

SubGSE – An Algorithm for Gene Set Enrichment Analysis

Introduction

The program, SubGSE, is developed for gene set enrichment analysis by Xiting Yan and Fengzhu Sun from the Computational Biology and Bioinformatics Program at the University of Southern California. The program is implemented and maintained by Xiting Yan. Fengzhu Sun has the full copyright of the program.

The primary objective of this program is to measure the enrichment of differentially expressed or phenotype associated genes in given gene sets. The input for the program includes the gene expression data of several samples with corresponding phenotypic data (categorical or continuous), and several given gene sets. The program will assess the significance of the enrichment of genes, whose gene expression profiles are associated with the phenotypic data, in each given gene set.

Compile the Program

To run SubGSE, you need a compiled copy of the program, three input data files and one output file. The program is written by standard C++, so you may compile and run it under Windows 95,98 or NT operating systems. You can also compile and execute it under Unix, Linux and other operating systems. In this website, you can only download the executable code for Windows Operating System(compiled using Visual C++ 6.0 under Windows XP) and Linux System (compiled using g++ 3.4.6 under Red Hat). The source code of the program may be available upon requesting by sending emails to Dr. Fengzhu Sun.

Execute the Program

At this moment, SubGSE can only be run through command lines. Compiled copies of the program are provided for both Windows and Linux Operating Systems. Under each OS, there are basically two ways to invoke the program.

The first way is to type in the following command in the shell (Linux) or MS-DOS (Windows):

```
SubGSE filepath1 filepath2 filepath3 filepath4 N C
```

Here, `filepath1`, `filepath2`, `filepath3` and `filepath4` are the paths of the four data files corresponding to the gene expression data, gene set data, phenotypic data and result, respectively. `N` is the number of permutations and `C` is the minimal size of the strict subsets to consider in the calculations. Users are referred to our paper for details on `N` and `C`. An example is:

```
SubGSE filepath1 filepath2 filepath3 filepath4 N C
```

The other way to run SubGSE is to type in the name of the executable file:

SubGSE

In windows, the command typing can be replaced by double clicking the executable file directly. In this way, an interactive menu will appear and guide you to set the parameters. The following is an example of the menu:

```
*****
****          Welcome to use SubGSE v1.0          ****
*****
Please read the SubGSE_readme.pdf before use this program!
```

Please input the path of the gene expression data file:

gexp.txt

Please input the path of the gene set data file:

gset.txt

Please input the path of the phenotypic data file:

phen.txt

Please input the path of the result file:

result.txt

Please input the permutation times:

100

Please input the minimal size of the strict subsets:

5

...

The interactive menu is quite straightforward. We recommend you to use the first way, especially when you need to execute the program many times.

Note: The number of permutations and the number gene sets might be the two most important parameters for the speed of the program. So if you have too many gene sets, it may be effective to divide all the gene sets into several parts so that each part has small number of gene sets inside. Then SubGSE can be applied to these files that have smaller number of gene sets separately. There will be small differences in the results due to the different random numbers generators. However the differences will not be significant.

Input Files Format

The input for SubGSE includes three files containing the gene expression data, the phenotypic data and the gene set data.

Attention: If you generate all your data files under windows and want to apply SubGSE on them under a Linux system, please change the format of all the data files using dos2unix. An example is:

You are safe if the data files are generated under Linux and serve as input under windows. All the data files should be text files.

Gene Expression Data File

The gene expression data file should be a tab-delimited text file. Suppose the gene expression experiment measures the gene expression levels or changes of 1000 genes in 100 samples, the gene expression will have 1001 lines. The first line will contain the names for all the 100 samples delimited by tab as following:

```
sample1    sample2    sample3    .....    sample1001
```

The remaining 1000 lines stand for the 1000 genes and have the following format:

```
gene1    3.20512  82.148   3.1921   ...  24.19412
gene2    12.9428  1.341    0.1274   ...  2.4927
...
gene1000  20.1238  28.13798 3.12389  ...  2.4914
```

In each of these 1000 lines, there are 101 elements. The first element, as shown as “gene1” and “gene2” in the example above, is the name of the gene. Following the gene names are 100 gene expression levels or changes of the gene in the 100 samples.

Gene Set Data File

The gene set data file required for SubGSE has the same format with those in MSigDB. So gene set data files downloaded from MSigDB can be used directly. The file should also be a tab-delimited text file. Here is an example:

```
geneset1    annotation for gene set1    gene1    gene2    gene3
geneset2    annotation for gene set2    gene3    gene4    gene5    gene6    gene7
geneset3    annotation for gene set3    gene8    gene9    gene10   gene11
```

In the example above, there are 3 gene sets in total. Each line corresponds to one gene set. The first element of each line is the name for the gene set. The second element should be some annotations for the gene set which cannot contain any tab. Finally the rest of the elements should correspond to the names of those genes inside the gene set. Of course, these gene names should also be used in the gene expression data. If there is some genes in the gene set that are not measured in the gene expression experiment, the program will remove it automatically so you do not need to remove them by hand. As we described in our paper, gene sets that contain too few genes should be removed from the gene set list. However, at this stage, SubGSE does not accomplish this job automatically. So please discard gene sets that contain too few genes before feeding the data to SubGSE v1.0.

Attention: The annotation element for the gene sets, the second element, should not contain any tabs. Otherwise, the words right after the tab will be considered as a gene name.

Phenotypic Data File

The phenotypic data file should also be a text file which contains the phenotypic data for the samples in the gene expression data. The format of the phenotypic data is different depending on the measurement levels of the phenotypic data (categorical or continuous).

Categorical Phenotypic Data

If the phenotypic data is categorical, there should be three lines in the file.

The first line contains the number of samples and the number of classes which are separated by space or tab. The number of samples should be the same with that in the gene expression data file. The format of this line should be:

```
(number of samples) (number of classes) 1
```

An example is:

```
32 2 1
```

The second line starts with the character # and describes the names for all the classes in the phenotypic data. The number of names should be consistent with the number of classes in the first line. The format of this line should be:

```
# (name of class 1) (name of class 2) ...
```

An example is:

```
# MALE FEMALE
```

Note: no space is allowed in the names of classes.

The third line contains the phenotypic data which is class label for each sample in the gene expression data. The class labels for different samples are separated by space or tab. For categorical phenotypic data, the class labels can be represented by either characters or numbers. No matter in which way the phenotypic data is presented, the format of this line should be:

```
(class label 1) (class label 2) (class label 3) ... (class label 32)
```

An example which uses characters to present the class labels is:

```
m m m m m m m m m m m m m m f f f f f f f f f f f f f f f f
```

An example which uses numbers to present the class labels is:

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

In summary, categorical phenotypic data file has only three lines. The format is:

```
(number of samples) (number of classes) 1
# (name of class 1) (name of class 2) ...
(class label 1) (class label 2) (class label 2) ... (class label m)
```

An example is:

```
32 2 1
# MALE FEMALE
m m m m m m m m m m m m m m m m m m m m m m f f f f f f f f f f f f f f f f
```

In this example, there are in total 32 individuals which are divided into two groups according to their gender. The gender of each individual is presented in the last line in the file.

Continuous Phenotypic Data

If the phenotypic data is continuous, there should also be three lines in the file but they will very different from those in categorical phenotypic data file.

The first line should be:

```
#numeric
```

The second line should contains the descriptions of the phenotype. The format should be

```
 #(name of phenotype)
```

An example is

```
#Height
```

The third line should contain all the data for the samples. The format should be

```
(number 1) (number 2) ... (number m)
```

An example is

```
-0.471709930437999 1.84650009240061 ... 0.750129079744733
```

All the numbers are separated by either spaces or tabs.

In summary, the format of continuous phenotypic data file should be

```
#numeric
```

```
 #(name of phenotype)
```

```
(number 1) (number 2) ... (number m)
```

An example is

```
#numeric
```

```
 #(name of phenotype)
```

```
-0.471709930437999 1.84650009240061 ... 0.750129079744733
```

Note: currently only one phenotypic data is allowed in one file.

Output Format

Screen Output

After invoking the program and setting the parameters, the program will print the progress of the calculation on the screen. An example is

```
*****  
****          Welcome to use SubGSE v1.0          ****  
*****  
Please read the SubGSE_readme.pdf before use this program!
```

```
-----  
Data Informations:  
-----
```

```
Expression Data: 10 genes, 20 samples.  
Phenotypic Data: Categorical, 2 categories.  
Gene Sets: 2 gene sets.
```

```
-----  
Data Loading:  
-----
```

```
Gene Expression Data Loading Finished.  
Gene Set Data Loading Finished.  
Phenotypic Data Loading Finished.
```

```
-----  
Calculations:  
-----
```

```
Association strength for observed data....  
Generating permutations.....  
Association strength for permuted data....  
Matrix of local statistic of strict subset in observed and permuted data....  
Matrix of nominal p-values.....  
Matrix of minimum p-values.....  
Assessment of p-values.....  
Matrix of nominal p-values.....  
Matrix of minimum p-values.....  
Assessment of p-values.....
```

```
-----  
Results:  
-----
```

```
GENESET1          0.520000  
GENESET2          0.520000
```

```
*****  
****          Thank you for using SubGSE          ****
```

The output on the screen can be divided into four parts:

Data Information: it shows the number of genes and the number of samples in the gene expression data, the number of classes and the type of the phenotypic data, and the number of defined gene sets in the gene set data. This information is shown to make sure the correct data sets are loaded.

Data Loading: the progress of the data loading is shown in this part.

Calculations: in this part, you can see the progress of the calculations on these two parts correspondingly.

Results: in this part, the assessed p-value for each given gene set is shown as one line.

Output File

Besides the information on the screen, SubGSE will also write the assessed p-values for all the given gene sets into the file whose path was given by the user. In the result file, each line corresponds to each given gene sets. Each line in the result file will have the following format:

(gene set name) (p-value)

A typical result file is as follows:

```
GENESET1    0.52
GENESET2    0.52
GENESET3    0.23
...
GENESET10   0.32
```

The gene set name and the p-value are separated by tab.

Note: At this stage, SubGSE cannot compute the q-values from the assessed p-values automatically. If you need the q-values, please load in the p-values in the result file into R and use the QVALUE R package to obtain the q-values.

Contact Information

If you have any problem with this program, please contact:

Fengzhu Sun, PhD
Department of Biology
University of Southern California
1050 Childs Way, RRI201
Los Angeles, CA 90089-2910
(213) 740-2413 (phone)
(213) 740-8631 (fax)
fsun@usc.edu
<http://www-rcf.usc.edu/~fsun/>