

Instruction

October 10, 2003

Please address comments and suggestions to Jung-Ying Tzeng at jytzeng@stat.ncsu.edu

1 What the R codes do

These R functions perform the analysis described in Tzeng et al. (2003 AJHG) and Tzeng et al. (2003 JASA) to detect the potential haplotype regions that may underlie the complex trait of interests.

The inputs are the haplotype frequency tables of cases and controls. One can obtain the frequencies from certain algorithms such as EM, PHASE or other algorithms. In the R functions, we first calculate three types of the Quadratic Statistics of Haplotype Similarity (QSHS; i.e., the $\Pi^T A \Pi$ statistic in the AJHG paper)—match statistic, length statistic, and count statistic. These statistics are calculated either by a full-dimension approach or by both full-dimension and reduced-dimension approaches (see the AJHG paper for details). After obtaining the QSHS for each haplotype region, we use Genomic Control to adjust for the confounding effect caused by relatedness among haplotypes and/or population substructure (see the JASA paper for details). Then we report the standardized QSHS (which we called “Z”-statistics) and p -values for each region, and the regions that are statistically significant based on the FDR controlling procedure.

2 References

Tzeng, J.Y., Byerley, W., Devlin, B., Roeder, K. and Wasserman, L. (2003). Outlier detection and false discovery rates for whole-genome DNA matching. *Journal of the American Statistical Association*, 98:236-246.

Tzeng, J.Y., Devlin, B., Wasserman, L. and Roeder, K. (2003). On the identification of disease mutations by the analysis of haplotype similarity and goodness-of-fit. *The American Journal of Human Genetics*, 72:891-902.

3 List of notations

- K = number of haplotype regions in the genome scan
- R = number of haplotype categories (i.e., number of distinct haplotypes)
- R^* = number of haplotypes categories that one wishes to reserved in the reduced-dimension analysis

4 How to use the functions in R

- Step 0: Download and install R. R is a free statistical software similar to Splus. It is available at <http://www.r-project.org/>.
- Step 1: Download the file `QSHS.FD.RD.r` from <http://www4.stat.ncsu.edu/~tzeng/QSHS/Rcodes/>. Save the file in the working directory.
- Step 2: Prepare the files of haplotype frequencies for cases and controls in the format of an $R \times 2$ table. The first column is haplotype, in which the allele of each locus is coded by a 2-digit number. Note the number of digits for allele codings must be constant across loci, but not necessary to be 2. The second column is haplotype frequency. Here are some example files:

```
Filename: hapfreq.cs1
4851495149 0.046409639
4851495150 0.004000000
4851515149 0.004000000
4950495149 0.012000000
.
```

```
Filename: hapfreq.cn1
4951485149 0.004000000
4951495049 0.005204819
4951495148 0.008000000
.
```

Assume that we have 3 haplotype regions in total. Please name the frequency files as “filename-for-cs1”, “filename-for-cn1”, and “filename-for-cs2”, “filename-for-cn2”, and “filename-for-cs3”, “filename-for-cn3”.

There are some example files available from <http://www4.stat.ncsu.edu/~tzeng/QSHS/Rcodes/>. They are `hapfreq.cs1`, `hapfreq.cn1`, `hapfreq.cs2`, `hapfreq.cn2`, `hapfreq.cs3` and `hapfreq.cn3`. From now on I will use these example files to illustrate the usage of these functions.

- Step 3: In R, type

```
> source('‘QSHS.FD.RD.r’')
```
- Step 4: To identify the regions harboring outliers, we need to run two main functions:
 1. `QSHStest.fun` — calculate QSHS by full-dim approaches or both full/reduced-dim approaches; for details please refer to the AJHG paper.

2. `getresult.fun` — obtain the Z statistics (i.e., the standardized QSHS) using genomic control to adjust for confounders and using FDR procedure to adjust for multiple testing; for details please refer to the JASA paper.

Before explaining the arguments, here is a quick example of what to type in R:

```
> qshs <- apply(as.matrix(1:3), 1, QSHStest.fun,
               csfile="hapfreq.cs", cnfile="hapfreq.cn",
               loci=5, nn=200, mm=200, cc.RD=0.2, Rstar=NULL)
> getresult.fun(qshs,type=3)
```

□ Arguments of `QSHStest.fun`

- `csfile`: filenames for hap freq of cases, e.g. `hapfreq.cs` in the example
- `cnfile`: filenames for hap freq of controls, e.g. `hapfreq.cn` in the example
- `fileIndex`: it can be “NULL” (if there is only one hap region to study) or a number (if there are more than 1 haplotype regions needed to be analyzed). In our example there are 3 regions, so `fileIndex= 1, 2, or 3`. To calculate QSHS for the 2nd region only, type

```
> QSHStest.fun(csfile="hapfreq.cs", cnfile="hapfreq.cn", fileIndex=2,
              loci=5, nn=200, mm=200, cc.RD=NULL, Rstar=NULL)
```

To get QSHS for for all 3 regions, one can either do a for-loop on `fileIndex` or use `apply` as shown here.

```
> apply(as.matrix(1:3),1,QSHStest.fun, csfile="hapfreq.cs", cnfile="hapfreq.cn",
       loci=5, nn=200, mm=200, cc.RD=NULL, Rstar=NULL)
```

ps. Here using for-loop on `fileIndex` may take about the same time as using `apply`. This is because I’ve used `apply` when defining some of the functions and in R double `apply` wont be any faster than `loop+apply` or double loop. If for-loop is used, remember to store the regional result by column (instead of by row)!

- `loci`: number of loci per haplotype region
- `nn`: number of case haplotypes
- `mm`: number of control haplotypes
- `cc.RD`: (i.e., c^* in the AJHG paper) It is a number $\in [0, 1]$; it specifies the cut-off criteria used in the reduced-dimension (RD) method. Those haplotype categories with frequency $\pi \geq cc.RD \times \pi_{max}$ would be reserved.
- `Rstar`: (i.e. R^* in the AJHG paper) It’s an integer. Instead of using `cc.RD` to determine which categories to reserved, one can directly specify the highest `Rstar` categories to be reserved in the RD method.

NOTE that `cc.RD` and `Rstar` together also control what approach we will use to calculate QSHS:

If one wish to perform a full-dim analysis only, then assign `cc.RD=NULL` and `Rstar=NULL`.

If one wish to perform a full-dim analysis and a reduced-dim analysis using $c^*\pi_{max}$ cut-off, then assign `cc.RD=(say)0.2` and `Rstar=NULL`.

If one wish to perform a full-dim analysis and a reduced-dim analysis with a fixed R^* , then assign `cc.RD=NULL` and `Rstar=(say)8`.

□ Arguments of `getresult.fun`

- `qshs.result`: a $6 \times K$ or $12 \times K$ matrix output from `QSHStest.fun`. It is $6 \times K$ if only a full-dim analysis is performed. It's $12 \times K$ if both full-dim and reduced-dim approaches were used.
- `type`: $\in \{1, 2, 3\}$. 1=match statistic; 2= length statistic; 3= count statistic.

5 Returned values and examples

Example I (multiple regions; Genomic Control & FDR procedure to detect mutation)

1. copy `QSHS.FD.RD.r` (the source code) and the example data files: `hapfreq.cs1` to `hapfreq.cs3`
`hapfreq.cn1` to `hapfreq.cn3`

2. in R, type:

```
> source("QSHS.FD.RD.r")
> qshs<-apply(as.matrix(1:3),1,QSHStest.fun, csfile="hapfreq.cs",
              cnfile="hapfreq.cn", loci=5, nn=200, mm=200,
              cc.RD=0.2, Rstar=NULL)
```

```
> qshs
```

Warning messages:

1: NAs introduced by coercion

2: NAs introduced by coercion

3: NAs introduced by coercion

4: NAs introduced by coercion

5: NAs introduced by coercion

6: NAs introduced by coercion #these warning messages are normal; one should expect to get $2 \times K$ such warnings

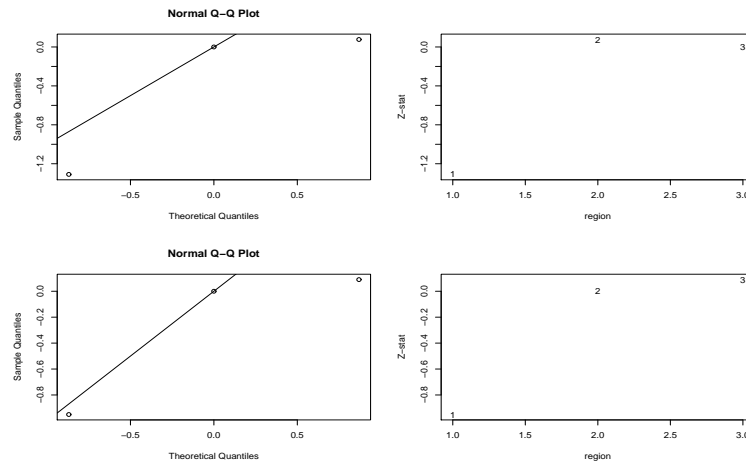
	region 1	region 2	region 3	
mtch	0.0028984255	0.0207122579	0.0229081600	#match stat by full-dim approach
len	0.1326394059	0.1974848741	0.3562220800	#length stat by full-dim approach
ct	0.1860585201	0.3175514681	0.3080070400	#count stat by full-dim approach
var.mtch	0.0001046128	0.0004466133	0.0001304666	#variance of the match stat
var.len	0.0055902168	0.0107979180	0.0203082416	#variance of the length stat
var.ct	0.0067585282	0.0120631875	0.0206738161	#variance of the count stat
mtch.rd	0.0115989489	0.1119812336	0.0007430400	#match stat by reduced-dim approach
len.rd	0.2201941367	0.4551034970	0.4632320000	#length stat by reduced-dim approach
ct.rd	0.2561867774	0.4551034970	0.4792140800	#count stat by reduced-dim approach
var.rd.mtch	0.0001794867	0.0012357990	0.0002094831	#variance of mtch.rd
var.rd.len	0.0228871567	0.0350438867	0.0484044302	#variance of len.rd
var.rd.ct	0.0307437873	0.0350438867	0.0507183490	#variance of ct.rd

```
> getresult.fun(qshs,type=3)
```

```
$result.FD          #results for full-dim approach
$result.FD$mu.med   # $\mu$  in the JASA paper; the baseline difference of similarity b/w cases and controls
0.3080070
$result.FD$tau      # $\tau$  in the JASA paper; the inflation due to confounders
1.132744
$result.FD$Z        #the standardized QSHS (here it's the count statistics because type=3)
-1.30953995 0.07671621 0.00000000
$result.FD$pvalue   #p-values
0.9048242 0.4694247 0.5000000
$result.FD$"No. of outliers"    #number of outliers
0
$result.FD$"Regions with outliers" #list of regions that contain outliers
NA
```

```
$result.RD          #results for reduced-dim approach
$result.RD$mu.med
0.4551035
$result.RD$tau
1.193124
$result.RD$Z
-0.95083890 0.00000000 0.08973038
$result.RD$pvalue
0.8291569 0.5000000 0.4642507
$result.RD$"No. of outliers"
0
$result.RD$"Regions with outliers"
NA
```

Note that `getresult.fun` also returns a normal Q-Q plot of the Z-statistics for the normality check. It also draw a Z-stat vs. region plot; if there exit outliers, then this plot would also have a horizontal line indicating the threshold of significance.



Example II (one region only; can't do Genomic Control & no need for FDR procedure)

If only one hap region only, then the output of `QSHStest.fun` is a vector with length 6 or 12. `getresult.fun` will return all 3 types of QSHS Z statistics regardless of the value of `type`.

1. copy `QSHS.FD.RD.r` (the source code) and two example data files: `hapfreq.cs` and `hapfreq.cn`
2. in R, type:

```
> source("QSHS.FD.RD.r")
> qshs<- QSHStest.fun(csfile="hapfreq.cs", cnfile="hapfreq.cn", fileIndex=NULL,
                    loci=5, nn=200, mm=200, cc.RD=0.2, Rstar=NULL)
> qshs      #the output is a 12 x 1 vector
```

Warning messages:

- 1: NAs introduced by coercion
- 2: NAs introduced by coercion

```
      mtch      len      ct      var.mtch      var.len      var.ct
0.0028984255 0.1326394059 0.1860585201 0.0001046128 0.0055902168 0.0067585282
      mtch.rd      len.rd      ct.rd      var.rd.mtch      var.rd.len      var.rd.ct
0.0115989489 0.2201941367 0.2561867774 0.0001794867 0.0228871567 0.0307437873
```

```
> getresult.fun(qshs)      #regardless of the value of type, it returns all 3 types of QSHS
      Z      pvalue
```

mtch	0.2833803	0.38844265
len	1.7740189	0.03803001
ct	2.2632029	0.01181159
rd.mtch	0.8657698	0.19330820
rd.len	1.4554920	0.07276655
rd.ct	1.4610936	0.07199488