HRSS v0.2 Manual

Description

This software provides a hybrid GO-based semantic similarity algorithm for evaluating the functional similarity between GO terms or gene products. The software uses the pre-downloaded GO database files and the GO annotation files. It allows the users to set organisms and evidence codes ignored. The software is composed of five modules, getGODAG, getGOAnno, hrssmatrix, hrsstps and hrsspps.

M1 getGODAG: it creates a GO database in a given MySQL database. The GO DAG contains three tables, ‘term’ and ‘term2term’ directly from pre-downloaded files, and ‘GOPaths_wholeDAG’.

M2 getGOAnno: given an organism or multi-organism (e.g. UniProtKB) GOA database, it parses the GO annotation file.

M3 hrssmatrix: given the same organism or multi-organism (e.g. UniProtKB) GOA database as that in M2, it calculates HRSS matrix for all term pairs in a given DAG or all three DAGs.

M4 hrsstps: given a GO ontology, it returns HRSS values for the term pairs in an input file.

M5 hrsspps: give a GO ontology, it returns functional similarity for the protein pairs in an input file.

Requirement

HRSS requires that MySQL is running in the system.

- If ‘LOAD LOCAL INFILE' statement is disabled, re-enable it from the mysql server side, by altering the my.cnf file (achieving the full path through ‘whereis my.cnf’).

Under the sections [mysqld] and [mysql] add 'local-infile=1':

```
[mysqld]
.... some configs ...
local-infile=1

[mysql]
.... some configs ....
local-infile=1
```

Restart mysql server with root permission.

```
/etc/rc.d/init.d/mysqld restart
```

See http://dev.mysql.com/doc/refman/5.5/en/load-data-local.html for detail.

- Secondly, login MySQL using the user with 'creating' privilage, and then create a database with the given name ('GOyeast' for example) and grant the user (e.g., name 'user' and password 'pwd') who will run the software the privileges.

```
mysql> create database GOyeast;
mysql> grant all on GOyeast.* to user@localhost identified by 'pwd';
```

For detail usage please check the MySQL manual: http://dev.mysql.com/doc/#manual

Installation

The latest release of HRSS can be accessed from http://sourceforge.net/projects/hrss/.

HRSS currently runs on linux platform. Simply put downloaded `HRSS_version.tar.gz` in any directory.

```
$ tar zxf HRSS_version.tar.gz
$ cd HRSS_version
$ chmod u+x bin/*.pl      # make the scripts executable
$ chmod u+x data/*.sh
```

There are three folders, `bin`, `data` and `results`.

- Folder `bin`: contains (1) source files in the C programming language, (2) 'Makefile' to compile the program and (3) Perl script files that will be called by hrss program. Compile the program in your platform in this way:

```
$ make clean
$ make
```

Then the compiled codes are within the same directory as the source. Add the program path to environmental variable by editing the file like ~/.bash_profile (or /etc/bashrc for root).

```
$ emacs ~/.bash_profile
            # appending like "export PATH=$PATH:$HOME/directory_to_HRSS/bin/"
$ source ~/.bash_profile
```

- Folder `data`: contains scripts for running the program and example input files.
- Folder `results`: contains result files after running the scripts in the folder of `data`.

Modules

There are five modules, namely getGODAG, getGOAnno, hrssmatrix, hrsstps, hrsspps, in the software. The example scripts are in folder '`data`'.

M1: getGODAG -- it creates a GO database in a given MySQL database. The GO DAG contains three tables, 'term' and 'term2term' directly from pre-downloaded files and 'GOPaths_wholeDAG'.

Inputs:

1. infile_of_mysql_info: input file containing the mysql information. This file is required by all modules.

```
user      password      database
```

2. directory_of_GO_database: the directory of downloaded GO database files. Only four files, namely term.txt, term.sql, term2term.txt and term2term.sql are used in the program.

Usage:

```
$ hrss --getGODAG infile_of_mysql_info directory_of_GO_database
```

Example:

```
$ hrss --getGODAG ./mysqldb.txt ./go_database/
```

Outputs:

- MySQL tables 'term' and 'term2term' created directly from pre-downloaded GO database files.
- MySQL table 'GOPaths_wholeDAG' containing all possible paths between two terms in any GO DAG.

M2: getGOAnno -- it parses the GO annotation file.

Inputs:

1. infile_of_mysql_info: input file containing the mysql information.
2. corpus: indicating an organism name or a multi-organism GOA database.
3. infile_of_GOA: the GO annotation (GOA) file for a corpus.

Usage:

```
$ hrss --getGOAnno infile_of_mysql_info corpus infile_of_GOA
```

Example:

```
$ hrss --getGOAnno ./mysqldb.txt yeast ./gene_association.sgd
```

Outputs:

- MySQL table '*corpus*_go' containing the GO annotation information. If the corpus is set as yeast, the table name is 'yeast_go'.
- Structure of MySQL tables '*corpus*_GOassos_allevi' ('none' evidence code is filtered out) and '*corpus*_GOassos_filevi' (there are evidence code(s) being filtered out later in M3).

M3: hrssmatrix -- given an organism or multi-organism (such as UniProt) GOA database, it does all the steps necessary for calculating semantic similarity scores of all term pairs.

Inputs:

1. infile_of_mysql_info: input file containing the mysql information.
2. corpus: indicating an organism name or a multi-organism GOA database.
3. evidence_codes_ignored: a comma-delimited string of evidence code(s) to be filtered out, e.g. 'IEA' and 'IEA,IKR'. If no code is ignored, set the parameter as 'none'.
4. ontology_alldags: the GO ontology to be considered, chosen from

   + alldags: all three ontologies will be considered separately.
   + BP: only biological process
   + CC: only cellular component
   + MF: only molecular function

   Note: Multiple programs in BP, CC and MF ontologies could run simultaneously.

5. directory_for_output: directory for output files.

Usage:

```
$ hrss --hrssmatrix infile_of_mysql_info corpus evidence_codes_ignored ontology_alldags \
directory_for_output
```

Example:

```
$ hrss --hrssmatrix ./mysqldb.txt yeast none alldags ../results/
$ hrss --hrssmatrix ./mysqldb.txt yeast IEA,IKR alldags ../results/
```

Outputs:

- Updated MySQL table '*corpus*_GOassos_allevi' ('none' evidence code is filtered out) OR '*corpus*_GOassos_filevi' (there are evidence code(s) being filtered out).
- Under output directory, HRSS matrix file of all term pairs (excluding root term) in a DAG is save, e.g., yeast_mx_allevi.MF indicates the matrix for yeast on MF ontology including all evidence codes. If 'alldags' is set, three such files in all three ontologies will be produced.

M4: hrsstps -- it returns the HRSS values for input term pairs in a given GO ontology.

Inputs:

1. infile_of_mysql_info: input file containing the mysql information.
2. corpus: indicating an organism name or a multi-organism GOA database.
3. evidence_codes_ignored: a comma-delimited string of Evidence code(s) to be filtered out, like 'IEA' and 'IEA,IKR'. If no code is ignored, set the parameter as 'none'.
4. ontology: the GO ontology to be considered, chosen from

   + BP: only biological process
   + CC: only cellular component
   + MF: only molecular function

   Note: Multiple programs in BP, CC or MF could run simultaneously.

5. directory_for_matrixfile: directory for pre-computed HRSS matrix file
6. infile_of_termpairs: input file containing tab-delimited term pairs. Only term accession (e.g. GO:0000001) is recognized by this software.
7. outfile_of_termpairs: name of output file

Usage:

```
$ hrss --hrsstps infile_of_mysql_info corpus evidence_codes_ignored ontology \
directory_for_matrixfile infile_of_termpairs outfile_of_termpairs
```

Example:

```
$ hrss --hrsstps ./mysqldb.txt yeast none BP ../results/ ./yeast_tps ../results/yeast_allevi_tpshrss.BP
$ hrss --hrsstps ./mysqldb.txt yeast IEA BP ../results/ ./yeast_tps ../results/yeast_filevi_tpshrss.BP
```

Outputs:

- Output file named by parameter outfile_of_termpairs, with tab-delimited string of term1_acc, term2_acc, term1_id, term2_id and HRSS in each line.

M5: hrsspps -- it returns the HRSS values for input protein pairs in a GO ontology.

Inputs:

1. infile_of_mysql_info: input file containing the mysql information.
2. corpus: indicating an organism name or a multi-organism GOA database.
3. evidence_codes_ignored: a comma-delimited string of Evidence code(s) to be filtered out, like. 'IEA' and 'IEA,IKR'. If no code is ignored, set the parameter as ‘none’.
4. pairwise: chosen from

   + max: maximum pairwise strategy
   + bma: best-match average pairwise strategy

5. ontology: GO ontology to be considered, chosen from

   + BP: only biological process
   + CC: only cellular component
   + MF: only molecular function

   Note: Multiple programs in BP, CC or MF could run simultaneously.

6. directory_for_matrixfile: directory for pre-computed HRSS matrix file
7. infile_of_proteinpairs: input file containing tab-delimited protein pairs. Only ‘DB_Object_ID’ in downloaded GO annotation file is recognized in the software.
8. outfile_of_proteinpairs: name of output file

Usage:

```
$ hrss --hrsspps infile_of_mysql_info corpus evidence_codes_ignored pairwise ontology \
```

```
directory_for_matrixfile infile_of_proteinpairs outfile_of_proteinpairs
```

Example:

```
$ hrss --hrsspps ./mysqldb.txt yeast none max BP \
../results/ ./yeast_pps ../results/yeast_allevi_maxhrss.BP

$ hrss --hrsspps ./mysqldb.txt yeast IKR max BP \
../results/ ./yeast_pps ../results/yeast_filevi_maxhrss.BP
```

Outputs:

- Output file named by parameter outfile_of_proteinpairs, with tab-delimited string of protein1, protein2 and HRSS in each line.

Note:

- The HRSS values of two proteins both with valid GO annotations range from 0 to 1. There are three negative values for a protein pair, -2, -3 and -4, indicating one or both proteins have no GO annotation. Given the protein pair P and Q, -2 means Q has no GO annotation, -3 means P has no annotation and -4 means both P and Q have no annotation.

Walkthrough examples

Two script files are under folder ‘data’, one for considering all evidence codes, and the other one for excluding IEA (Inferred from Electronic Annotation) codes. The step by step example is for considering all annotations. The input files in the example are in folder ‘data’. Run the commands under the directory of ‘data’.

Step 1. run M1 (getGODAG) to create a MySQL database of GO DAG

```
$ hrss --getGODAG ./mysqldb.txt ./go_database/
```

Step 2. run M2 (getGOAnno) to parse the pre-downloaded GO annotation file of yeast.

```
$ hrss --getGOAnno ./mysqldb.txt yeast ./gene_association.sgd
```

Step 3. run M3 (hrssmatrix) to calculate HRSS values for all term pairs in three ontologies when all evidence codes are considered.

```
$ hrss --hrssmatrix ./mysqldb.txt yeast none alldags ../results/
```

Then, users can achieve HRSS values for term pairs or protein pairs of interests.

Step 4. run M4 (hrsstps) to fetch HRSS values (yeast annotation on BP including all annotations) for the term pairs in an input file.

```
$ hrss --hrsstps ./mysqldb.txt yeast none BP ../results/ ./yeast_tps ../results/yeast_allevi_tpshrss.BP
```

Step 5. run M5 (hrsspps) to fetch HRSS (MAX) values (yeast annotation on BP including all annotations) for the protein pairs in an input file.

```
$ hrss --hrsspps ./mysqldb.txt yeast none max BP \
../results/ ./yeast_pps ../results/yeast_allevi_maxhrss.BP
```

Help screen

```
HRSS -- Hybrid Relative Specificity Similarity based on Gene Ontology (version 0.2)

Usage:
Module 1: getGODAG -- create a MySQL database of GO DAG
   hrss --getGODAG infile_of_mysql_info directory_of_GO_database
   Example:
   hrss --getGODAG ./mysqldb.txt ./go_database/
   ------------------------------------------------------------------------------------------------------------
Module 2: getGOAnno -- parse the pre-downloaded GO annotation file
   hrss --getGOAnno infile_of_mysql_info corpus infile_of_GOA
   Example:
   hrss --getGOAnno ./mysqldb.txt yeast ./gene_association.sgd
   ------------------------------------------------------------------------------------------------------------
Module 3: hrssmatrix -- calculate HRSS matrix for all term pairs in a DAG or in all three DAGs
   hrss --hrssmatrix infile_of_mysql_info corpus evidence_codes_ignored ontology_alldags \
directory_for_output
   Example:
   hrss --hrssmatrix ./mysqldb.txt yeast none alldags ../results/
   hrss --hrssmatrix ./mysqldb.txt yeast IEA,IKR BP ../results/
   ------------------------------------------------------------------------------------------------------------
Module 4: hrsstps -- fetch HRSS values for the term pairs in an input file
   hrss --hrsstps infile_of_mysql_info corpus evidence_codes_ignored ontology \
directory_for_matrixfile infile_of_termpairs outfile_of_termpairs
   Example:
   hrss --hrsstps ./mysqldb.txt yeast none BP ../results/ ./yeast_tps ../results/yeast_allevi_tpshrss.BP
   ------------------------------------------------------------------------------------------------------------
Module 5: hrsspps -- fetch HRSS values for the protein pairs in an input file
```

```
   hrss --hrsspps infile_of_mysql_info corpus evidence_codes_ignored pairwise ontology \
directory_for_matrixfile infile_of_proteinpairs outfile_of_proteinpairs
   Example:
   hrss --hrsspps ./mysqldb.txt yeast none max BP \
../results/ ./yeast_pps ../results/yeast_allevi_maxhrss.BP

Inputs:
 * directory_of_GO_database: need files term.*, term2term.*.
 * infile_of_mysql_info: contains 'user pwd dbname' and required for all modules.
 * infile_of_GOA: GO annotation file for a corpus.
 * corpus: an organism name or a multi-organism GOA database.
 * evidence_codes_ignored: comma-delimited of evidence code(s) (like 'IKR,IEA') or 'none' is set.
 * pairwise: from {max, bma }
 * ontology_alldags: from {alldags, BP, CC, MF}
 * ontology: from {BP, CC, MF}
 * infile_of_termpairs: tab-delimited pairs of term accessions.
 * infile_of_proteinpairs: tab-delimited pairs of proteins (DB_Object_ID).

Report bugs to <wuxm@gmail.com>.
```

Changelog

- Jan. 11, 2013 (version 0.1) Initial release.
- May. 2, 2013 (version 0.2) Improve the program to reduce the computational time via MySQL for running Module 'hrssmatrix', and solve the bugs resulted from the incompatibility issue across different versions of linux/MySQL/GCC.
- Nov. 19, 2013 Update documentation by appending 'Note' in Module 5 hrsspps.
- Mar. 20, 2014 Update documentation (online and PDF versions) and the file 'main.cpp' in software package by explaining the parameter 'evidence_codes_ignored' in more detail.

Contact

Any questions, problems, bugs are welcome and should be dumped to

Xiaomei Wu : wuxm at gmail dot com

Center for Plant Environmental Sensing, College of Life and Environmental Sciences, Hangzhou Normal University.