MGF-R test: adaptive decorrelation procedure for signal detection

Florian Hébert, Mathieu Emily, David Causeur

1 Introduction

This vignette provides help and examples of usage of the functions of the R package `MGFRTest`. This package aims at computing the p-value of the MGF-R (Moment Generating Function - Ratio) test, which is based on an adaptive decorrelation procedure.

GWAS aim at identifying genetic markers (such as Single Nucleotide Polymorphisms or SNPs) associated to a disease. Due to the great size of the genome, SNP-sets or genes, which correspond to groups of neighboring SNPs, are often considered. First, for a given gene, a test statistic for the association between each SNP and the disease is computed. Then, the individual test statistics are aggregated to construct a global test at the gene scale. This package provides functions to compute the p-value of the MGF-R (Moment Generating Function Ratio) test. The p-value is computed using permutations of the phenotype.

In the following, examples of usage of the functions of this package are given in the GWAS context. However, it can be used more generally for testing the nullity of the mean vector of a multivariate normal distribution.

2 Computing the Vector of Association Test Statistics Between a Gene and a Phenotype

Here, `X` is assumed to be a matrix of n rows and p columns, each row corresponding to an individual and each column to a SNP. Typically, `X` is a matrix of genotypes corresponding to a gene. It takes the values 0, 1 or 2, corresponding to the number of copies of the minorallele. `Y` is a vector giving the disease status, 0 or 1 for a control or a case, respectively. An optional covariates matrix `U` of size $n \times q$ can be given. Each row of `U` corresponds to an individual and each column to a covariate.

First, to compute the vector of test statistics between each SNP of the gene and the phenotype, the function `ScoreTest` can be used. It takes as arguments the SNP matrix `X`, the vector of disease status `Y`, the optional covariates matrix `U`, an optional matrix `Y0` containing permuted versions of `Y` and a number of permutations `N` (default to 1,000). If `Y0` is given, each column must be a permutation of `Y`. It can be used for example to use specific resampling methods, such as parametric bootstrap. If `Y0` is not given, `N` permutations of `Y` are computed. The result is a list containing 3 elements named `Z`, `Z0` and `Sigma`. `Z` is the vector of test statistics, `Z0` is the matrix containing on its i-th row the vector of test statistics corresponding to the i-th permutation of `Y` and `Sigma` is the estimated correlation matrix of `Z`.

As an example, we generate a matrix of genotype values for 10 SNPs and 2000 individuals (for simplicity, no dependence is introduced between the SNPs), and the vector of disease status:

```
X = matrix(sample(0:2,2000*10,TRUE),ncol=10)
Y = sample(c(rep(0,1000),rep(1,1000)))
```

The two following commands ar equivalent (and give exactly the same result if the random seed is fixed):

```
res1 = ScoreTest(X=X,Y=Y,U=NULL,Y0=NULL,N=1000)
res2 = ScoreTest(X=X,Y=Y,U=NULL,Y0=sapply(1:1000,function(i)sample(Y)))
```

3 Computing the p-value of the MGF-R test

The p-value of the MGF-R test can be computed by using the `MGFR` function. Its arguments are the vector `Z` of test statistics, the matrix `Z0` of test statistics corresponding to the permutations of `Y` and the correlation matrix `Sigma`.The following commands compute the p-value with 1,000 permutations:

```
res = ScoreTest(X=X,Y=Y,U=NULL,Y0=NULL,N=1000)
MGFR(res$Z,res$Z0,res$Sigma)
```

Instead of `Sigma`, the eigenvalue decomposition of `Sigma` can be given using the `eigSigma` argument. This can be useful for instance for using only the first K eigenvalues.

```
ev = eigen(Z$Sigma)
MGFR(res$Z,res$Z0,eigSigma=ev)
```