TREAT

March 24, 2014

Introduction

The TREAT package is designed to enhance statistical power for testing non-additive association between SNP sets and binary outcome in case-control studies. Various environmental and genetic covariates can be properly adjusted, as long as they are categorical or can be discretized into factors. The statistical test implemented in TREAT is complementary to those commonly used multi-locus tests that assume an additive risk model. TREAT is a tree-based test. The state-of-the-art Boolean operation method is adopted to accelerate the tree-building procedure, which makes TREAT applicable to large-scale association studies, e.g., genome-wide association studies.

```
> library(TREAT)
```

Testing association between SNP sets and a binary outcome

For the treat function, the input data must be in a data frame. The data frame should include a binary outcome (y), covariates to be adjusted (covar) and the SNPs to be tested (SNPs). Note that TREAT only accepts categorical covariates. If you have a continuous covariate, e.g., an eigenvector computed from the GWAS data, we recommend discretizing it into a factor variable X with a few levels. To do so, you may fit a simple logistic regression model on factor X and other covariates, then collapse those levels with similar effects on the outcome. The SNPs data must be coded as 0, 1, and 2.

```
> set.seed(123)
> y <- c(rep(0, 1000), rep(1, 1000))
> covar <- sample(0:2, 2000, replace = TRUE, prob = rep(1/3, 3))
> SNPs <- matrix(rbinom(2000 * 10, 2, c(0.36, 0.48, 0.16)), nrow = 2000)
> SNPs <- as.data.frame(SNPs)
> colnames(SNPs) <- paste("rs", 1:10, sep = "")
> data <- data.frame(y, covar, SNPs)
```

To test an association, simply call the treat function as below. You can specify the outcome and covariates in a formula. The data frame is passed into the function. The association between the outcome and those SNPs specified in snp.vars is then tested. The function treat also accepts options (op) to control the tree-building procedure. Please refer to manual for more information.

```
> tr <- treat(y ~ factor(covar), data = data, snp.vars = colnames(SNPs))

NOTE: The TREAT package is ONLY for 64-bit operating systems
```

The function returns a p-value adjusted for multiple-comparisons. Some details about how the tree is built based on the observed data are also provided in $tree.model.forward. For example, according to SNP rs2, node 1 (root node) is split as node 2 (if rs2 = 0 or 1) and node 3 (if rs2 = 2).

```
> tr$adj.pval

[1] 0.5954046

> tr$tree.model.forward

  splitting rule parent child1 (rule = FALSE) child2 (rule = TRUE)
1          rs2=2      1                     2                    3
2          rs4=2      2                     4                    5
3          rs2=0      5                     6                    7
4          rs6=0      6                     8                    9
```

PLINK data fromat

The TREAT package can load genotype data from plink binary files directly. This feature is particular useful when you are planning to perform a genome-wide gene-based analysis. The binary PED file (*.bed), the pedigree/phenotype information file (*.fam) and the extended MAP file (*.bim) are required in function treat.plink. Please refer to the website of PLINK for more information about how to create these files from the PED file and MAP file. The TREAT package provides an example dataset in plink format, which were converted from the example provided by PLINK (http://pngu.mgh.harvard.edu/ purcell/plink/dist/example.zip).

Get the data files from TREAT package

```
> bed <- system.file("sampleData", "data.bed", package = "TREAT")
> bim <- system.file("sampleData", "data.bim", package = "TREAT")
> fam <- system.file("sampleData", "data.fam", package = "TREAT")
```

A data frame containing the binary outcome, covariates and id variables is required by the function treat.plink. The id variables (e.g., "FAMILY", "SUBJECT") are used to describe pedigree information. Here we load them from the *fam file.

```
> x <- read.table(fam, header = 0, stringsAsFactors = FALSE)
> colnames(x) <- c("FAMILY", "SUBJECT", "FATHER", "MOTHER", "SEX",
+     "Y")
```

The function treat.plink allows testing all given gene sets sequentially. Here we specify two genes by constructing a gene object gene.obj. It is a vector of gene names or a matrix with at least 3 columns called "Chr", "Start" and "Stop" which give the chromosome, starting location (physical position in base units) and ending location for each gene to analyze. This matrix can also have a column called "Gene" for the name of genes which is used in creating the names

of the output files. Note that the format of gene.obj used here is very similar with that generate from PLINK (see section "Extract a subset of SNPs: file-list options" in http://pngu.mgh.harvard.edu/ purcell/plink/dataman.shtml)

```
> gene.obj <- rbind(c(8, 12799052, 12895289, "Gene1"), c(8, 12868315,
+     12989321, "Gene2"))
> colnames(gene.obj) <- c("Chr", "Start", "Stop", "Gene")
```

Create the formula object, vector of id variables and call the treat.plink function. Note that the example data shown here has small sample size, we therefore modify the options to grow a larger tree, which will illustrate more details of the output.

```
> formula <- as.formula(Y ~ SEX)
> id.vars <- c("FAMILY", "SUBJECT")
> op <- list(thr.sample = 10, thr.leaf.size = 5)
> tr <- treat.plink(c(bed, bim, fam), formula, x, id.vars, gene.obj,
+     op = op)

NOTE: The TREAT package is ONLY for 64-bit operating systems
NOTE: The TREAT package is ONLY for 64-bit operating systems
```

The function will return a list which contains a character vector of the output files created. Each output file is an R object file in directory op$out.dir with name <gene>.rda containing the returned object from treat. The R object can be loaded back into R, from which a list with name obj can be obtained.

```
> tr$summary

   Gene    Pvalue N.Marker.All N.Marker.Test
1 Gene1 0.3396603           10            10
2 Gene2 0.5844156           15            15
                                 Best.SNPs
1  rs2460911,rs11203962,rs754238,rs7835221
2 rs17786052,rs2460911,rs607499,rs11203962

> load(tr$saved.files["Gene1"])
> obj$adj.pval

[1] 0.3396603

> obj$tree.model.forward

  splitting rule parent child1 (rule = FALSE) child2 (rule = TRUE)
1    rs2460911=0      1                     2                    3
2   rs11203962=0      3                     4                    5
3     rs754238=0      2                     6                    7
4    rs7835221=0      7                     8                    9
```

Session Information

```
> sessionInfo()
```

```
R version 2.13.2 (2011-09-30)
Platform: x86_64-pc-mingw32/x64 (64-bit)

locale:
[1] LC_COLLATE=C
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] splines   stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
[1] TREAT_0.0.5     snpStats_1.2.1  Matrix_1.0-4    lattice_0.19-33
[5] survival_2.36-9

loaded via a namespace (and not attached):
[1] grid_2.13.2  tools_2.13.2
```