A Genetic Analysis Package with R

Jing Hua Zhao

MRC Epidemiology Unit, University of Cambridge, Cambridge, UK
http://www.mrc-epid.cam.ac.uk/people/jing-hua-zhao

Contents

1 Introduction 1
2 Implementation 2
3 Independent programs 9
4 Demos 10
5 Examples 10
  5.1 Pedigree drawing ................................................................. 10
  5.2 Kinship calculation .............................................................. 10
  5.3 Study design ................................................................. 12
  5.4 Graphics examples .......................................................... 17
6 Polygenic modeling 24
7 Known bugs 24
8 Summary 24
9 Bibliographic note 25

1 Introduction

This package was initiated to integrate some C/Fortran/SAS programs I have written or used over the years. As such, it would rather be a long-term project, but an immediate benefit would be something complementary to other packages currently available from CRAN, e.g. genetics, hwde, etc. I hope eventually this will be part of a bigger effort to fulfill most of the requirements foreseen by many, e.g. Guo and Lange (2000), within the portable environment of R for data management, analysis, graphics and object-oriented programming. My view has been outlined more formally in Zhao and Tan (2006a) and Zhao and Tan (2006b) in relation to other package systems. Also reported are Zhao (2005) and Zhao (2006) on package kinship.
The number of functions are quite limited and experimental, but I already feel the enormous advantage by shifting to R and would like sooner rather than later to share my work with others. I will not claim this work as exclusively done by me, but would like to invite others to join me and enlarge the collections and improve them.

2 Implementation

The following list shows the data and functions currently available.

* ANALYSIS *

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE3</td>
<td>AE model using nuclear family trios</td>
</tr>
<tr>
<td>bt</td>
<td>Bradley-Terry model for contingency table</td>
</tr>
<tr>
<td>ccsizer</td>
<td>Power and sample size for case-cohort design</td>
</tr>
<tr>
<td>fbsizer</td>
<td>Sample size for family-based linkage and association design</td>
</tr>
<tr>
<td>gc.em</td>
<td>Gene counting for haplotype analysis</td>
</tr>
<tr>
<td>gcontrol</td>
<td>Genomic control</td>
</tr>
<tr>
<td>gcontrol2</td>
<td>Genomic control based on p values</td>
</tr>
<tr>
<td>gcp</td>
<td>Permutation tests using GENECOUNTING</td>
</tr>
<tr>
<td>genecounting</td>
<td>Gene counting for haplotype analysis</td>
</tr>
<tr>
<td>gif</td>
<td>Kinship coefficient and genetic index of familiality</td>
</tr>
<tr>
<td>gsmMCMC</td>
<td>Mixed modeling with genetic relationship matrices</td>
</tr>
<tr>
<td>hap</td>
<td>Haplotype reconstruction</td>
</tr>
<tr>
<td>hap.em</td>
<td>Gene counting for haplotype analysis</td>
</tr>
<tr>
<td>hap.score</td>
<td>Score statistics for association of traits with haplotypes</td>
</tr>
<tr>
<td>htr</td>
<td>Haplotype trend regression</td>
</tr>
<tr>
<td>hwe</td>
<td>Hardy-Weinberg equilibrium test for a multiallelic marker</td>
</tr>
<tr>
<td>hwe.cc</td>
<td>A likelihood ratio test of population Hardy-Weinberg equilibrium</td>
</tr>
<tr>
<td>hwe.hardy</td>
<td>Hardy-Weinberg equilibrium test using MCMC</td>
</tr>
<tr>
<td>kin.morgan</td>
<td>Kinship matrix for simple pedigree</td>
</tr>
<tr>
<td>LD22</td>
<td>LD statistics for two diallelic markers</td>
</tr>
<tr>
<td>LDk1</td>
<td>LD statistics for two multiallelic markers</td>
</tr>
<tr>
<td>masizer</td>
<td>Sample size calculation for mediation analysis</td>
</tr>
<tr>
<td>mia</td>
<td>Multiple imputation analysis for hap</td>
</tr>
<tr>
<td>mtdt</td>
<td>Transmission/disequilibrium test of a multiallelic marker</td>
</tr>
<tr>
<td>mtdt2</td>
<td>Transmission/disequilibrium test of a multiallelic marker by Bradley-Terry model</td>
</tr>
<tr>
<td>mvmeta</td>
<td>Multivariate meta-analysis based on generalized least squares</td>
</tr>
<tr>
<td>pbsizer</td>
<td>Power for population-based association design</td>
</tr>
<tr>
<td>pbsizer2</td>
<td>Power for case-control association design</td>
</tr>
<tr>
<td>pfc</td>
<td>Probability of familial clustering of disease</td>
</tr>
<tr>
<td>pfc.sim</td>
<td>Probability of familial clustering of disease</td>
</tr>
<tr>
<td>pgc</td>
<td>Preparing weight for GENECOUNTING</td>
</tr>
<tr>
<td>print.hap.score</td>
<td>Print a hap.score object</td>
</tr>
<tr>
<td>s2k</td>
<td>Statistics for 2 by K table</td>
</tr>
<tr>
<td>tscc</td>
<td>Power calculation for two-stage case-control design</td>
</tr>
</tbody>
</table>
### *GRAPHICS*

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asplot</td>
<td>Regional association plot</td>
</tr>
<tr>
<td>ESplot</td>
<td>Effect-size plot</td>
</tr>
<tr>
<td>mhtplot</td>
<td>Manhattan plot</td>
</tr>
<tr>
<td>mhtplot2</td>
<td>Manhattan plot with annotations</td>
</tr>
<tr>
<td>pedtodot</td>
<td>Converting pedigree(s) to dot file(s)</td>
</tr>
<tr>
<td>plot.hap.score</td>
<td>Plot haplotype frequencies versus haplotype score statistics</td>
</tr>
<tr>
<td>qqfun</td>
<td>Quantile-comparison plots</td>
</tr>
<tr>
<td>qqunif</td>
<td>Q-Q plot for uniformly distributed random variable</td>
</tr>
</tbody>
</table>

### *DATASETS*

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD</td>
<td>A study of Parkinson’s disease and APOE, LRRK2, SNCA makers</td>
</tr>
<tr>
<td>aldh2</td>
<td>ALDH2 markers and alcoholism</td>
</tr>
<tr>
<td>apoeapoc</td>
<td>APOE/APOC1 markers and schizophrenia</td>
</tr>
<tr>
<td>cf</td>
<td>Cystic Fibrosis data</td>
</tr>
<tr>
<td>crohn</td>
<td>Crohn’s disease data</td>
</tr>
<tr>
<td>fa</td>
<td>Friedreich ataxia data</td>
</tr>
<tr>
<td>fsnps</td>
<td>A case-control data involving four SNPs with missing genotype</td>
</tr>
<tr>
<td>hla</td>
<td>HLA markers and schizophrenia</td>
</tr>
<tr>
<td>l51</td>
<td>An example pedigree data</td>
</tr>
<tr>
<td>lukas</td>
<td>An example pedigree</td>
</tr>
<tr>
<td>mao</td>
<td>A study of Parkinson’s disease and MAO gene</td>
</tr>
<tr>
<td>mayer</td>
<td>A pedigree data on 282 animals deriving from two generations</td>
</tr>
<tr>
<td>nep499</td>
<td>A study of Alzheimer’s disease with eight SNPs and APOE</td>
</tr>
</tbody>
</table>
* UTILITIES *

SNP Functions for single nucleotide polymorphisms (SNPs)
BFDP Bayesian false-discovery probability
FPRP False-positive report probability
ab Test/Power calculation for mediating effect
b2r Obtain correlation coefficients and their variance-covariances
chow.test Chow’s test for heterogeneity in two regressions
comp.score score statistics for testing genetic linkage of quantitative trait
h2 Heritability estimation according to twin correlations for case-control studies
klem Haplotype frequency estimation based on a genotype table of two multiallelic markers
makeped A function to prepare pedigrees in post-MAKEPED format
metap Meta-analysis of p values
metareg Fixed and random effects model for meta-analysis
muvar Means and variances under 1- and 2-locus (diallelic) QTL model
read.ms.output A utility function to read ms output
twinan90 Classic twin models
whscore Whittemore-Halpern scores for allele-sharing
GRM functions
heritability functions

Assuming proper installation, you will be able to obtain the list by typing `library(help=gap)` or view the list within a web browser via `help.start()`. Assuming that you have already loaded the package via `library(gap)`, you can use `lsf.str("package:gap")` to generate a list of functions and a list of datasets, respectively. If this looks odd to you, you might try `search()` within R to examine what is available in your environment before issuing the `lsf.str` command.

```r
> library(gap)
> search()

[1] 

[4] 

[7] 

[10] 

> lsf.str("package:gap")

a2g : function (a1, a2)
ab : function (type = "power", n = 25000, a = 0.15, sa = 0.01, b = log(1.19), sb = 0.01, alpha = 0.05, fold = 1)
```
h2G : function (V, VCOV, verbose = TRUE)
h2GE : function (V, VCOV, verbose = TRUE)
h2l : function (K = 0.05, P = 0.5, h2, se, verbose = TRUE)
hap : function (id, data, nloci, loci = rep(2, nloci), names = paste("loci",
1:nloci, sep = ":"), control = hap.control())
hap.control : function (mb = 0, pr = 0, po = 0.001, to = 0.001, th = 1, maxit = 100,
n = 0, ss = 0, rs = 0, rp = 0, ro = 0, rv = 0, sd = 0, mm = 0, mi = 0,
mc = 50, ds = 0.1, de = 0, q = 0, hapfile = "hap.out", assignfile = "assign.out")
HapDesign : function (HaploEM)
hap.em : function (id, data, locus.label = NA, converge.eps = 1e-06, maxiter = 500,
miss.val = 0)
HapFreqSE : function (HaploEM)
hap.score : function (y, geno, trait.type = "gaussian", offset = NA, x.adj = NA, skip.haplotype = FALSE,
locus.label = NA, miss.val = 0, n.sim = 0, method = "gc", id = NA,
handle.miss = 0, mloci = NA, sexid = NA)
hmht.control : function (data = NULL, colors = NULL, yoffset = 0.25, cex = 1.5, boxed = FALSE)
htr : function (y, x, n.sim = 0)
hwe : function (data, data.type = "allele", yates.correct = FALSE, miss.val = 0)
hwe.cc : function (model, case, ctrl, k0, initial1, initial2)
hwe.hardy : function (a, alleles = 3, seed = 3000, sample = c(1000, 1000, 5000))
invlogit : function (x = 0)
k : function (r, N, adjust = TRUE)
KCC : function (model, GRR, p1, K)
klem : function (obs, k = 2, l = 2)
LD22 : function (h, n)
LDk1 : function (n1 = 2, n2 = 2, h, n, optrho = 2, verbose = FALSE)
logit : function (p = 0.5)
m2plem : function (a1, a2)
makoped : function (pifile = "pedfile.pre", pofile = "pedfile.ped", auto.select = 1,
with.loop = 0, loop.file = NA, auto.proband = 1, proband.file = NA)
masize : function (model, opts, alpha = 0.025, gamma = 0.2)
MCMCgrm : function (model, prior, data, GRM, eps = 0, n.thin = 10, n.burnin = 3000,
n.iter = 13000, ...)
metap : function (data, N, verbose = "Y", prefixp = "p", prefixn = "n")
metareg : function (data, N, verbose = "Y", prefixb = "b", prefixse = "se")
mht.control : function (type = "p", usepos = FALSE, logscale = TRUE, base = 10, cutoffs = NULL,
colors = NULL, labels = NULL, srt = 45, gap = NULL, cex = 0.4, yline = 3,
xline = 3)
mhtplot : function (data, control = mht.control(), hcontrol = hmht.control(), ...)
mhtplot2 : function (data, control = mht.control(), hcontrol = hmht.control(), ...)
mia : function (hapfile = "hap.out", assfile = "assign.out", miafile = "mia.out",
so = 0, ns = 0, mi = 0, allsnps = 0, sas = 0)
micombine : function (est, std.err, confidence = 0.95)
mtdt : function (x, verbose = TRUE, n.sim = NULL, ...)
mtdt2 : function (x, verbose = TRUE, n.sim = NULL, ...)
uivar : function (n.loci = 1, y1 = c(0, 1, 1), y12 = c(1, 1, 1, 1, 1, 0, 0, 0,
mvmeta : function (b, V)
PARN : function (p, RRlist)
psize : function (kp, gamma = 4.5, p = 0.15, alpha = 5e-08, beta = 0.2)
psize2 : function (N, fc = 0.5, alpha = 0.05, gamma = 4.5, p = 0.15, kp = 0.1, model = "additive")
pedtodot : function (pedfile, makeped = FALSE, sink = TRUE, page = "B5", url = "http://www.mrc-epid.cam.ac.uk",
height = 0.5, width = 0.75, rotate = 0, dir = "none")
pfc : function (famdata, enum = 0)
pfc.sim : function (famdata, n.sim = 1e+06, n.loop = 1)
pgc : function (data, handle.miss = 1, is.genotype = 0, with.id = 0)
pem2m : function (a)
plot.hap.score : function (x, ...)
print.hap.score : function (x, ...)
qqfun : function (x, distribution = "norm", ylab = deparse(substitute(x)), xlab = paste(d "%quantiles"), main = NULL, envelope = 0.95, labels = FALSE, col = palette()[4], lcol = palette()[2], xlim = NULL, ylim = NULL, lwd = 1, pch = 1, bg = palette()[4], cex = 0.4, line = c("quartiles", "robust", "none"), ...)
qqunif : function (u, type = "unif", logscale = TRUE, base = 10, col = palette()[4], lcol = palette()[2], ci = FALSE, alpha = 0.05, ...)
ReadGRM : function (prefix = 51)
ReadGRMBin : function (prefix, AllN = FALSE, size = 4)
ReadGRMPCA : function (prefix)
ReadGRMPLINK : function (prefix, diag = 1)
read.ms.output : function (msout, is.file = TRUE, xpose = TRUE, verbose = TRUE, outfile = NULL, outfileonly = FALSE)
revhap : function (loci, hapid)
revhap.i : function (loci, hapid)
s2k : function (y1, y2)
se.exp : function (p, se.p)
se.invlogit : function (logit.p, se.logit)
snp.ES : function (beta, SE, N)
snp.HWE : function (g)
snp.PAR : function (RR, MAF, unit = 2)
solve_skol : function (rootfun, target, lo, hi, e)
toETDT : function (a)
tsc : function (model, GRR, p1, n1, n2, M, alpha.genome, pi.samples, pi.markers, K)
ungcode : function (g)
VR : function (v1, v1, v2, v2, v2, c12)
whscore : function (allele, type)
WriteGRM : function (prefix = 51, id, N, GRM)
WriteGRMBin : function (prefix, grm, N, id, size = 4)
WriteGRMSAS : function (grmlist, outfile = "gwas")
x2 : function (p1, p2, n1, n2)
z : function (p1, p2, n1, n2, r)
> data(package="gap")$results

<table>
<thead>
<tr>
<th>Package</th>
<th>LibPath</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;CDKNgenes (CDKN)&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;CDKNlocus (CDKN)&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;CDKNmap (CDKN)&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;PD&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;aldh2&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;apoepapoc&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;cf&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;crohn&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;fa&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;fsnps&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;hla&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;hr1420&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;151&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;lukas&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;mao&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;mey&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;mflong&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;mhtdata&quot;</td>
</tr>
<tr>
<td>gap</td>
<td>&quot;/tmp/Rtmp0vHrux/Rinst1f9d77041cac&quot;</td>
<td>&quot;nep499&quot;</td>
</tr>
</tbody>
</table>

Title

[1,] "Example data for association plot"
[2,] "Example data for association plot"
[3,] "Example data for association plot"
[4,] "A study of Parkinson's disease and APOE, LRRK2, SNCA makers"
[5,] "ALDH2 markers and Alcoholism"
[6,] "APOE/APOC1 markers and Alzheimer's"
[7,] "Cystic fibrosis data"
[8,] "Crohn's disease data"
[9,] "Friedreich Ataxia data"
[10,] "A case-control data involving four SNPs with missing genotype"
[11,] "The HLA data"
[12,] "An example data for Manhattan plot with annotation"
[13,] "An example pedigree data"
[14,] "An example pedigree"
[15,] "A study of Parkinson's disease and MAO gene"
[16,] "A pedigree data on 282 animals deriving from two generations"
[17,] "Example data for ACEnucfam"
[18,] "An example data for Manhattan plot"
[19,] "A study of Alzheimer's disease with eight SNPs and APOE"

A PDF version of this file can be viewed with command vignette("gap",package="gap").

You can cut and paste examples at end of each function's documentation.
Both *genecounting* and *hap* are able to handle SNPs and multiallelic markers, with the former be flexible enough to include features such as X-linked data and the later being able to handle large number of SNPs. But they are unable to recode allele labels automatically, so functions *gc.em* and *hap.em* are in *haplo.em* format and used by a modified function *hap.score* in association testing.

It is notable that multilocus data are handled differently from that in *hwde* and elegant definitions of basic genetic data can be found in the *genetics* package.

Incidentally, I found my C mixed-radixed sorting routine as in Zhao and Sham (2003) is much faster than R’s internal function.

With exceptions such as function *pfc* which is very computer-intensive, most functions in the package can easily be adapted for analysis of large datasets involving either SNPs or multiallelic markers. Some are utility functions, e.g. *muvar* and *whscore*, which will be part of the other analysis routines in the future.

The benefit with R compared to standalone programs is that for users, all functions have unified format. For developers, it is able to incorporate their C/C++ programs more easily and avoid repetitive work such as preparing own routines for matrix algebra and linear models. Further advantage can be taken from packages in *Bioconductor*, which are designed and written to deal with large number of genes.

### 3 Independent programs

To facilitate comparisons and individual preferences, I have made the source codes available for 2LD, EHPLUS, GENECOUNTING, HAP which have enjoyed great popularity ahead of the genomewide association studies (GWAS) therefore are likely to be more familiar than their R counterparts in *gap*. However, you need to follow their instructions to compile for a particular computer system.

I have included ms code (which is required by *read.ms.output*) and .xls files to accompany *read.ms.output* and *FPRP* and *BFDP* functions as with a classic twin example for ACE model in *OpenMx*. The package is now available from CRAN and earlier it can be installed with command,

```r
source('http://openmx.psyc.virginia.edu/getOpenMx.R')
```

For these models it is actually simpler to use facilities as in package *mets*, which I now suggest.

A final category is *twinan90*, which is now dropped from the package function list due to difficulty to keep up with the requirements by the R environment but nevertheless you will still be able to compile and use otherwise.
4 Demos

You can also try several simple examples via demo:

```r
library(gap)
demo(gap)
```

5 Examples

I would like to highlight `pedtodot pbsize`, `fbsize` and `ccsize` functions used for pedigree drawing and power/sample calculations in a genome-wide association study as reported in Zhao (2007).

5.1 Pedigree drawing

I have included the original file for the R News as well as put examples in separate vignettes. They can be accessed via `vignette("rnews",package="gap")` and `vignette("pedtodot",package="gap")`, respectively.

5.2 Kinship calculation

Next, I will provide an example for kinship calculation using `kin.morgan`. It is recommended that individuals in a pedigree are ordered so that parents always precede their children. In this regard, package `pedigree` can be used, and package `kinship2` can be used to produce pedigree diagram as with kinship matrix.

Pedigree diagram

```r
> # pedigree diagram
> data(lukas,package="gap")
> library(kinship2)
> ped <- with(lukas,pedigree(id,father,mother,sex))
> pdf("figures/lukas.pdf",height=14,width=15)
> plot(ped)
> dev.off()

null device
1
```

The pedigree diagram is as follows,
Kinship calculation

We then turn to the kinship calculation.

```r
> # unordered individuals
> library(gap)
> gk1 <- kin.morgan(lukas)
> write.table(gk1$kin.matrix,"results/gap_1.txt",quote=FALSE)
> library(kinship2)
> kk1 <- kinship(lukas[,1],lukas[,2],lukas[,3])
> write.table(kk1,"results/kinship_1.txt",quote=FALSE)
> d <- gk1$kin.matrix-kk1
> sum(abs(d))

[1] 2.443634
```

> # order individuals so that parents precede their children
> library(pedigree)
We see that in the second case, the result agrees with \texttt{kinship2}.

5.3 Study design

Family-based design

The example involving family-based design is as follows,

```r
> library(gap)
> models <- matrix(c(
+   4.0, 0.01,
+   4.0, 0.10,
+   4.0, 0.50,
+   4.0, 0.80,
+   2.0, 0.01,
+   2.0, 0.10,
+   2.0, 0.50,
+   2.0, 0.80,
+   1.5, 0.01,
+   1.5, 0.10,
+   1.5, 0.50,
+   1.5, 0.80), ncol=2, byrow=TRUE)
> outfile <- "fbsize.txt"
> cat("gamma","p","Y","N_asp","P_A","H1","N_tdt","H2","N_asp/tdt","L_o","L_s
",
+   file=outfile,sep="\t")
> for(i in 1:12) {
+   g <- models[i,1]
+   p <- models[i,2]
+   z <- fbsize(g,p)
+   cat(z$gamma,z$p,z$y,z$n1,z$pA,z$h1,z$n2,z$h2,z$n3,z$lambdao,z$lambda,
+       file=outfile,append=TRUE,sep="\t")
+   cat("\n",file=outfile,append=TRUE)
+ }
> table1 <- read.table(outfile,header=TRUE,sep="\t")
> nc <- c(4,7,9)
> table1[,nc] <- ceiling(table1[,nc])
```
> dc <- c(3,5,6,8,10,11)
> table1[,dc] <- round(table1[,dc],2)
> unlink(outfile)
> # APOE-4, Scott WK, Pericak-Vance, MA & Haines JL
> # Genetic analysis of complex diseases 1327
> g <- 4.5
> p <- 0.15
> cat("\nAlzheimer's:\n\n")

Alzheimer's:

> fbsize(g,p)

$gamma
[1] 4.5

$p
[1] 0.15

$y
[1] 0.6256916

$n1
[1] 162.6246

$pA
[1] 0.8181818

$h1
[1] 0.4598361

$n2
[1] 108.994

$h2
[1] 0.6207625

$n3
[1] 39.97688

$lambdao
[1] 1.671594

$lambdas
[1] 1.784353

> table1
Population-based design

The example involving population-based design is as follows,

```r
> library(gap)
> kp <- c(0.01,0.05,0.10,0.2)
> models <- matrix(c(4.0, 0.01,
+ 4.0, 0.10,
+ 4.0, 0.50,
+ 4.0, 0.80,
+ 2.0, 0.01,
+ 2.0, 0.10,
+ 2.0, 0.50,
+ 2.0, 0.80,
+ 1.5, 0.01,
+ 1.5, 0.10,
+ 1.5, 0.50,
+ 1.5, 0.80), ncol=2, byrow=TRUE)
> outfile <- "pbsize.txt"
> cat("gamma","p","p1","p5","p10","p20\n",sep="\t",file=outfile)
> for(i in 1:dim(models)[1]) {
+  g <- models[i,1]
+  p <- models[i,2]
+  n <- vector()
+  for(k in kp) n <- c(n,ceiling(pbsize(k,g,p)))
+  cat(models[i,1:2],n,sep="\t",file=outfile,append=TRUE)
+  cat("\n",file=outfile,append=TRUE)
+ }
> table5 <- read.table(outfile,header=TRUE,sep="\t")
> table5
```

<table>
<thead>
<tr>
<th>gamma</th>
<th>p</th>
<th>Y</th>
<th>N_asp</th>
<th>P_A</th>
<th>H1</th>
<th>N_tdt</th>
<th>H2</th>
<th>N_asp.tdt</th>
<th>L_o</th>
<th>L_s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.0</td>
<td>0.01</td>
<td>0.52</td>
<td>6402</td>
<td>0.80</td>
<td>0.05</td>
<td>1201</td>
<td>0.11</td>
<td>257</td>
<td>1.08</td>
</tr>
<tr>
<td>2</td>
<td>4.0</td>
<td>0.10</td>
<td>0.60</td>
<td>277</td>
<td>0.80</td>
<td>0.35</td>
<td>165</td>
<td>0.54</td>
<td>53</td>
<td>1.48</td>
</tr>
<tr>
<td>3</td>
<td>4.0</td>
<td>0.50</td>
<td>0.58</td>
<td>446</td>
<td>0.80</td>
<td>0.50</td>
<td>113</td>
<td>0.42</td>
<td>67</td>
<td>1.36</td>
</tr>
<tr>
<td>4</td>
<td>4.0</td>
<td>0.80</td>
<td>0.53</td>
<td>3024</td>
<td>0.80</td>
<td>0.24</td>
<td>244</td>
<td>0.16</td>
<td>177</td>
<td>1.12</td>
</tr>
<tr>
<td>5</td>
<td>2.0</td>
<td>0.01</td>
<td>0.50</td>
<td>445964</td>
<td>0.67</td>
<td>0.03</td>
<td>6371</td>
<td>0.04</td>
<td>2155</td>
<td>1.01</td>
</tr>
<tr>
<td>6</td>
<td>2.0</td>
<td>0.10</td>
<td>0.52</td>
<td>8087</td>
<td>0.67</td>
<td>0.25</td>
<td>761</td>
<td>0.32</td>
<td>290</td>
<td>1.07</td>
</tr>
<tr>
<td>7</td>
<td>2.0</td>
<td>0.50</td>
<td>0.53</td>
<td>3753</td>
<td>0.67</td>
<td>0.50</td>
<td>373</td>
<td>0.47</td>
<td>197</td>
<td>1.11</td>
</tr>
<tr>
<td>8</td>
<td>2.0</td>
<td>0.80</td>
<td>0.51</td>
<td>17909</td>
<td>0.67</td>
<td>0.27</td>
<td>701</td>
<td>0.22</td>
<td>431</td>
<td>1.05</td>
</tr>
<tr>
<td>9</td>
<td>1.5</td>
<td>0.01</td>
<td>0.50</td>
<td>6944779</td>
<td>0.60</td>
<td>0.02</td>
<td>21138</td>
<td>0.03</td>
<td>8508</td>
<td>1.00</td>
</tr>
<tr>
<td>10</td>
<td>1.5</td>
<td>0.10</td>
<td>0.51</td>
<td>101926</td>
<td>0.60</td>
<td>0.21</td>
<td>2427</td>
<td>0.25</td>
<td>1030</td>
<td>1.02</td>
</tr>
<tr>
<td>11</td>
<td>1.5</td>
<td>0.50</td>
<td>0.51</td>
<td>27048</td>
<td>0.60</td>
<td>0.50</td>
<td>1039</td>
<td>0.49</td>
<td>530</td>
<td>1.04</td>
</tr>
<tr>
<td>12</td>
<td>1.5</td>
<td>0.80</td>
<td>0.51</td>
<td>101926</td>
<td>0.60</td>
<td>0.29</td>
<td>1820</td>
<td>0.25</td>
<td>1030</td>
<td>1.02</td>
</tr>
</tbody>
</table>

Case-cohort design

For case-cohort design, we obtain results for ARIC and EPIC studies.

```r
for (i in 1:3)
  cat("n","pD","p1","hr","q","power","ssize\n",file=outfile,sep="\t")
```

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>pD</td>
<td>p1</td>
<td>hr</td>
<td>q</td>
<td>power</td>
<td>ssize</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>15792</td>
<td>0.03</td>
<td>0.25</td>
<td>1.35</td>
<td>0.09264184</td>
<td>0.8</td>
<td>1463</td>
</tr>
<tr>
<td>15792</td>
<td>0.03</td>
<td>0.25</td>
<td>1.40</td>
<td>0.04571935</td>
<td>0.8</td>
<td>722</td>
</tr>
<tr>
<td>15792</td>
<td>0.03</td>
<td>0.25</td>
<td>1.45</td>
<td>0.02963526</td>
<td>0.8</td>
<td>468</td>
</tr>
</tbody>
</table>

```r
> read.table(outfile,header=TRUE,sep="\t")
```

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>pD</td>
<td>p1</td>
<td>hr</td>
<td>q</td>
<td>power</td>
<td>ssize</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>15792</td>
<td>0.03</td>
<td>0.25</td>
<td>1.35</td>
<td>0.09264184</td>
<td>0.8</td>
<td>1463</td>
</tr>
<tr>
<td>15792</td>
<td>0.03</td>
<td>0.25</td>
<td>1.40</td>
<td>0.04571935</td>
<td>0.8</td>
<td>722</td>
</tr>
<tr>
<td>15792</td>
<td>0.03</td>
<td>0.25</td>
<td>1.45</td>
<td>0.02963526</td>
<td>0.8</td>
<td>468</td>
</tr>
</tbody>
</table>

> unlink(outfile)
alpha <- 0.00000005
power <- 0.8
s_pD <- c(0.3,0.2,0.1,0.05)
s_p1 <- seq(0.1,0.5,by=0.1)
s_hr <- seq(1.1,1.4,by=0.1)
cat("n","pD","p1","hr","alpha","ssize\n",file=outfile,sep="\t")
# direct calculation
for(pD in s_pD)
  {
    for(p1 in s_p1)
      {
        for(hr in s_hr)
          {
            ssize <- ccsize(n,q,pD,p1,alpha,log(hr),power)
            if (ssize>0) cat("n","pD","p1","hr","alpha","ssize\n",file=outfile,append=TRUE)
          }
      }
  }
read.table(outfile,header=TRUE,sep="\t")

 n  pD  p1  hr  alpha  ssize
1  25000 0.3 0.1 1.3 5e-08 14391
2  25000 0.3 0.1 1.4 5e-08  5732
3  25000 0.3 0.2 1.2 5e-08  21529
4  25000 0.3 0.2 1.3 5e-08  5099
5  25000 0.3 0.2 1.4 5e-08  2613
6  25000 0.3 0.3 1.2 5e-08 11095
7  25000 0.3 0.3 1.3 5e-08  3490
8  25000 0.3 0.3 1.4 5e-08  1882
9  25000 0.3 0.4 1.2 5e-08  8596
10 25000 0.3 0.4 1.3 5e-08  2934
11 25000 0.3 0.4 1.4 5e-08  1611
12 25000 0.3 0.5 1.2 5e-08  7995
13 25000 0.3 0.5 1.3 5e-08  2786
14 25000 0.3 0.5 1.4 5e-08  1538
15 25000 0.2 0.1 1.4 5e-08  9277
16 25000 0.2 0.2 1.3 5e-08  7725
17 25000 0.2 0.2 1.4 5e-08  3164
18 25000 0.2 0.3 1.3 5e-08  4548
19 25000 0.2 0.3 1.4 5e-08  1805
20 25000 0.2 0.4 1.2 5e-08 20131
21 25000 0.2 0.4 1.3 5e-08  3648
22 25000 0.2 0.4 1.4 5e-08  1805
23 25000 0.2 0.5 1.2 5e-08 17120
24 25000 0.2 0.5 1.3 5e-08  3422
25 25000 0.2 0.5 1.4 5e-08  1713
16

I now include some figures from the documentation that may be of interest.

### Genome-wide association

The following code is used to obtain a Q-Q plot via \texttt{qqunif} function,

```r
> library(gap)
> pdf("figures/qqunif.pdf",height=10,width=10)
> u_obs <- runif(1000)
> r <- qqunif(u_obs,pch=21,bg="blue",bty="n")
> u_exp <- r$y
> hits <- u_exp >= 2.30103
> points(r$x[hits],u_exp[hits],pch=21,bg="green")
> dev.off()
```

null device

1
Based on a chicken genome scan data, the code below generates a Manhattan plot, demonstrating the use of gaps to separate chromosomes.

```r
> library(gap)
> load("4w.rda")
> ord <- with(d, order(chr, pos))
> d <- d[ord,]
> pdf("figures/4w.pdf", height=9, width=10)
> oldpar <- par()
> par(cex=0.6)
> colors <- c(rep("blue", "red"), 15), "red")
> mhtplot(d, control=mht.control(colors=colors, gap=1000), pch=19, srt=0)

Plotting points 1 - 7244
Plotting points 7245 - 12710
Plotting points 12711 - 16875
Plotting points 16876 - 20271
Plotting points 20272 - 22463
Plotting points 22464 - 24192
Plotting points 24193 - 26021
Plotting points 26022 - 27371
Plotting points 27372 - 28558
Plotting points 28559 - 29860
Plotting points 29861 - 31101
Plotting points 31102 - 32504
Plotting points 32505 - 33690
```
> axis(2, cex.axis=2)
> suggestiveline <- -log10(3.60036E-05)
> genomewideline <- -log10(1.8E-06)
> abline(h=suggestiveline, col="blue")
> abline(h=genomewideline, col="green")
> abline(h=0)
> dev.off()

null device
  1
The code below obtains a Manhattan plot with gene annotation,

```r
library(gap)
png("figures/mhtplot.png", height=10, width=16, units="cm", res=300)
data <- with(mhtdata, cbind(chr, pos, p))
glist <- c("IRS1", "SPRY2", "FTO", "GRIK3", "SNED1", "HTR1A", "MARCH3", "WISP3", "PPP1R3B", +   "RP1L1", "FDFT1", "SLC39A14", "GFRA1", "MC4R")
hdata <- subset(mhtdata, gene%in%glist)[c("chr", "pos", "p", "gene")]
color <- rep(c("lightgray", "gray"),11)
glen <- length(glist)
hcolor <- rep("red", glen)
par(las=2, xpd=TRUE, cex.axis=1.8, cex=0.4)
ops <- mht.control(colors=color, yline=1.5, xline=3)
hops <- hhmht.control(data=hdata, colors=hcolor)
mhtplot(data, ops, hops, pch=19)
```

Plotting points  1 - 12123
Plotting points  12124 - 26444
Plotting points 26445 - 37326
Plotting points 37327 - 47549
Plotting points 47550 - 58877
Plotting points 58878 - 71908
Plotting points 71909 - 79690
Plotting points 79691 - 90464
Plotting points 90465 - 101267
Plotting points 101268 - 109000
Plotting points 109001 - 116159
Plotting points 116160 - 124094
Plotting points 124095 - 130329
Plotting points 130330 - 134176
Plotting points 134177 - 139300
Plotting points 139301 - 143751
Plotting points 143752 - 148345
Plotting points 148346 - 153379
Plotting points 153380 - 155466
Plotting points 155467 - 157052
Plotting points 157053 - 159312
... highlighting 1559 - 1657 GRIK3
... highlighting 26343 - 26349 SNED1
... highlighting 55142 - 55144 MARCH3
... highlighting 66533 - 66539 WISP3
... highlighting 81546 - 81551 RP1L1
... highlighting 82146 - 82168 FDFT1
... highlighting 83425 - 83458 SLC39A14
... highlighting 107866 - 107894 GFRA1
... highlighting 141457 - 141576 FTO
... highlighting 152037 - 152037 MC4R

> axis(2,pos=2,at=1:16)
> title("Manhattan plot with genes highlighted",cex.main=1.8)
> dev.off()

null device
1
All these look familiar, so revised form of the function called \texttt{mhtplot2} was created for additional features such as centering the chromosome ticks, allowing for more sophisticated coloring schemes, using prespecified fonts, etc. Please refer to the function's documentation example.

The code below obtains a regional association plot with the \texttt{asplot} function,

\begin{verbatim}
> library(gap)
> pdf("figures/asplot.pdf",height=14,width=14)
> asplot(CDKNlocus, CDKNmap, CDKNgenes, best.pval=5.4e-8, sf=c(3,6))

CDKN2A
CDKN2B

> title("CDKN2A/CDKN2B Region")
> dev.off()

null device
1
\end{verbatim}
The function predates the currently popular **locuszoom** software but leaves the option open for generating such plots on the fly and locally.

**Effect size plot**

The code below obtains an effect size plot via the ESplot function.

```r
> library(gap)
> pdf("figures/ESplot.pdf",height=10,width=10)
> options(stringsAsFactors=FALSE)
> testdata <- data.frame(models=c("Basic model","Adjusted","Moderately adjusted", "Heavily adjusted","Other"),
+ OR = c(4.5,3.5,2.5,1.5,1),
+ SElogOR = c(0.2,0.1,0.5,0.5,0.2))
> ESplot(testdata,v=1)
> title("This is a fictitious plot")
> dev.off()
```
null device

1

This is a fictitious plot

Note that all these can serve as templates to customize features of your own.

6 Polygenic modeling

In line with the recent surge of interest in the polygenic models, a separate vignette is available through `vignette("h2", package="gap")` demonstrating aspect of the models on heritability.

7 Known bugs

Unaware of any bug. However, better memory management is expected.

8 Summary

I believe by now the package should have given you a flavour of initiatives I have made so far in relation to how the project was envisaged. More importantly, it is clear that availability of the package will serve as a platform on which future work can be accumulated and collaboration can be built.
9 Bibliographic note

The main references are Chow (1960); Guo and Thompson (1992); Williams et al. (1992); Gholamic and Thomas (1994); Hartung et al. (2008); Risch and Merikangas (1996); Spielman and Ewens (1996); Risch and Merikangas (1997); Miller (1997); Sham (1997); Elston (1975); Sham (1998); Devlin and Roeder (1999); Zhao et al. (1999); Guo and Lange (2000); Hirotsu et al. (2001); Zhao et al. (2002); Zaykin et al. (2002); Zhao (2004); Wacholder et al. (2004); Wang (2005); Skol et al. (2006); Wakefield (2007).

References


