Genome-Wide Gene-Gene Interaction analysis (GWGGI)

Version 1.0, 03/10/2014

Overview:

GWGGI (Genome-wide gene-gene interaction analysis) is written in c++ language and originally designed to perform gene-gene interaction analysis for genome-wide association study. It has two major functional modules: likelihood ratio mann-whitney (LRMW) and trees assembling mann-whitney (TAMW). LRMW can be used to detect the joint action and interactions of multiple genetic variants with moderate marginal effect sizes, while, TAMW can be used to detect the joint action and interactions of large number of variants with low marginal effect sizes.

Questions and Help Requests

If you have any questions or bug reports, please contact Changshuai Wei at cwei@epi.msu.edu.

General Usage:

The package includes pre-complied programs for both unix and windows system. You can also build the program from the source codes. Beside the c++ standard library, the program doesn't depend on other c++ libraries. For example, in windows, you can use visual studio to build the program. First, create an empty "win32 console application" project. Then, add the header file (.h) and the code file (.cpp) to the project, and build the program.

Under Windows, use "gwggi".
Under Unix, use "./ gwggi".
This program also include three example data files that you can play with: "example.fam", "example.bim", and "example.bed".
In this document, we assume the users are under windows system. For Unix users, just replace "gwggi" with "./ gwggi".

All the result files are in text format, which can be opened by the usual text editors.
Data files can be in text format or binary format. The binary format data are the same as plink binary format (please refer to Plink manual). The text format data are defined as follows:

1) "*.ped": Each row in the file represents a subject. There are six columns, representing family ID, Individual ID, Paternal ID, Maternal ID, Sex, and Phenotype respectively.

0	374	0	0	2	2
0	960	0	0	2	1
0	4408	0	0	2	2
0	1760	0	0	1	1
0	4493	0	0	1	2
0	4503	0	0	1	1
.	.	.	.	.	.
.	.	.	.	.	.

2) "*.map": Each row in the file represents a SNP. There are six columns, representing chromosome, SNP identifier, genetic distance (morgan), base-pair position, major allele, minor allele

1	rs910696	0	30340183	T	C
1	rs10888734	0	52038830	A	G
1	rs2747525	0	52056606	T	G
1	rs11205896	0	52063572	C	A
1	rs2077725	0	52066158	G	A
.	.	.	.	.	.
.	.	.	.	.	.

3) "*.gen": Each row represents the genotypes for a subject, coded as 0, 1 and 2 (AA, Aa and aa).

1	1	1	1	1	.	.
1	0	2	0	2	.	.
1	1	1	1	1	.	.
2	2	2	2	2	.	.
2	2	0	2	0	.	.
2	0	2	1	2	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Data format

1. GWGGI can read plink binary format, using "--bfile filename".

This will read the file **filename.fam**, **filename.bim**, **and filename.bed**.

Example:

"gwggi --bfile example --recode --out example.recode"

This will read in binary format files example.fam, example.bim, and example.bed (which is available in the package), and output text format, example.recode.ped, example.recode.map, and example.recode.gen.

2. GWGGI can also read text format, using "--file filename".
This will read the file **filename.ped**, **filename.map**, **and filename.gen**.
Example:

"gwggi --file example.recode --make-bed --out example.binary"

This will read in text format file example.recode.ped, example.recode.map, and example.recode.gen (which you can generate in the previous example), and then output the binary format, example.binary.fam, example.binary.bim, and example.binary.bed.

Likelihood Ratio Forword Mann-Whitney

1. Perform fast forward searching higher order joint association, using --lmw.

Example:

"gwggi --bfile example --lmw --out lmw.rst".

The above command will generate several files: "lmw.rst.log", "lmw.rst.scan", "lmw.rst.cv", "lmw.rst.lmw.lr", and subset data file ("lmw.rst.subset.gen", "lmw.rst.subset.map", and "lmw.rst.subset.ped").

"lmw.rst.log": log file
This file record the major event happened.

```
.....
Setting Parameters for command: gwggi --bfile example --lmw --out lmw.rst
Writing log file at [ lmw.rst.log ]
Analysis started: Wed Mar 12 14:33:56 2014

Reading map (extended format) from [ example.bim ]
472 markers to be included from [ example.bim ]
......
```

"lmw.rst.scan": result file for the first scan.
The "Geno" column record how the node is split on the snp selected. For example, "2" means the split is "2" v.s. "0, 1". The "AUC" column have the AUC value when this SNP is added into the tree model

```
#result of association scanning (0=A1A1, 1=A1A2, 2=A2A2)
#order  Chr  Name       Pos  Bp         Allel1  Allel2  Geno  AUC
1       10   rs3793790  0    50510742   G       A       2     0.528629
2       4    rs4425326  0    156326686  T       C       0     0.545203
3       9    rs1547696  0    90883940   T       C       2     0.558032
4       15   rs7181405  0    76735207   G       A       0     0.570197
```

"lmw.rst.cv": result file after the cross validation.
From the last line, we can find the AUC value and its p-value for the optimal model. The "AUC_tr" column

gives the AUC value for training data set, and the “AUC_ts” column gives the AUC value for testing data set. The “AUC_var” column gives the variance of the AUC value. The “P-value” column gives significance level compare with the null model when “AUC=0.5”.

```
#result of associatio with testing (0=A1A1, 1=A1A2,2=A2A2)

#order  Chr  …  Geno  AUC_tr    AUC_ts    AUC_var   P-value
1       10   …  2     0.528629  0.528629  0.000131  0.006263
2       4    …  0     0.545203  0.545203  0.000108  6.64E-06
3       9    …  2     0.558032  0.558032  9.15E-05  6.57E-10
```

“lmw.rst.lmw.lr”: file for the tree structure and the likelihood ratio values.

“lmw.rst.subset.gen” “lmw.rst.subset.map” and “lmw.rst.subset.ped”: the subset of data after variable selection.

More complicated example: “gwggi --bfile example --lmw --hz --hiorder 10 --maxauc 0.99 --out lmw.rst”

Trees Assembling Mann-Whitney

1. Construct assembling multiple trees to search low-marginal effect, using --tamw.

Example:

“gwggi --bfile example --tamw --showntree --out tamw.rst”.

The above command will generate several files: “tamw.rst.tamw.rst”, “tamw.rst.tamw.smr”, “tamw.rst.tamw.lr”, and “tamw.rst.log”.

“tamw.rst.tamw.rst”: result file for the overall model

The first three line have the AUC value, variance and p-value for the model. The file also have the likelihood ratio value for each subject.

```
#AUC:      0.51823
#Var:      8.64E-05
#P-value:  0.024909
#LR for each subject (. for missing)

#FID  IID   PAT  MAT  GENDER  PHENO  LRmean
0     374   0    0    2       2      1.00523
0     960   0    0    2       1      0.986573
0     4408  0    0    2       2      0.987607
.     .     .    .    .       .      .
.     .     .    .    .       .      .
```

"tamw.rst.tamw.smr": result file for the importance measurements for the selected SNPs. Each line has the importance score and selection times for the selected SNPs.

```
#Only show Predictors with selection times >1 and average Z score > 0
#Chr  Name       ...  meanZ     medianZ   Z95L      Z95U      nsel
1     rs1109374  ...  0.511922  0.735002  -1.01209  1.92409   9
1     rs2072660  ...  0.304505  0.716206  -0.7001   1.24341   10
4     rs3792211  ...  0.474873  0.425854  0.070986  0.780722  6
4     rs4425326  ...  1.38896   0.983677  -0.22487  2.82783   14
```

"tamw.rst.tamw.lr": the tree structures for the assembling model
"tamw.rst.log": log file

More complicated example: "gwggi --bfile example --tamw --converge --ntree 5000 --td-burnin --showntree --out tamw.rst"

Options&Functions

Functions

1. --bfile string, read binary file

Example: "gwggi --bfile example --recode --out example.recode"

2. --file string, read text file

Example: "gwggi --file example.recode –make-bed --out example.binary"

3. --out string, file for output

Example: "gwggi --bfile example --recode --out example.recode"

4. --lmw, Likelihood Ratio Mann-Whitney

Example: "gwggi --bfile example --lmw --out lmw.rst"

5. --split, split the data to 2/3 training and 1/3 testing part

Example:

"gwggi --bfile example --split --out split".

Then we can train the model on training data set, by "gwggi --bfile split.train --lmw --out lmw.rst", and then test the model on testing data set, by "gwggi --bfile split.test --lmw-apl lmw.rst.lmw.lr --out lmw.test.rst"

6. --lmw-apl string, apply lmw to a new data

Example:

First split the data by "gwggi --bfile example --split --out split" and get the model from training data set "gwggi --bfile split.train --lmw --out lmw.rst". Then using "gwggi --bfile split.test --lmw-apl lmw.rst.lmw.lr --out lmw.test.rst" to apply the model to the testing data set.

“lmw.test.rst.apl” is the file for the result after applying the model to the testing data set.

```
#result of associatio with testing (0=A1A1, 1=A1A2, 2=A2A2)
#order  Chr  Name       ...  Geno  AUC_tr    AUC_ts    AUC_var   P-value
1       10   rs3793790  ...  2     0.527145  0.527145  0.000401  0.087609
2       4    rs4425326  ...  0     0.561276  0.561276  0.000327  0.00035
3       9    rs1547696  ...  2     0.567865  0.567865  0.000275  2.16E-05
```

7. --tamw, Trees Assembling Mann-Whitney

Example: “gwggi --bfile example --tamw --showntree --out tamw.rst”

8. --tamw-apl, apply tamw to a new data

Example:

1) split data “gwggi --bfile example --split --out split”;
2) training data “gwggi --bfile split.train --tamw --showntree --out tamw.rst”;
3) testing data “gwggi --bfile split.test --tamw-apl tamw.rst.tamw.lr --out tamw.test.rst”

The major results are in “tamw.test.rst.tamw.rst” and “tamw.test.rst.tamw.smr”.

“tamw.test.rst.tamw.rst” is the result for overall model.

```
#AUC:      0.534633
#Var:      0.000259
#P-value:  0.015707
#LR for each subject (. for missing)

#FID  IID   PAT  MAT  GENDER  PHENO  LRmean
0     1760  0    0    1       1      1.03748
0     2608  0    0    2       2      1.12691
0     2494  0    0    1       1      1.21478
```

“tamw.test.rst.tamw.smr” is the result for the important predictor.

```
#Only show Predictors with selection times > 1 and average Z score > 0
#Chr  Name        ...  meanZ     medianZ   Z95L      Z95U      nsel
1     rs2077725   ...  0.081983  -0.02471  -1.99686  1.797     22
1     rs10888740  ...  0.340313  1.35822   -1.49192  1.94659   17
1     rs1109374   ...  0.003748  -1.72936  -1.72936  0.982941  53
1     rs2072660   ...  0.430995  -0.08905  -1.75505  2.05213   43
.     .           .    .         .         .         .         .
.     .           .    .         .         .         .         .
```

Options

1. --hz, allow heterozygosity (Aa v.s. aa,AA)
Example: “gwggi --bfile example --lmw --hz --out lmw.rst” or “gwggi --bfile example --tamw --hz --showntree --out tamw.rst”

2. --hiorder int, define the highest order for searching, default is 10
Example: “gwggi --bfile example --lmw --hiorder 15 --maxauc 0.999 --no-cv --out lmw.rst”

3. --maxauc float, define the highest auc for searching, default is 0.9
Example: “gwggi --bfile example --lmw --hiorder 15 --maxauc 0.999 --no-cv --out lmw.rst”

4. --converge, show the converging status of tamw in *.monitor file
Example: “gwggi --bfile example --tamw --converge --ntree 500 --tree-depth 8 --showntree --out tamw.rst”

5. --ntree int, number of trees in tamw, default is 2000
Example: “gwggi --bfile example --tamw --converge --ntree 500 --tree-depth 8 --showntree --out tamw.rst”

6. --tree-depth int, set tree depth, default is 6
Example: “gwggi --bfile example --tamw --converge --ntree 500 --tree-depth 8 --showntree --out tamw.rst”

7. --td-burnin, use first 50 bootstrap sample to determine the trees depth
Example: “gwggi --bfile example --tamw --converge --ntree 1500 --td-burnin --showntree --out tamw.rst”

8. --showntree, show the number of tree built as program is running
Example: “gwggi --bfile example --tamw --converge --ntree 1500 --td-burnin --showntree --out tamw.rst”

9. --clsf int, number of genetic variants randomly selected in tamw.
Example: “gwggi --bfile example --tamw --converge --ntree 1500 --td-burnin --clsf 20 --showntree --out tamw.rst”

10. --clsf-sqrt, using sqrt of total genetic variant for --clsf in tamw, default is ln
Example: “gwggi --bfile example --tamw --converge --ntree 1500 --td-burnin --clsf 20 --showntree --out tamw.rst”

11. --showcv, show the progress of the program when runing lmw
Example: “gwggi --bfile example --lmw --showcv --out lmw.rst”

12. --no-cv, scaning without cross-validation for lmw
Example: “gwggi --bfile example --lmw --no-cv --out lmw.rst”