Generalized Berk-Jones (GBJ) Tutorial

Ryan Sun

2018-06-27

The GBJ package implements the Generalized Berk-Jones (GBJ) test for set-based inference in genetic association studies.

Also included in this package are routines to perform the Generalized Higher Criticism (GHC), Higher Criticism (HC), Berk-Jones (BJ), and Minimum p-value (minP) tests. Some guidance on choosing between these methods (which, in principal, test the same null hypothesis and could be used interchangeably) is also given in the above paper.

The remainder of this vignette provides:

- A high-level explanation of the problems that GBJ/GHC/HC/BJ/minP can solve.
- A worked example in the context of Genome-Wide Association Studies (GWAS) where individual-level genotype data is available.
- A worked example in the context of GWAS where only single SNP summary statistics are available.

Uses for GBJ

We find it instructional to begin with a short anecdote from the class notes of John Tukey (Donoho and Jin, 2004):

> "A young psychologist administers many hypothesis tests as part of a research project, and finds that, of 250 tests 11 were significant at the 5% level. The young researcher feels very proud of this fact and is ready to make a big deal about it, until a senior research suggests that one would expect 12.5 significant tests even in the purely null case, merely by chance. In that sense, finding only 11 significant results is actually somewhat disappointing! … [Tukey] then proposed a sort of *second-level significance testing*, … [to] indicate a kind of *significance of the overall body of tests*."

All the tests performed in this package are designed to carry out the sort of second-level significance testing suggested in the story. That is, they are designed to test if there is at least one non-null hypothesis in the entire group of hypotheses. In statistical terms, assume that the researcher above had calculated 250 Z-scores, where \(Z_{i}\) has mean \(\mu_{i}\) and variance 1. Then the GBJ null hypothesis is that \(\mu_{i}=0\) for all \(i\), and the GBJ alternative is that \(\mu_{i} \neq 0\) for at least one \(i\).

GBJ can be used to test either an entire collection of hypotheses, or it may make more sense to partition the group into smaller, predefined sets and then apply GBJ multiple times. For example, in the case of GWAS, we can group the individual SNP test statistics into genes/pathways and apply GBJ to each gene or pathway. In this case, GBJ is testing if the entire gene has any association with the outcome.

Notable advantages to using GBJ are:

1. GBJ is a generalization of the Berk-Jones test, which is known to be optimal - in a certain sense - for detecting rare and weak signals when factors in a set are independent. This is clearly a very relevant guarantee for the genetics setting. GBJ modifies Berk-Jones to provide better finite sample rejection regions when factors in a set are correlated.
2. Analytic calculation of p-values (no need for permutation).
3. No tuning parameters. Standard inputs (similar to SKAT). Reasonable to apply GBJ on sets ranging from 2 to 2000 factors.
    a. Theoretically there is no upper limit, but we are currently bound by the numerical precision in standard C++ routines. There are plans to remove this obstacle through arbitrary precision libraries.

Worked Example - Individual Level Genotype Data

Suppose we are interested in testing whether a specific gene is associated with pancreatic cancer. We have 1000 patients in our study, half with pancreatic cancer and half without. Our dataset consists of 50 SNPs in the gene of interest, and for each patient we have their minor allele count (0,1,2) at each of the 50 SNPs. Additionally we have information on each patient's age and gender:

```
library(GBJ)
set.seed(0)
cancer_status <- c(rep(1,500), rep(0,500))

# All of our SNPs have minor allele frequency of 0.3 in this example
genotype_data <- matrix(data=rbinom(n=1000*50, size=2, prob=0.3), nrow=1000)
age <- round( runif(n=1000, min=30, max=80) )
gender <- rbinom(n=1000, size=1, prob=0.5)     # Let 1 denote a female and 0 a male
```

Under the null hypothesis of no association between gene and pancreatic cancer, we can assume the true logistic model to be:

\[\text{logit}(\mu_{i}) = \beta_{0} + \beta_{1}*Age_{i} + \beta_{2}*Gender_{i}\]

(actually since we generated the data, we know \(\beta_{1}=\beta_{2}=0\), but this is just for illustration)

The function `calc_score_stats()` can be used to calculate score statistics for each of the 50 SNPs. Under the null, these statistics will have an asymptotic N(0,1) distribution.

```
null_mod <- glm(cancer_status~age+gender, family=binomial(link="logit"))
log_reg_stats <- calc_score_stats(null_model=null_mod, factor_matrix=genotype_data, link_function="logit")
log_reg_stats$test_stats
```

```
##  [1] -0.68383391 -1.52282289 -0.04324001  1.02900792 -2.08636456
##  [6] -0.47750816  1.02134595  0.57157512  0.09543809  1.88826217
## [11] -0.50314642 -0.39237040 -1.00903095  0.70567669 -0.74408222
## [16]  1.17911250  0.63650614 -1.45052034 -0.04092308  1.06692697
## [21] -1.39140523 -0.15765491  0.11381805  0.44208014 -0.62752319
## [26]  0.77393856 -0.06840144 -0.27651351  0.03332305  0.96798877
## [31]  0.65627672  0.18681278 -1.11695309 -1.58261616  1.44547339
## [36]  1.78314038  0.88944355  0.44694854 -1.34907210  1.02299625
## [41] -0.59392262 -0.87602299  0.55980215 -1.39571015 -0.16295461
## [46] -0.86944840 -0.21030648 -1.61408239  0.64425823  1.71642921
```

```
log_reg_stats$cor_mat[1:5,1:5]
```

```
##              [,1]        [,2]        [,3]        [,4]         [,5]
## [1,]           NA  0.02020415 -0.01580513  0.03906158  0.005420753
## [2,]  0.020204152          NA -0.03967446  0.03321533  0.028975255
## [3,] -0.015805130 -0.03967446          NA -0.02172816  0.034011177
## [4,]  0.039061583  0.03321533 -0.02172816          NA -0.043455163
## [5,]  0.005420753  0.02897526  0.03401118 -0.04345516           NA
```

If our outcome was continuous and we wanted to assume a linear regression model, then we could still use `calc_score_stats()` with `link_function='linear'`, or if the outcome was non-negative count data and we assume a Poisson regression model, then use `link_function='log'`.

Now we have both the test statistics and their correlation matrix, we can apply GBJ.

```
cor_Z <- log_reg_stats$cor_mat
score_stats = log_reg_stats$test_stats
GBJ(test_stats=score_stats, cor_mat=cor_Z)
```

```
## $GBJ
## [1] 0.6109885
## 
## $GBJ_pvalue
## [1] 0.6091488
## 
## $err_code
## [1] 0
```

And that’s it! Now you are not convinced that GBJ is the correct test for your application, you can also apply GHC, HC, BJ, or minP, as demonstrated below:

```
GHC(test_stats=score_stats, cor_mat=cor_Z)
```

```
## $GHC
## [1] 1.505614
## 
## $GHC_pvalue
## [1] 0.6328255
##
## $err_code
## [1] 0
```

```
HC(test_stats=score_stats, cor_mat=cor_Z)
```

```
## $HC
## [1] 1.514436
##
## $HC_pvalue
## [1] 0.633645
```

```
BJ(test_stats=score_stats, cor_mat=cor_Z)
```

```
## $BJ
## [1] 0.6208709
##
## $BJ_pvalue
## [1] 0.607419
```

```
minP(test_stats=score_stats, cor_mat=cor_Z)
```

```
## $minP
## [1] 0.03694561
##
## $minP_pvalue
## [1] 0.8439997
```

Worked Example - Summary Statistics

Suppose now that we only have GWAS summary statistics, not individual-level data, but we still want to perform a set-level test using GBJ. We then need to estimate the correlations between these summary statistics using genotypes from a reference panel. We have provided a function, estimate_ss_cor() to perform the estimation. estimate_ss_cor() requires as input (1) a matrix of \(m\) PCs calculated from a reference panel (of the same ethnicity) and (2) a matrix of the genotypes at the summary statistic SNPs from the same reference panel. Here \(m\) is approximately the number of PCs used in the original analysis that produced the summary statistics.

```
# Load the genotype data at FGFR2 SNPs for 91 Great Britain (GBR) subjects from the 1000 Genomes Project (publically available).
data(FGFR2)

# Load PCs for these same 91 subjects (calculated, for example, with EIGENSTRAT)
data(gbr_pcs)

# Suppose we were given the following 64 test statistics for the FGFR2 SNPs (must be the same SNPs that
# are in our genotype matrix!)
FGFR2_stats <- rnorm(n=64)

# Estimate correlation matrix for summary statistics
FGFR2_cor_mat <- estimate_ss_cor(ref_pcs=gbr_pcs, ref_genotypes=FGFR2, link_function='logit')

# Run GBJ
GBJ(test_stats=FGFR2_stats, cor_mat=FGFR2_cor_mat)
```

```
## $GBJ
## [1] 0.3366795
##
## $GBJ_pvalue
## [1] 0.6390038
##
## $err_code
## [1] 0
```

Some advice for special cases

- Note that in the first example (individual-level data) we know each of our factors (SNPs) are independent, so another option would be to just input a correlation matrix where all the off-diagonal elements are zero. This should not drastically change the results since our estimated correlations are so close to 0 anyway.
- Limits on the mathematical precision of R and C++ currently limit us somewhat for very large sets and very small p-values. There are plans to remove these limitations in future iterations of the software by using arbitrary precision libraries, however this has not yet been tested. At the moment, we limit sets to 2000 factors at most. Also p-values less than $1*10^{(-14)}$ are generally rounded to this value.

Questions or novel applications? Please let me know! Contact information can be found in the package description.