MLRT Software: Binomial Mixture Model-based Association Tests under Genetic Heterogeneity

Hui Zhou, Wei Pan

Division of Biostatistics, School of Public Health, University of Minnesota, Minneapolis, MN 55455

Email: zhoux292@umn.edu, weip@biostat.umn.edu

June 25, 2009

We desribe a computer program that implements the mixture likelihood ratio test (MLRT) as discussed in Zhou and Pan (2009) to test for disease association under genetic heterogeneity based on case-control data with possibly multiple genotyped SNPs. The main body of the program is written in C++, however, the user can call it in R.

1 Input

Suppose there are n cases and m controls. Each individual has $nSNP$ genotyped, with a possible "causal" (in simulations) SNP number *disease_causing*, which would be excluded when calculating the Max and Mean (equivalent to the Sum) MLRT statistics; if no causal SNP is present, just simply specify *disease_causing* as any number outside 1 to $nSNP$. Each SNP carries a value of 0, 1 or 2, the number of copies of one allele at the locus. A part of the input includes a data matrix with $n+m$ rows and $nSNP$ columns; each individual's genotype scores are in a row, and the first n rows are for the cases while the next m rows are for the controls. Suppose $SNP = 5$, then the data matrix should look like the following:

$$
\text{n rows}\left\{\begin{array}{ccccc} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 2 & 1 & 0 & 0 \end{array}\right.
$$

$$
\text{m rows}\left\{\begin{array}{ccccc} 1 & 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 0 & 2 \end{array}\right.
$$

A parameter called *Boot* is to be specified by the user as the number of permutations to to be used to obtain permutation p-values.

2 Output

p_value_Mean_MLRT	p-value for MEAN-MLRT
p_value_Max_MLRT	p-value for MAX-MLRT
MLRT_MAX	test-statistic of MAX-MLRT
MLRT_MEAN	test-statistic of MEAN-MLRT
MLRT_STAT	test-statistic of MLRT for each SNP
theta	estimated θ
theta_b	estimated background probabity θ_b
alpha	estimated mixing proportion

3 Example

First of all, to compile the C++ code, type at the command prompt

```
R CMD SHLIB C_method_1.cpp
```

where C_method_1.cpp is the name of the supplied C++ program, which of course can be differently named. The compiled file name is C_method_1.so. At the beginning of the user's R program, include the following line to upload the compiled C++ program:

```
dyn.load("C_method_1.so")
```

Now one can call it in R.

Suppose we already have a genotype score matrix in R, then do the following:

```
x <- t(data)
n_case <- 500
n_control <- 500
SNP <- 5
disease_causing <- 1
Boot <- 200

p_value_Max_MLRT <- 0
p_value_Mean_MLRT <- 0
MLRT_MAX <- 0
MLRT_MEAN <- 0
MLRT_STAT <- rep(0,SNP)
theta <- 0
theta_b  <- 0
alpha <- 0

out <- .C("MLRT", x=as.integer(x), n_case=as.integer(n_case),
n_control=as.integer(n_control),SNP=as.integer(SNP),
disease_causing=as.integer(disease_causing),
Boot=as.integer(Boot),p_value_Max_MLRT=as.double(p_value_Max_MLRT),
```

```
p_value_Mean_MLRT=as.double(p_value_Mean_MLRT),MLRT_MAX=as.numeric(MLRT_MAX),
        MLRT_MEAN=as.numeric(MLRT_MEAN),MLRT_STAT=as.numeric(MLRT_STAT),
theta=as.numeric(theta),theta_b=as.numeric(theta_b),alpha=as.numeric(alpha))

out$p_value_Mean_MLRT
out$p_value_Max_MLRT
out$MLRT_MAX
out$MLRT_MEAN
out$MLRT_STAT
out$theta
out$theta_b
out$alpha
```

The first six lines tell the C++ program the data matrix, numbers of cases and controls, number of total SNPs, the causal SNP number and permutation number. Here, we have 500 cases, 500 controls and each subject has five SNPs of interest with the first one as the causal SNP. Note that if none of the SNPs are causal, then simply assign *disease_causing* an integer value that is larger than SNP or less than 0. The next eight lines in front of the ".C" statement serve as place holders to store the output, and their values can be arbitrary; here to 0's are used. The ".C" statement passes R objects to C++ forcing correct R storage mode to prevent mismatching of data types. The final eight lines following the ".C" statement would print out their values within the list object "out".

References

[1] Zhou H, Pan W (2009). Binomial Mixture Model-based Association Tests under Genetic Heterogeneity. Submitted to *Annals of Human Genetics*.