description

The module 2 evaluates individual functional dataset on a (phenotypic) benchmark and to rescores 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 were divided to bins of n pair of genes (e.g. n=500) and the mean (or the median) of the semantic similarity scores measured with Mouse Phenotype (MGI) or Human Phenotype Ontology (HPO) annotations 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 MGI (or HPO) similarity.

- To determine what is part of genomic dataset is informative to predict phenotypic benchmark, we model with a multivariate adaptive regression splines (MARS) the relation between our functional relatedness measure of genomic dataset (x) and phenotypic (MGI or HPO) semantic score (y). We consider the threshold value x_{th} from which functional relatedness measure of genomic dataset dataset are informative by using x-intercept of our MARS model with $y=val_{random.}$
- We model then the relation between functional relatedness measure of genomic dataset (x) greater than the informativeness threshold x_{th} and phenotypic (MGI or HPO) semantic score (y) with a new multivariate adaptive regression splines (MARS). We use then the prediction of this MARS models to re-score the functional relatedness genomics dataset with a value > xth in phenotypic semantic score measure.

Options

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

2 *<File providing the list of functional dataset file>*

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 *<Suffixe of output file>*

Example

python pipeline_evaluation_rescore.py

$AP_PLN_HOME/example/semantic_sim_gene_mgi_all.scale

$AP_PLN_HOME/example/list_data $AP_PLN_HOME/CW016_SANDOR_AP_PLN/example

$AP_PLN_HOME /CW016_SANDOR_AP_PLN/example 500 0 mgi_all

Input files

The two input files are phenotypic similarity score and functional relatedness measure of a genomic dataset evaluated. 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 the gene pairs.

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

- 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: Gene1 Gene2 Score is equivalent to Gene2 Gene1 Score

Output files

- .eval file: The ordered pairs were divided to bins of n pair of genes (e.g. n=500) and the mean (or the median) of the phenotipic semantic similarity scores is calculated for each

bin.

- .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 line, while the threshold value of informativness is . 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**).
- .rescore file correspond to functional genomic dataset rescored in phenotypic semantic similarity score measure. The format is the same than both input files

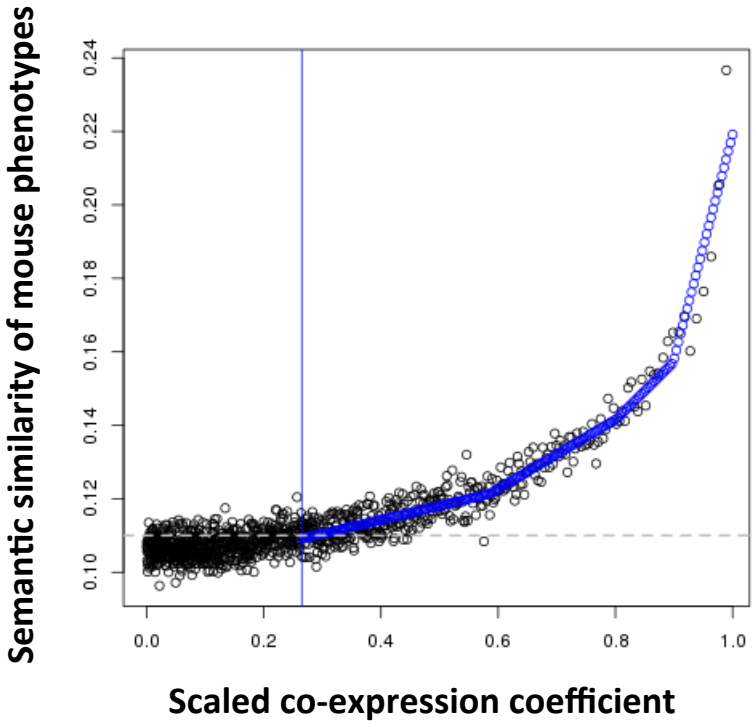


Figure 2: gse3594_11757_ordered_filter_0.4_sort.png

The data types were assessed and weighted according to phenotypic benchmark which is measure

of semantic similarity estimated from gene annotations . 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 dataset into single phenotypic linkage genes network

Description

The module3, pipeline_integration.py integrates multiple functional genomic dataset 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. [2]:

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

, where L represents a re-scored functional measure from a single data set, L0 being the largest functional measure among all the functional genomic dataset 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 dataset on a phenotypic benchmark>*

3 *<Data directory, directory where are re-scored functional dataset 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

Input files

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

Output files

-

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 dataset

The python script pipeline_evaluation.py (directory $AP_PLN_HOME/src/module4_comparison_dataset) allows comparing visually different functional genomic dataset 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 dataset 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_mgi_all.scale your $AP_PLN_HOME/example

/list_file_rescore_wl.mgi_all $AP_PLN_HOME/example $AP_PLN_HOME/example 200 0

mgi_allnet

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

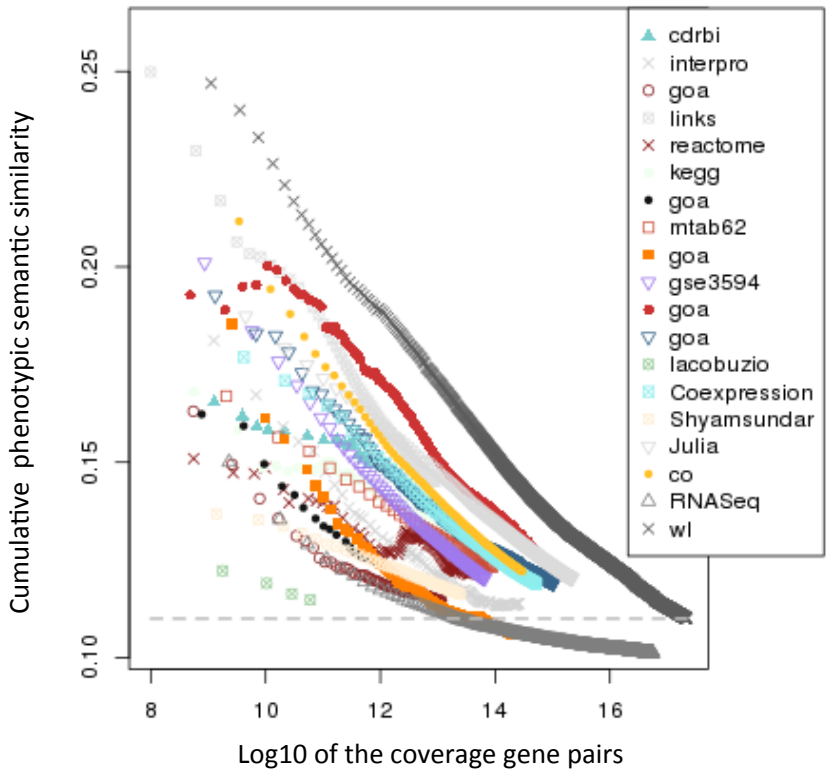


Figure 3: Comparison of information provided by different data types and by the final PLN.

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. 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.

2. 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.