Example R code to perform small-scale analyses using GENETICS

```
library(DGCgenetics)
library(dgc.genetics)
casecon <- read.table("casecondata.txt",header=T)
casecon[1:2,]
attach(casecon)
pedigree
case <- affected-1
case
g1 <- genotype(loc1_1,loc1_2)
g1 <- genotype(loc2_1,loc2_2)
g1 <- genotype(loc3_1,loc3_2)
g1 <- genotype(loc1_1,loc1_2)
g2 <- genotype(loc2_1,loc2_2)
g3 <- genotype(loc3_1,loc3_2)
g4 <- genotype(loc4_1,loc4_2)
g1
```

```
table(g1,case)
chisq.test(g1,case)
allele.table(g1,case)
gcontrasts(g1) <- "genotype"
names(casecon)
help(gcontrasts)
logit(case~g1)
anova(logit(case~g1))
1-pchisq(18.49,2)
gcontrasts(g1) <- "genotype"
gcontrasts(g3) <- "genotype"
logit(case~g1+g3)
anova(logit(case~g1+g3))          # This is in fact already a multiple SNP analysis
gcontrasts(g1) <- "genotype"      # But you can see how easy it is within a
gcontrasts(g3) <- "additive"      # regression framework
logit(case~g1+g3)
anova(logit(case~g1+g3))
detach(casecon)
```

Example R code to perform small-scale analyses using SNPassoc

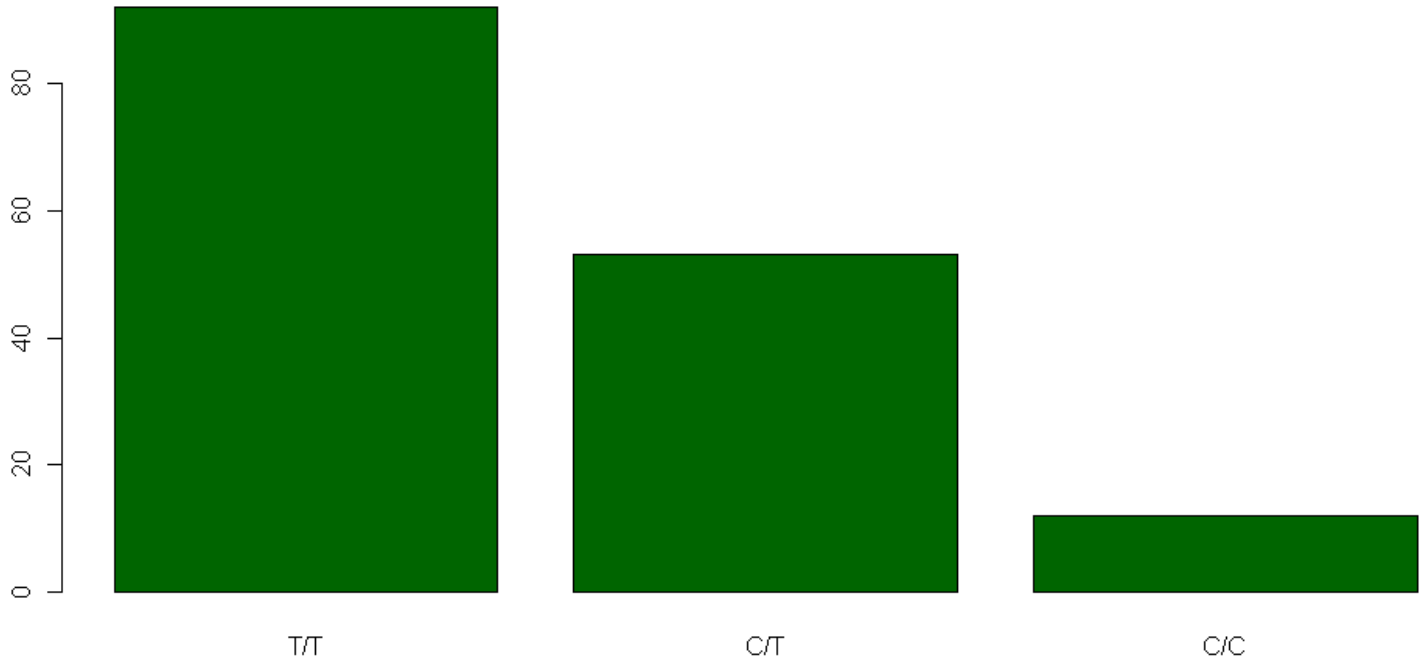
```
#Let's load library SNPassoc
library(SNPassoc)
#get the data example:
#both data.frames SNPs and SNPs.info.pos are loaded typing data(SNPs)
data(SNPs)
#look at the data (only first four SNPs)
SNPs[1:10,1:9]
table(SNPs[,2])
mySNP<-snp(SNPs$snp10001,sep="")
mySNP
summary(mySNP)
```

snp10001

	frequency	percentage
T	237	75.48
C	77	24.52

	frequency	percentage
T/T	92	58.60
C/T	53	33.76
C/C	12	7.64

HWE (pvalue): 0.281639



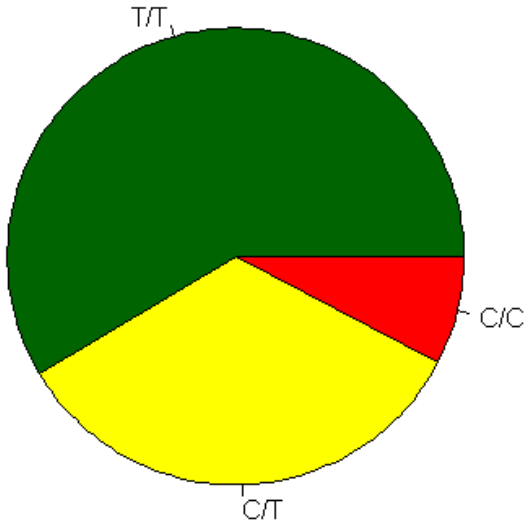
plot(mySNP,label="snp10001",col="darkgreen")

snp10001

	frequency	percentage
T	237	75.48
C	77	24.52

	frequency	percentage
T/T	92	58.60
C/T	53	33.76
C/C	12	7.64

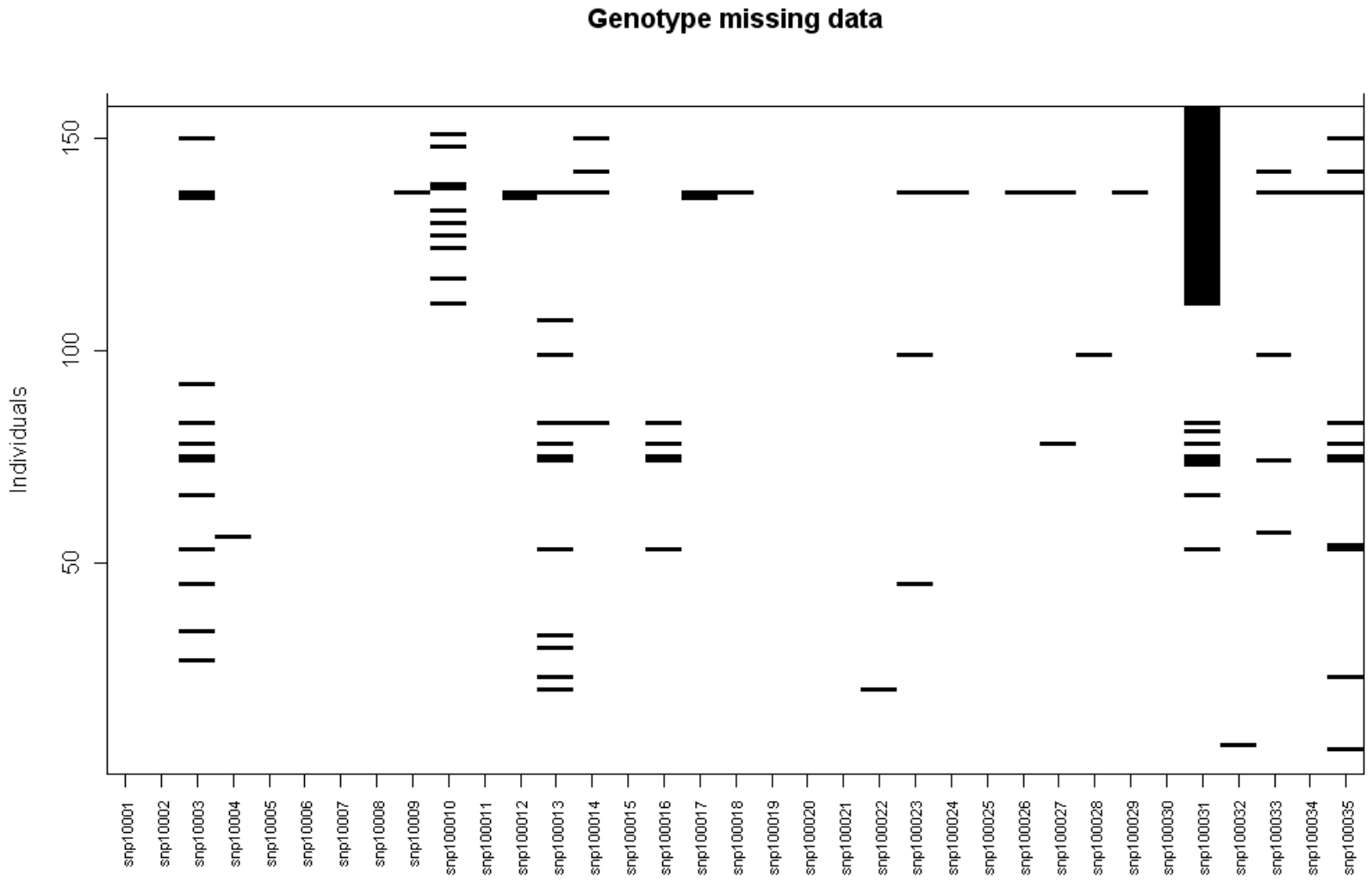
HWE (pvalue): 0.281639




```
plot(mySNP,type=pie,label="snp10001",col=c("darkgreen","yellow","red"))
```

Example R code to perform small-scale analyses using SNPassoc

```
reorder(mySNP,ref="minor")
gg<-
c("het","hom1","hom1","hom1","hom1","hom1","het","het","het","hom1","hom2","hom
1","hom2")
snp(gg,name.genotypes=c("hom1","het","hom2"))
myData<-setupSNP(data=SNPs,colSNPs=6:40,sep="")
myData.o<-setupSNP(SNPs, colSNPs=6:40, sort=TRUE,info=SNPs.info.pos, sep="")
labels(myData)
summary(myData)
plot(myData,which=20)
```



plotMissing(myData)

Example R code to perform small-scale analyses using SNPassoc

```
res<-tableHWE(myData)
res
res<- tableHWE(myData,strata=myData$sex)
res
```

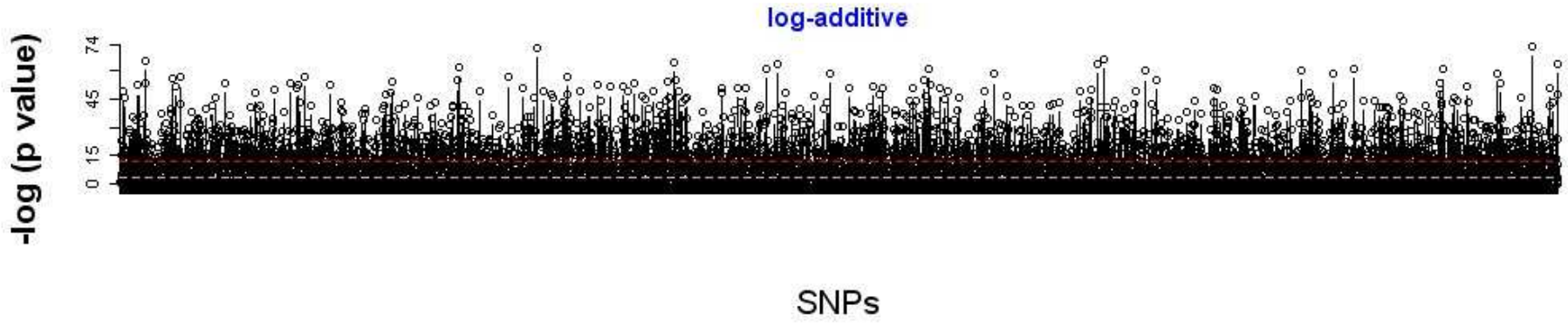
What is the difference between the two previous commands?
Why is the latter analysis important?

Example R code to perform GWA using SNPassoc

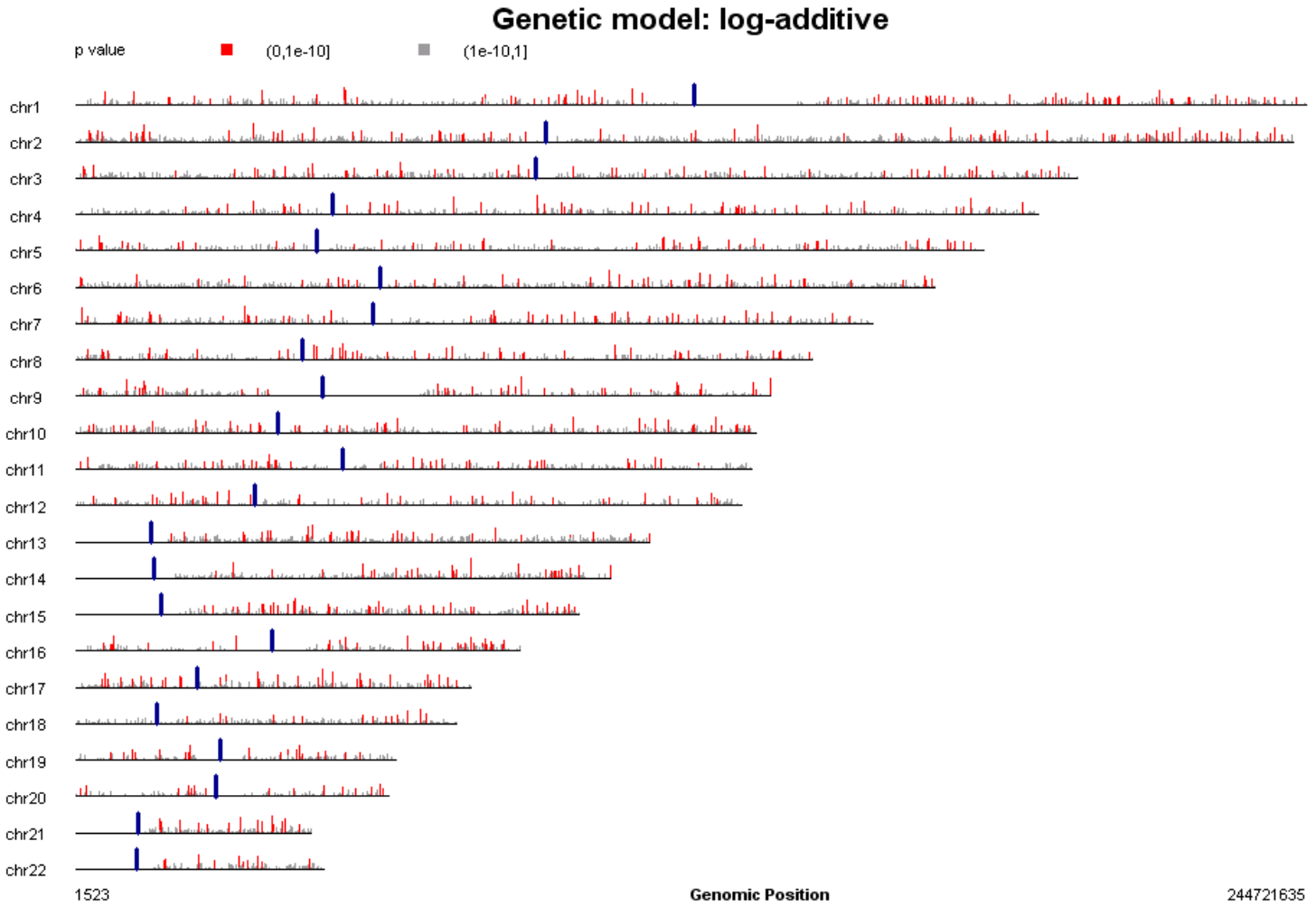
```
data(HapMap)

> HapMap[1:4,1:9]
      id group rs10399749 rs11260616 rs4648633 rs6659552 rs7550396 rs12239794
rs6688969
1  NA06985   CEU       CC       AA       TT       GG       GG       GG       CC
2  NA06993   CEU       CC       AT       CT       CG       GG       GG       CT
3  NA06994   CEU       CC       AA       TT       CG       GG       GG       CT
4  NA07000   CEU       CC       AT       TT       GG       GG     <NA>       CC

myDat.HapMap<-setupSNP(HapMap, colSNPs=3:9307, sort =
TRUE,info=HapMap.SNPs.pos, sep="")

> HapMap.SNPs.pos[1:3,]
       snp chromosome position
1 rs10399749       chr1    45162
2 rs11260616       chr1  1794167
3  rs4648633       chr1  2352864
```

Example R code to perform GWA using SNPassoc

```
resHapMap<-WGassociation(group, data=myDat.HapMap, model="log-add")

plot(resHapMap, whole=FALSE, print.label.SNPs = FALSE)
```




```
> summary(resHapMap)
SNPs (n) Genot error (%) Monomorphic (%) Significant* (n)  (%)
chr1     796         3.8          18.6          163 20.5
chr2     789         4.2          13.9          161 20.4
chr3     648         5.2          13.0          132 20.4
```



plot(resHapMap, whole=TRUE, print.label.SNPs = FALSE)

Example R code to perform GWA using SNPassoc

```
resHapMap.scan<-scanWGassociation(group, data=myDat.HapMap, model="log-add")
resHapMap.perm<-scanWGassociation(group, data=myDat.HapMap,model="log-add",
nperm=1000)
res.perm<- permTest(resHapMap.perm)
```

- Check out the SNPassoc manual (supporting document to R package) to read more about the analytical methods used

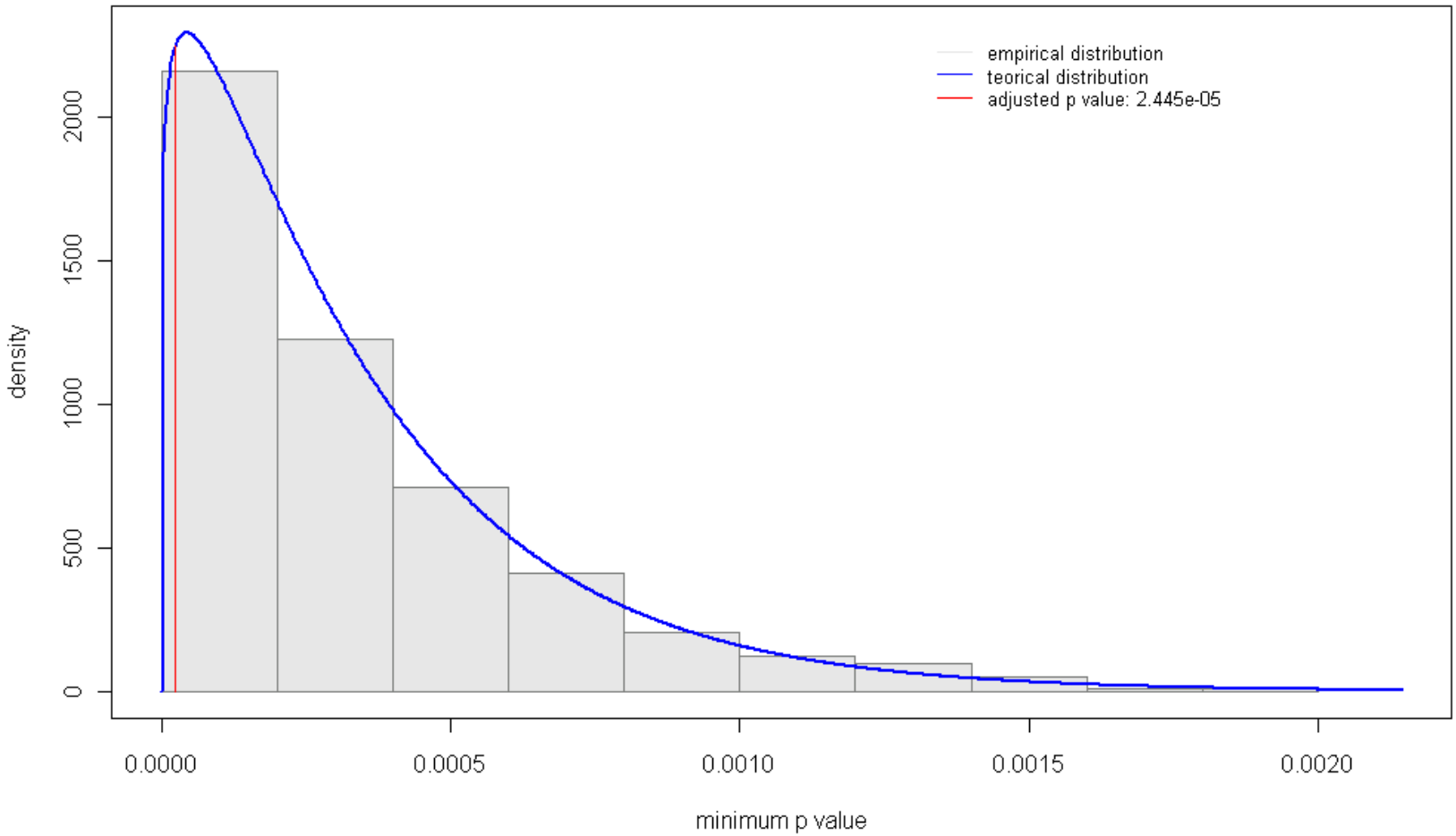
Example R code to perform GWA using SNPassoc

```
> print(resHapMap.scan[1:5,])
          comments    log-additive
rs10399749 Monomorphic -
rs11260616 -          0.34480
rs4648633  -          0.00000
rs6659552  -          0.00000
rs7550396  -          0.31731
> print(resHapMap.perm[1:5,])
          comments    log-additive
rs10399749 Monomorphic -
rs11260616 -          0.34480
rs4648633  -          0.00000
rs6659552  -          0.00000
rs7550396  -          0.31731
```

```
perms <- attr(resHapMap.perm, "pvalPerm")   #what does this object contain?
```

Example R code to perform GWA using SNPassoc

```
> print(res.perm)

Permutation test analysis (95% confidence level)
-----------------------------------------------
Number of SNPs analyzed:  9305
Number of valid SNPs (e.g., non-Monomorphic and passing calling rate):  7320
P value after Bonferroni correction:  6.83e-06

P values based on permutation procedure:
P value from empirical distribution of minimum p values:  2.883e-05
P value assuming a Beta distribution for minimum p values:  2.445e-05
```



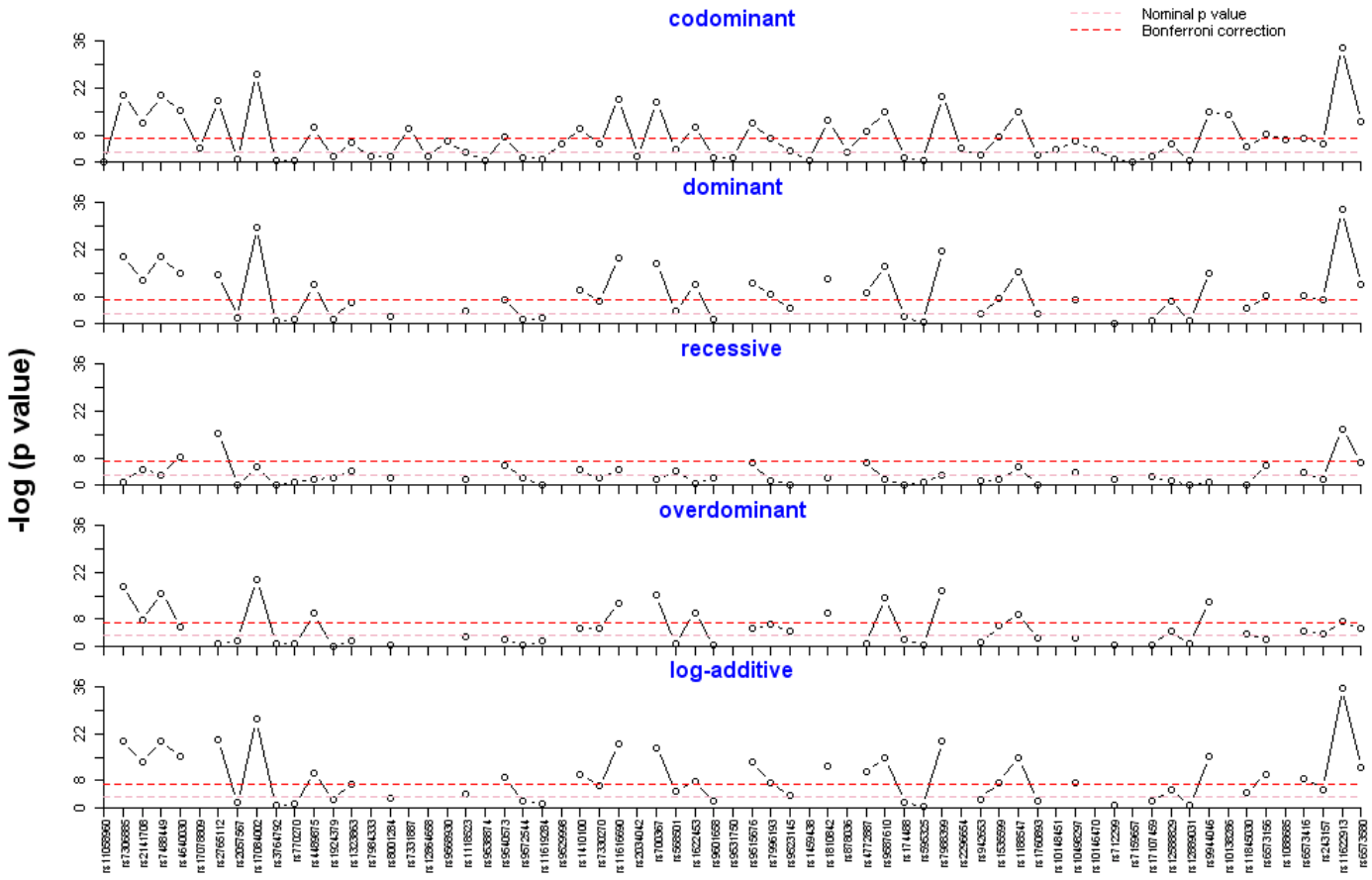
plot(res.perm)

Example R code to perform GWA using SNPassoc

```
res.perm.rtp<- permTest(resHapMap.perm,method="rtp",K=20)
> print(res.perm.rtp)

Permutation test analysis (95% confidence level)
-----------------------------------------------
Number of SNPs analyzed:  9305
Number of valid SNPs (e.g., non-Monomorphic and passing calling rate):
7320
P value after Bonferroni correction:  6.83e-06

Rank truncated product of the K=20 most significant p-values:
Product of K p-values (-log scale):  947.2055
Significance:  <0.001
```

Example R code to perform a variety of medium/large-scale analyses using SNPassoc

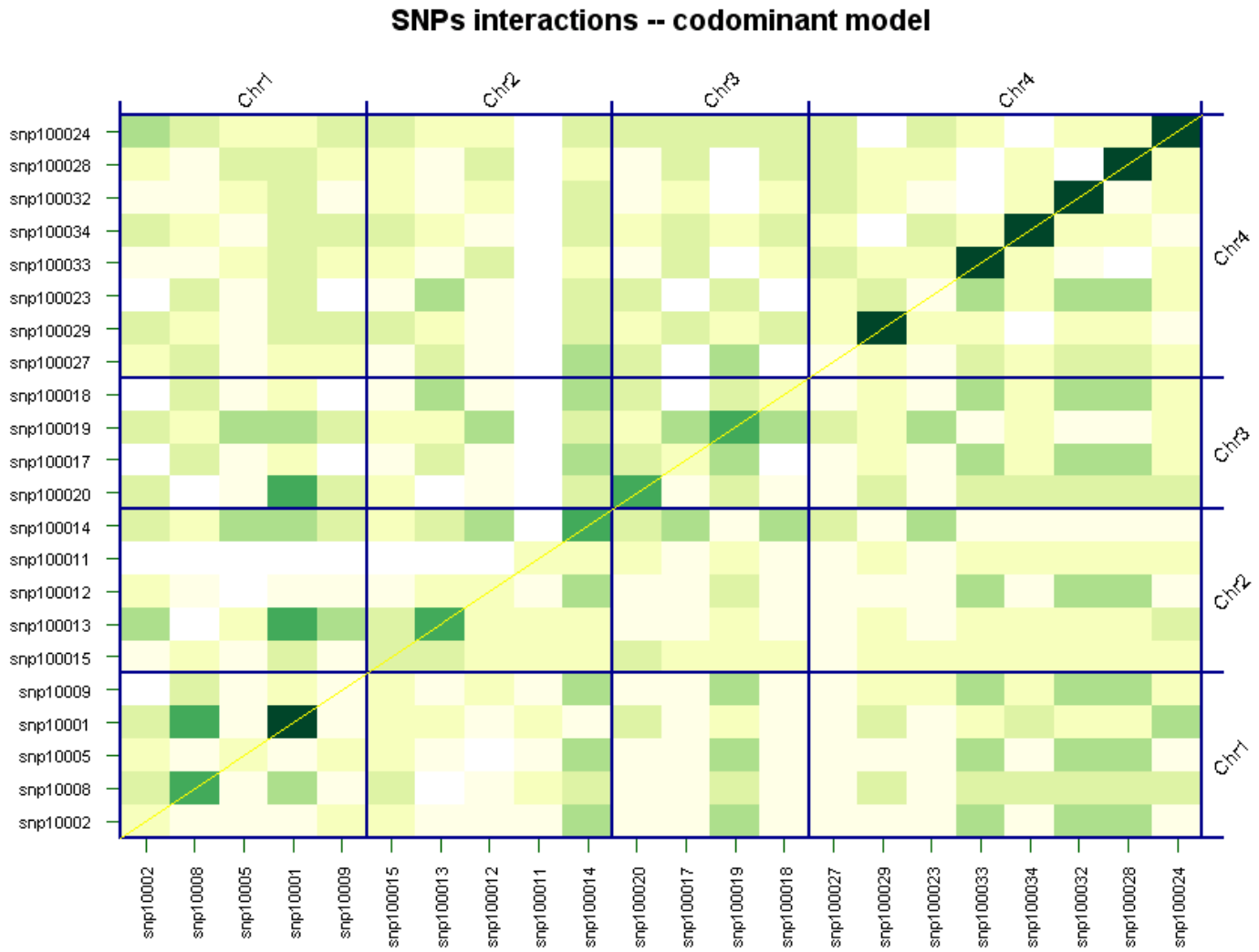
```
getSignificantSNPs(resHapMap,chromosome=5)
association(casco~snp(snp10001,sep=""), data=SNPs)
myData<-setupSNP(data=SNPs,colSNPs=6:40,sep="")
association(casco~snp10001, data=myData)
association(casco~snp10001, data=myData, model=c("cod","log"))
association(casco~sex+snp10001+blood.pre, data=myData)
association(casco~snp10001+blood.pre+strata(sex), data=myData)
association(casco~snp10001+blood.pre, data=myData,subset=sex=="Male")
association(log(protein)~snp100029+blood.pre+strata(sex), data=myData)
ans<-association(log(protein)~snp10001*sex+blood.pre,
data=myData,model="codominant")
print(ans,dig=2)
ans<-association(log(protein)~snp10001*factor(recessive(snp100019))+blood.pre,
data=myData, model="codominant")
print(ans,dig=2)
```




```
sigSNPs<-getSignificantSNPs(resHapMap,chromosome=5,sig=5e-8)$column
myDat2<-setupSNP(HapMap, colSNPs=sigSNPs, sep="")
resHapMap2<-WGassociation(group~1, data=myDat2)
plot(resHapMap2,cex=0.8)
```

Example R code using SNPassoc

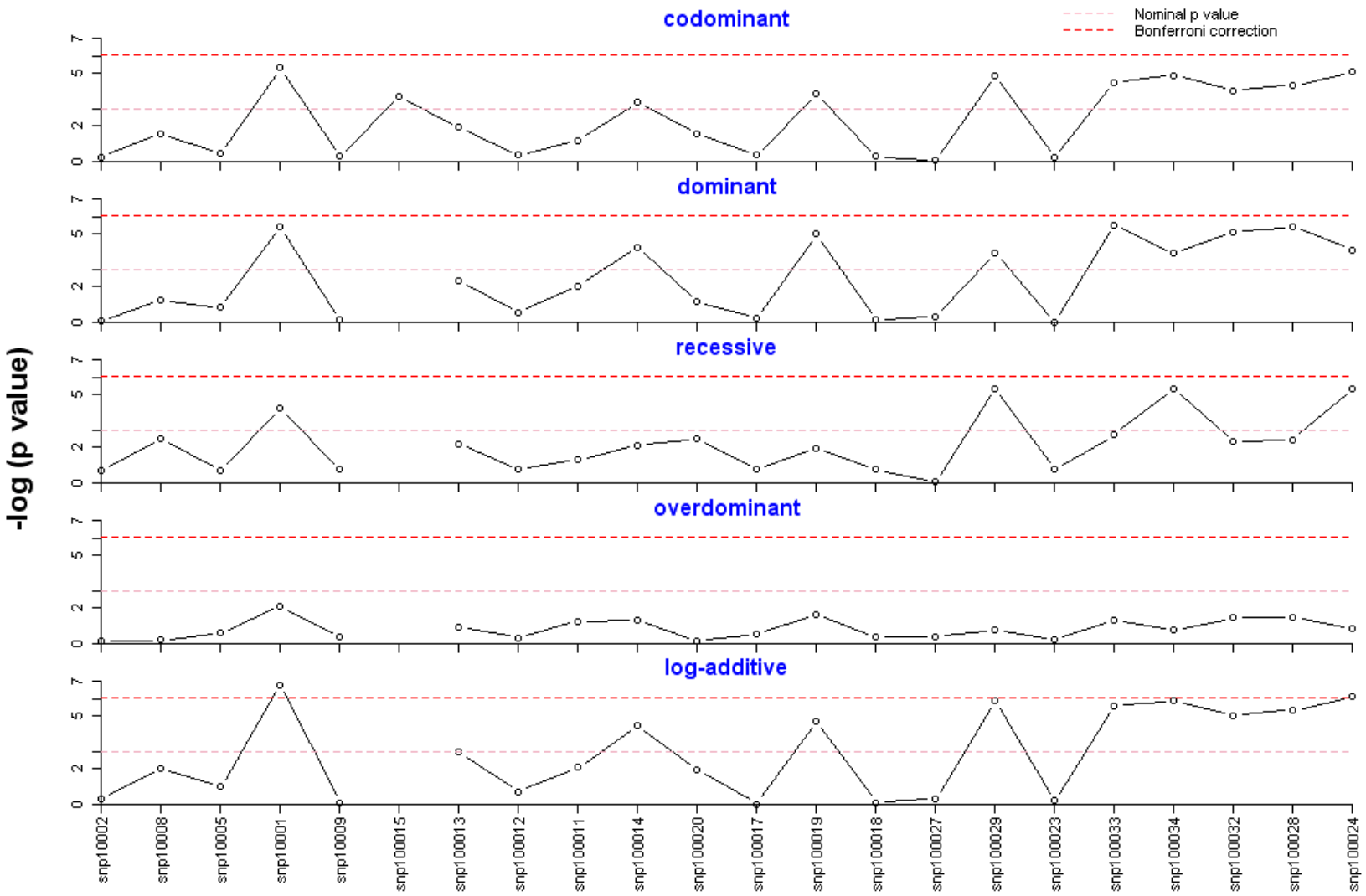
```
datSNP<-setupSNP(SNPs,6:40,sep="")
tag.SNPs<-c("snp100019", "snp10001", "snp100029")
geno<-make.geno(datSNP,tag.SNPs)
mod<-
haplo.glm(log(protein)~geno,data=SNPs,family=gaussian,locus.label=tag.SNPs,allele.lev=at
tributes(geno)$unique.alleles,
control = haplo.glm.control(haplo.freq.min=0.05))
mod
intervals(mod)
ansCod<-interactionPval(log(protein)~sex, data=myData.o,model="codominant")
```


SNPs interactions -- codominant model
Chr1
Chr2
Chr3
Chr4
snp100024
snp100028
snp100032
snp100034
snp100033
snp100023
snp100029
snp100027
snp100018
snp100019
snp100017
snp100020
snp100014
snp100011
snp100012
snp100013
snp100015
snp10009
snp10001
snp10005
snp10008
snp10002
snp10002
snp10008
snp10005
snp10001
snp10009
snp100015
snp100013
snp100012
snp100011
snp100014
snp100020
snp100017
snp100019
snp100018
snp100027
snp100029
snp100023
snp100033
snp100034
snp100032
snp100028
snp100024

plot(ansCod)

Example R code using SNPassoc

```
myData<-setupSNP(SNPs, colSNPs=6:40, sep="")
myData.o<-setupSNP(SNPs, colSNPs=6:40, sort=TRUE,info=SNPs.info.pos, sep="")
ans<-WGassociation(protein~1,data=myData.o)
library(Hmisc)
SNP<-pvalues(ans)
out<-latex(SNP,file="c:/temp/ans1.tex", where="'h",caption="Summary of case-control
study for SNPs data set.",center="centering", longtable=TRUE, na.blank=TRUE,
size="scriptsize", collabel.just=c("c"), lines.page=50,rownamesTexCmd="bfseries")
WGstats(ans,dig=5)
```


codominant
dominant
recessive
overdominant
log-additive
Nominal p value
Bonferroni correction
-log (p value)
0
2
5
7
snp10002
snp10008
snp10005
snp10001
snp10009
snp100015
snp100013
snp100012
snp100011
snp100014
snp100020
snp100017
snp100019
snp100018
snp100027
snp100029
snp100023
snp100033
snp100034
snp100032
snp100028
snp100024

plot(ans)

Example R code using SNPassoc

```
Bonferroni.sig(ans, model="log-add", alpha=0.05,include.all.SNPs=FALSE)
pvalAdd<-additive(resHapMap)
pval<-pval[!is.na(pval)]
library(qvalue)
qobj<-qvalue(pval)
max(qobj$qvalues[qobj$pvalues <= 0.001])
procs<-c("Bonferroni","Holm","Hochberg","SidakSS","SidakSD","BH","BY")
res2<-mt.rawp2adjp(rawp,procs)
mt.reject(cbind(res$rawp,res$adjp),seq(0,0.1,0.001))$r
```