assoc-aSPU 1.0 is a program based on Linux command line operations. Main executable is 'assoc-aSPU.r'. All useful R functions are put in the root folder together with 'assoc-aSPU.r'. When assoc-aSPU is executed with proper inputs and options, all necessary functions there will be called. There are several folders under the root path, each of which stores the sample input data in Rdata binary format (required. If your raw data is in txt, please transform to Rdata binary format). Results from the program will be save to folder "OutputData". If any error occurs during the program run, the error message together with normal output messages of the program will be logged in a '\*.log' file under program root path. In the same time, an '\*.rda' file containing R runtime environment will be saved in the 'OutputData' format to help error debugging.

Flowchart of 1st time running the program:

1. install all the packages as in "head.r" manually.
2. change the first line of 'assoc-aSPU.r',

#!/work/02040/yyang/Software/R/bin/Rscript --slave --vanilla

to

#!/your-path-of-Rscript --slave --vanilla

1. $ chmod u+x assoc-aSPU.r

to add executable privilege to current user.

1. $ ./assoc-aSPU.r -h

to see help documents of arguments you can provide

1. check 'Demo' folder

'ExampleData\_demo.txt': shows the format of all kinds of input data you need to prepare.

'commandLine\_demo.sh': shows the example command line using the example data.

1. you are ready to make your own data and run the program.

# Some Usage Notes:

1. for fast speed, you want to use '--include\_aSPU FALSE' to only include five asymptotic tests (which are fast).
2. when including aSPU family tests, you'd better specify to use parallel computing in '--parallel\_scheme' argument.
3. check the '--parallel\_scheme' and '--parallel\_over\_gene' explanations using '$ ./assoc-aSPU.r -h ' to better figure out a parallel computing scheme for your dataset.
4. when enable '--usePermutedU TRUE' for aSPU family tests, you will wait for longer time since permutation is much slower than simulation based method.
5. If you have multiple nodes available, check '--jobExecuteOnMultipleNodes' and '--MultipleNodes\_profile' explanations to learn how to enable parallel run on multiple nodes with multiple cores. This is usually only necessary when you want to run aSPU family tests with '--usePermutedU TRUE'.