Extreme phenotype sampling

Thea Bjornland

3 Dec 18

This package provides functions for testing hypothesis of genetic association with a continously measurable trait when data are from the extreme phenotype sampling (EPS) design.

```
library(extremesampling)
```

The assumed linear regression model is of the form

$$\mathbf{y} = X\beta + G\beta_g + \varepsilon,$$

where y is a phenotype vector (continuosly measurable trait), X is a matrix of non-genetic/environmental covariates (including intercept) and G is a matrix of genetic covariates. The vector ε is multivariate normal with mean 0 and covariance $I\sigma^2$ (I is the identity matrix).

Hypothesis tests

We have implemented score tests for testing $H_0 : \beta_g = 0$ against $H_0 : \beta_g \neq 0$, for two special situations

1. Test one genetic covariate (one column of G) at a time (the number of columns of G can then be very large).
2. Test all genetic variants simultaneously (then then number of columns of G should be relatively small).

We consider two types of data analysis

1. Complete case analysis. The response y_i, non-genetic covariates $\mathbf{x}_i$ and genetic covarirates $\mathbf{g}_i$ are observed only for extreme phenotype individuals, i.e. individuals who satisfy $y_i < l_i$ or $y_i > u_i$, where thresholds l_i and u_i are known, but might be different for each individual in the sample.
2. All case analysis. The response y_i and non-genetic covariates $\mathbf{x}_i$ are observed for a full sample (randomly drawn from infinite population) while the genetic covarirates $\mathbf{g}_i$ are observed only for extreme phenotype individuals.

For complete case analysis, the thresholds must be supplied by the user.

For all case analysis, the genotype matrix (or vector) must be of the same length as $\mathbf{y}$ and coded as NA where genotypes are not observed (non-extreme individuals). Further, it is assumed that the genotype is a discrete variable. The genotype distribution can be group-specific, and a matrix (dummy coded categorical covariates) that indicates these groups must be supplied.

Example data

We simulate data to illustrate score test functions. Extreme phenotype thresholds are set accoriding to the quantiles of the residual phenotype distribution (z-extreme sampling).

```
N = 5000 # Number of individuals in a population
xe1 = rnorm(n = N, mean = 2, sd = 1) # Environmental covariate
xe2 = rbinom(n = N, size = 1, prob = 0.3) # Environmental covariate
Gmat = cbind(rbinom(N,2,0.4),rbinom(N,2,0.1),rbinom(N,2,0.25),
             rbinom(N,2,0.01),rbinom(N,2,0.01),rbinom(N,2,0.01)) # Genetic variants
```

```r
# Generate phenotype
betag = c(0.1,-0.1,0.3,0.5,-0.2,0)
y = rnorm(N, mean = 10+5*xe1+1*xe2+Gmat%*%betag, sd = 4)

# Identify extremes, here upper and lower 25% of residual phenotype distribution
z = lm(y~xe1+xe2)$residuals
uz = quantile(z,probs = 3/4,na.rm=TRUE)
lz = quantile(z,probs = 1/4,na.rm=TRUE)
extreme = (z < lz) | (z >= uz)
li = lz + (y[extreme]-z[extreme]); ui = uz + (y[extreme]-z[extreme])

# Create the EPS complete case data set
yCC = y[extreme]; zCC = z[extreme]
xe1CC = xe1[extreme]; xe2CC = xe2[extreme]
GmatCC = Gmat[extreme,]

# Create the EPS all case data set
yAC = y; zAC = z
xe1AC = xe1; xe2AC = xe2
GmatAC = Gmat; GmatAC[!extreme,] = NA
```

Single variant association tests

Each of the six columns of the genetic covariate matrix G are tested one at a time against the null model $\mathbf{y} = X\beta + \varepsilon$.

Complete case tests

```r
cc = epsCC.test(yCC~xe1CC + xe2CC, xg = GmatCC, l = li, u = ui)
```

```r
print(cc)
```

```
## $statistic
##             t
## xg1 3.4070532
## xg2 0.1722763
## xg3 7.7851635
## xg4 5.2260897
## xg5 0.2498471
## xg6 0.6854948
##
## $p.value
##         p.value
## xg1 0.064918277
## xg2 0.678096687
## xg3 0.005267702
## xg4 0.022250501
## xg5 0.617182782
## xg6 0.407700996
```

All case tests

```
ac = epsAC.test(yAC~xe1AC + xe2AC, xg = GmatAC)
```

```
print(ac)
```

```
## $statistic
##             t
## xg1 3.4004811
## xg2 0.1671243
## xg3 7.7631329
## xg4 5.2432454
## xg5 0.2551514
## xg6 0.6698486
##
## $p.value
##         p.value
## xg1 0.065177407
## xg2 0.682680249
## xg3 0.005332339
## xg4 0.022032136
## xg5 0.613470949
## xg6 0.413104409
```

Multiple variant association tests

Complete case tests

```
# Direct test of all 6 variants
cc_d = epsCC.rv.test(yCC~xe1CC + xe2CC, xg = GmatCC, l = li, u = ui)$p.value

# Collapsing test (weights = 1 for all variants)
cc_c = epsCC.rv.test(yCC~xe1CC + xe2CC, xg = GmatCC, l = li, u = ui, method = "collapse")$p.value

# Variance component test (weights = 1 for all variants)
cc_v = epsCC.rv.test(yCC~xe1CC + xe2CC, xg = GmatCC, l = li, u = ui, method = "varcomp")$p.value

ccall = data.frame(cbind(cc_d, cc_c, cc_v))
names(ccall) = c("CC", "CC collapse", "CC varcomp")
```

```
print(ccall)
```

```
##              CC CC collapse  CC varcomp
## xg1 0.009845225   0.4514474 0.005630768
```

All case tests

```
# Direct test of all 6 variants
ac_d = epsAC.rv.test(yAC~xe1AC + xe2AC, xg = GmatAC)$p.value

# Collapsing test (weights = 1 for all variants)
ac_c = epsAC.rv.test(yAC~xe1AC + xe2AC, xg = GmatAC, method = "collapse")$p.value
```

```r
# Variance component test (weights = 1 for all variants)
ac_v = epsAC.rv.test(yAC~xe1AC + xe2AC, xg = GmatAC, method = "varcomp")$p.value

acall = data.frame(cbind(ac_d, ac_c, ac_v))
names(acall) = c("AC", "AC collapse", "AC varcomp")
```

```r
print(acall)
```

```
##              AC AC collapse  AC varcomp
## xg1 0.009931397   0.4504835 0.005712189
```