A toy example: defining data

- Define genotype counts in cases and controls

```
> gc <- c(900,100,3,800,190,10)
```

- Calculate allele counts

```
> ac <- c(2*gc[1]+gc[2],gc[2]+2*gc[3],2*gc[4]+gc[5],gc[5]+2*gc[6])
```

- Count individuals having at least one common/variant allele

```
> gc1 <- c(gc[1]+gc[2],gc[3],gc[4]+gc[5],gc[6])
> gc2 <- c(gc[1],gc[2]+gc[3],gc[4],gc[5]+gc[6])
```

A toy example: testing

- Test for genotypic association

```
> pvg <- chisq.test(matrix(gc,ncol=3,byrow=TRUE),corr=FALSE)$p.value
```

- Test for allelic association (additive, valid provided HWE holds)

```
> pva <- chisq.test(matrix(ac,ncol=2,byrow=TRUE),corr=FALSE)$p.value
```

- Test for dominant/recessive model and keeping minimal p-value

```
> pvg1 <- chisq.test(matrix(gc1,ncol=2,byrow=TRUE),corr=FALSE)$p.value
> pvg2 <- chisq.test(matrix(gc2,ncol=2,byrow=TRUE),corr=FALSE)$p.value
> pvb <- min(pvg1,pvg2)
```

- Results

```
> print(c(pvg,pva,pvb))
[1] 6.918239e-09 9.150309e-10 1.224003e-09
```

A toy example: testing

- Exact tests

```
> pvg .f <- fisher.test(matrix(gc,ncol=3,byrow=TRUE))$p.value
> pva .f <- fisher.test(matrix(ac,ncol=2,byrow=TRUE))$p.value
> pvg1.f <- fisher.test(matrix(gc1,ncol=2,byrow=TRUE))$p.value
> pvg2 .f <- fisher.test(matrix(gc2,ncol=2,byrow=TRUE))$p.value
> pvb .f <- min(pvg1 .f,pvg2 .f)
> print(c(pvg .f,pva .f,pvb .f))
[1] 2.412721e-09 8.047005e-10 1.132535e-09
```

- Trend test (additive model, valid regardless of HWE assumption)

```
> pvcat <- prop.trend.test(gc[1:3],gc[1:3]+gc[4:6],score=c(0,0.5,1))$p.value
> print(pvcat)
[1] 9.820062e-10
```

A toy example: testing

- Double sample size

```
> gc<-gc*2
...
> print(c(pvg,pva,pvb))
[1] 4.786203e-17 4.716312e-18 8.379499e-18
> print(c(pvg.f,pva.f,pvb.f))
[1] 1.231881e-17 3.485271e-18 6.810263e-18
> print(pvcat)
[1] 5.422705e-18
```

A toy example: estimation

- Function to calculate OR and CI

```
> ci.or <- function(counts,alpha){
+  f <- qnorm(1-alpha/2)
+  or <- counts[1]*counts[4]/(counts[2]*counts[3])
+  sq <- sqrt(1/counts[1]+1/counts[2]+1/counts[3]+1/counts[4])
+  upper <- exp(log(or)+f*sq)
+  lower <- exp(log(or)-f*sq)
+  res <- c(lower,or,upper)
+  res
+ }
```

- OR and 95% CI (alpha=0.05)

```
> print(ci.or(ac,0.05))
[1] 1.650411 2.102878 2.679390
```

A toy example: estimation

- Decrease significance level: 99% CI (alpha=0.01)

```
> print(ci.or(ac,0.01))
[1] 1.529428 2.102878 2.891339
```

- Double sample size

```
> gc<-gc*2
> ac <- c(2*gc[1]+gc[2],gc[2]+2*gc[3],2*gc[4]+gc[5],gc[5]+2*gc[6])
> print(ci.or(ac,0.05))
[1] 1.771784 2.102878 2.495842
> print(ci.or(ac,0.01))
[1] 1.678927 2.102878 2.633882
```

Installing R-package SNPassoc

- As SNPassoc is not available for recent R versions, we first need to install R version 2.9.2 (or lower, but at least 2.4.0) from http://cran.r-project.org/bin/windows/base/old/2.9.2/
- Install dependencies haplo.stats and mvtnorm

```
> install.packages(c('haplo.stats','mvtnorm'))
```

- Download Windows binary of SNPassoc package from http://www.mirrorservice.org/sites/lib.stat.cmu.edu/R/CRAN/src/contrib/Descriptions/SNPassoc.html and install using

```
> install.packages('SNPassoc_1.4-9.zip',repos=NULL)
```

- At the start of each session load the SNPassoc package using

```
> library(SNPassoc)
```

Data manipulation: loading data

- Load example data frames SNPs and SNPs.info.pos by typing

```
> data(SNPs)
```

- Look at the data (first two individuals, first three SNPs)

```
> SNPs[1:2,1:9]
  id casco    sex blood.pre  protein snp10001 snp10002 snp10003 snp10004
1  1     1 Female      13.7 75640.52       TT       CC       GG       GG
2  2     1 Female      12.7 28688.22       TT       AC       GG       GG
> SNPs.info.pos[1:3,]
       snp  chr     pos
1 snp10001 Chr1 2987398
2 snp10002 Chr1 1913558
3 snp10003 Chr1 1982067
```

Data manipulation: class snp

- Assess numbers of cases (110) and controls (47)

```
> table(SNPs[,2])
  0   1
 47 110
```

- Create object of class snp

```
> mySNP<-snp(SNPs$snp10001,sep="")
> mySNP[1:7]
[1] T/T T/T T/T C/T T/T T/T T/T
Genotypes: T/T C/T C/C
Alleles:  T C
```

Descriptive analysis: class snp

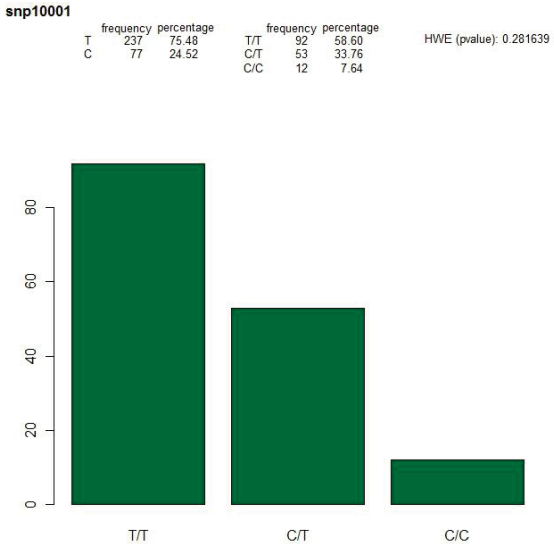
- Summarize object of class snp

```
> summary(mySNP)
Genotypes:
    frequency percentage
T/T        92  58.598726
C/T        53  33.757962
C/C        12   7.643312
Alleles:
  frequency percentage
T       237   75.47771
C        77   24.52229
HWE (p value): 0.2816392
```

Descriptive analysis: class snp

- Summarize object of class snp using a barplot

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



Descriptive analysis: class snp

- Summarize object of class snp using a pie chart

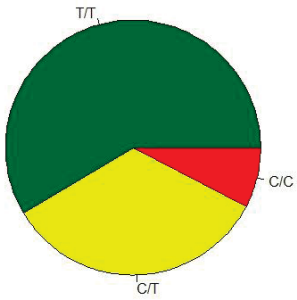
```
> plot(mySNP,type=pie,label="snp10001",col=c("darkgreen","yellow","red"))
```

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



Data manipulation: class snp

- Change the reference category from genotype with common allele to genotype with minor allele

```
> reorder(mySNP,ref="minor")[1:7]
[1] T/T T/T T/T C/T T/T T/T T/T
Genotypes: C/C C/T T/T
Alleles:
```

- Flexibly indicate genotype codes

```
> gg<-c("het","hom1","hom1","hom2","hom1","hom1","het","het")
> snp(gg,name.genotypes=c("hom1","het","hom2"))
[1] A/B A/A A/A B/B A/A A/A A/B A/B
Genotypes: A/A A/B B/B
Alleles:  A B
```

Data manipulation: class setupSNP

- Create an object of class setupSNP

```
> myData<-setupSNP(data=SNPs,colSNPs=6:40,sep="")
> myData[1:2,1:8]
  id casco    sex blood.pre  protein snp10001 snp10002 snp10003
1  1     1 Female      13.7 75640.52      T/T      C/C      G/G
2  2     1 Female      12.7 28688.22      T/T      A/C      G/G
```

- Sort by chromosome and genomic position

```
> myData.o[1:2,1:8]
  id casco    sex blood.pre  protein snp10004 snp10007 snp100010
1  1     1 Female      13.7 75640.52      G/G      C/C       T/T
2  2     1 Female      12.7 28688.22      G/G      C/C       T/T
```

Descriptive analysis: class setupSNP

- Get labels of object of class setupSNP

```
> labels(myData)[1:3]
[1] "snp10001" "snp10002" "snp10003"
```

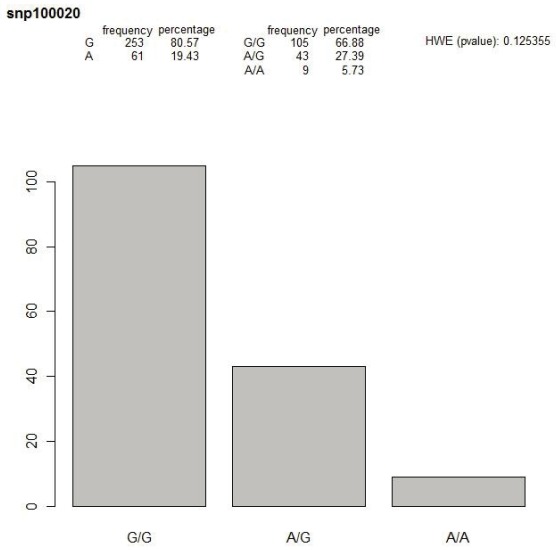
- Summarize object of class setupSNP

```
> summary(myData)
         alleles major.allele.freq HWE      missing (%)
snp10001  T/C      75.5            0.281639  0.0
snp10002  C/A      72.0            0.004945  0.0
snp10003  G       100.0            -         8.3
snp10004  G       100.0            -         0.6
snp10005  G/A      75.8            0.008020  0.0
...
```

Descriptive analysis: class setupSNP

- Summarize and plot a particular SNP

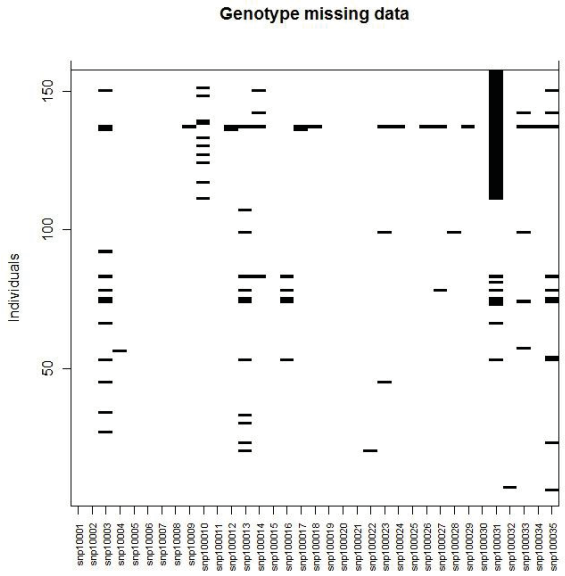
```
> plot(myData,which=20)
```



Descriptive analysis: missing data

- Plot missingness patterns

```
> plotMissing(myData)
```



Descriptive analysis: Hardy-Weinberg equilibrium

- Assess Hardy-Weinberg equilibrium (HWE)

```
> res<-tableHWE(myData)
> res
          HWE (p value) flag
snp10001  0.2816
snp10002  0.0049       <-
snp10003  -
snp10004  -
snp10005  0.0080       <-
…
```

Descriptive analysis: Hardy-Weinberg equilibrium

- Assess HWE stratified by sex

```
> res
          all.groups   Male Female
snp10001      0.2816 0.3941 0.7388
snp10002      0.0049 0.1660 0.0075
snp10003           -      -      -
snp10004           -      -      -
snp10005      0.0080 0.2755 0.0257
...
```

GWA analysis: loading data

- Load HapMap data

```
> data(HapMap)
> HapMap[1:2,1:5]
      id group rs10399749 rs11260616 rs4648633
1 NA06985   CEU         CC         AA         TT
2 NA06993   CEU         CC         AT         CT
> HapMap.SNPs.pos[1:3,]
         snp chromosome position
1 rs10399749       chr1    45162
2 rs11260616       chr1  1794167
3  rs4648633       chr1  2352864
```

GWA analysis: class WGassociation

- Create object of class setupSNP

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

```
> myDat.HapMap[1:2,1:5]
```

```
      id group rs10399749 rs11260616 rs4648633
1 NA06985   CEU       C/C       A/A      T/T
2 NA06993   CEU       C/C       A/T      C/T
```

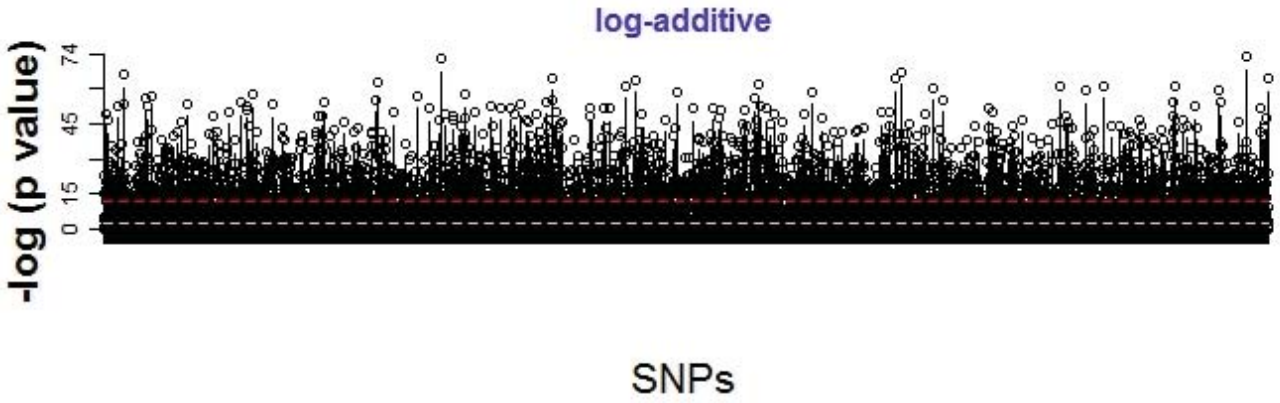
- Perform GWA on object of class setupSNP

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

GWA analysis: class WGassociation

- Plot results of GWA analysis

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



GWA analysis: class WGassociation

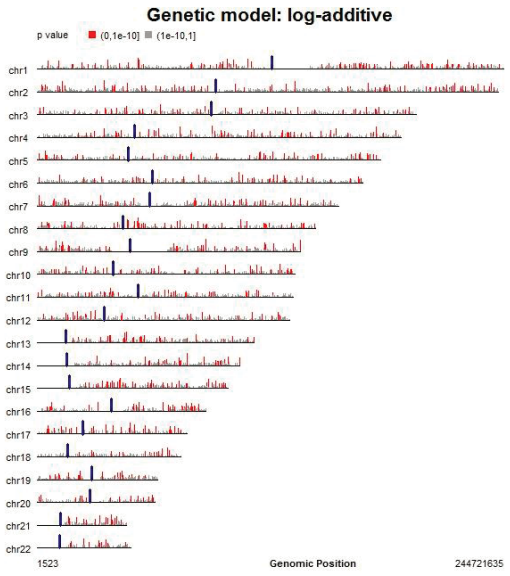
- Summarize results of GWA analysis

```
> 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
chr4      622             6.3            17.7               104 16.7
chr5      587             4.4            14.7               118 20.1
chr6      556             4.1            16.9               101 18.2
...
```

GWA analysis: class WGassociation

- Plot results of GWA analysis (alternative using whole=TRUE)

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



GWA analysis: class WGassociation

- Scanning is fast alternative when only p-values are needed

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

> summary(resHapMap.scan)
```

	SNPs (n)	Genot error (%)	Monomorphic (%)	Significant* (n)	(%)
chr1	796	0	18.6	143	18.0
chr2	789	0	13.9	143	18.1
chr3	648	0	13.0	115	17.7
chr4	622	0	17.7	92	14.8
chr5	587	0	14.7	104	17.7
chr6	556	0	16.9	86	15.5

…

Performing variety of analyses: significant SNPs

- Get significant SNPs from chromosome 5

```
> getSignificantSNPs(resHapMap,chromosome=5)

$names

 [1] "rs6555568"  "rs4702723"  "rs4866272"  "rs7720894"  "rs6452430"  "rs10067664"
"rs6880750"  "rs267030"   "rs179194"   "rs809039"   "rs1015565"  "rs6871275"
"rs1864998"  "rs263890"

[15] "rs11955678" "rs1702380"  "rs1106986"

$column

 [1] 6726 6742 6807 6927 6985 7022 7099 7101 7107 7123 7143 7157 7204 7260 7268
7277 7290
```

Performing variety of analyses: binary trait

- Association of case-control status with single SNP

```
> association(casco~snp(snp10001,sep=""), data=SNPs)
SNP: snp10001, sep = ""  adjusted by:
            0    %   1    %   OR lower upper p-value   AIC
Codominant
T/T         24 51.1  68 61.8 1.00             0.1323 193.6
C/T         21 44.7  32 29.1 0.54  0.26  1.11
C/C          2  4.3  10  9.1 1.76  0.36  8.64
Dominant
T/T         24 51.1  68 61.8 1.00             0.2118 194.1
C/T-C/C     23 48.9  42 38.2 0.64  0.32  1.28
…
log-Additive
0,1,2       47 29.9 110 70.1 0.87  0.51  1.47  0.5945 195.4
```

Performing variety of analyses: binary trait

- Alternative implementation

```
> myData<-setupSNP(data=SNPs,colSNPs=6:40,sep="")
> association(casco~snp10001, data=myData)
```

- Restrict to certain genetic models

```
> association(casco~snp10001, data=myData, model=c("cod","log"))
```

Performing variety of analyses: adjustment

- Adjust analysis for gender and arterial blood pressure

```
> association(casco~sex+snp10001+blood.pre, data=myData)
SNP: snp10001  adjusted by: sex blood.pre
            0    %   1    %   OR lower upper p-value   AIC
Codominant
T/T         24 51.1  68 61.8 1.00            0.15410 195.8
C/T         21 44.7  32 29.1 0.55  0.26  1.14
C/C          2  4.3  10  9.1 1.74  0.35  8.63
Dominant
T/T         24 51.1  68 61.8 1.00            0.22859 196.1
C/T-C/C     23 48.9  42 38.2 0.65  0.32  1.31
…
log-Additive
0,1,2       47 29.9 110 70.1 0.87  0.51  1.49 0.60861 197.3
```

Performing variety of analyses: stratification

- Stratify analysis by gender

```
> association(casco~snp10001+blood.pre+strata(sex), data=myData, model="dom")
      strata: sex=Male
SNP: snp10001  adjusted by: blood.pre
        0    %  1    %   OR lower upper p-value  AIC
Dominant
T/T      11 52.4 29 53.7 1.00              0.895 94.7
C/T-C/C  10 47.6 25 46.3 0.93  0.34  2.57
      strata: sex=Female
SNP:  adjusted by:
        0  %  1    %   OR lower upper p-value   AIC
Dominant
T/T      13 50 39 69.6 1.00             0.1309 100.8
C/T-C/C  13 50 17 30.4 0.47  0.17  1.25
```

Performing variety of analyses: subsetting

- Analyze within subset of males

```
> association(casco~snp10001+blood.pre, data=myData,subset=sex=="Male")
SNP: snp10001  adjusted by: blood.pre
            0    %  1    %   OR lower upper p-value  AIC
Codominant
T/T        11  52.4 29 53.7 1.00            0.04070 90.3
C/T        10  47.6 17 31.5 0.63  0.22  1.80
C/C         0   0.0  8 14.8       0.00
Dominant
T/T        11  52.4 29 53.7 1.00            0.89492 94.7
C/T-C/C    10  47.6 25 46.3 0.93  0.34  2.57
…
log-Additive
0,1,2      21  28.0 54 72.0 1.35  0.62  2.95 0.44244 94.1
```

Performing variety of analyses: continuous trait

- Analyze continuous trait

```
> association(log(protein)~snp100029+blood.pre, data=myData)
SNP: snp100029  adjusted by: blood.pre
              n     me      se      dif   lower    upper   p-value   AIC
Codominant
G/G           94 10.620 0.05449  0.00000                  3.319e-05 311.6
A/G           48 10.414 0.10043 -0.20457 -0.4289  0.01981
A/A           14  9.793 0.28182 -0.82447 -1.1869 -0.46206
Dominant
G/G           94 10.620 0.05449  0.00000                  1.553e-03 319.6
A/G-A/A       62 10.274 0.10461 -0.34408 -0.5572 -0.13098
…
log-Additive
0,1,2                   -0.33595 -0.4914 -0.18049 2.281e-05 312.2
```

Medium scale analysis

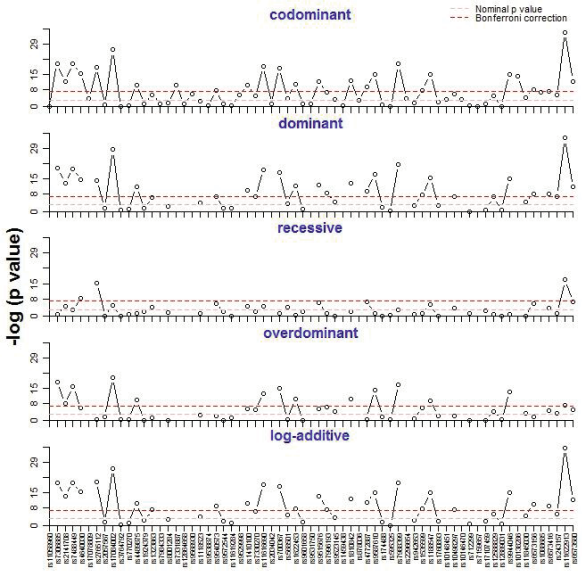
- Analyze subset of SNPs selected from previous analysis

```
> sigSNPs<-getSignificantSNPs(resHapMap,chromosome=5,sig=5e-8)$column
> myDat2<-setupSNP(HapMap, colSNPs=sigSNPs, sep="")
> resHapMap2<-WGassociation(group~1, data=myDat2)
> summary(resHapMap2)
 SNPs (n) Genot error (%) Monomorphic (%) Significant* (n)  (%)
       86            5.8            16.3                13 15.1
 *Number of statistically significant associations at level 1e-06
```

Medium scale analysis

- Plot results of medium scale analysis

```
> plot(resHapMap2,cex=0.8)
```



Medium scale analysis

- Analyze multiple SNPs

```
> 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)
> ans
          comments    codominant dominant recessive overdominant log-additive
snp10004  Monomorphic -          -        -         -           -
snp10007  Monomorphic -          -        -         -           -
snp100010 Monomorphic -          -        -         -           -
snp10002  -           0.78525    0.93292  0.48600   0.87267     0.76807
snp10003  Monomorphic -          -        -         -           -
snp10008  -           0.20293    0.29843  0.08453   0.83628     0.13289
…
```

Medium scale analysis

- Export results to LaTeX

```
> library(Hmisc)

> SNP<-pvalues(ans)

> out<-latex(SNP,file="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")
```

- This creates a latex file ans1.tex containing the table of results

Medium scale analysis

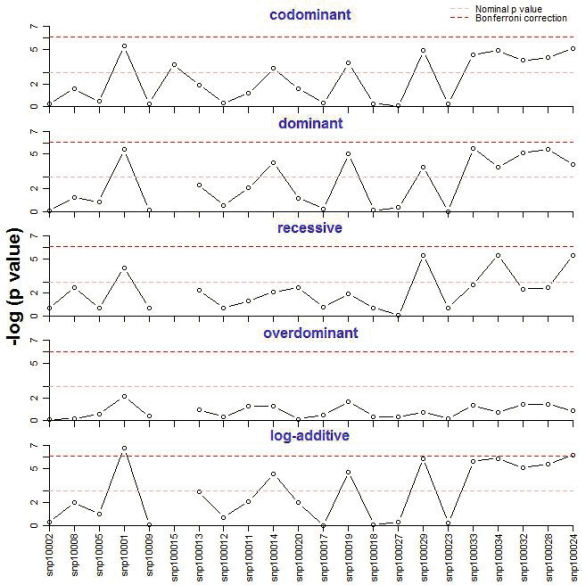
- One can also get the same output as for single SNP analyses

```
> WGstats(ans,dig=5)
…
$snp100010
SNP: snp100010  adjusted by:
Monomorphic
$snp10002
SNP: snp10002  adjusted by:
            n    me   se    dif  lower upper p-value  AIC
Codominant
C/C         74 42876 2890    0.0              0.7853 3612
A/C         78 42740 2576 -135.8  -7648  7377
A/A          5 50262 6879 7385.6 -14006 28777
…
```

Medium scale analysis

- Plot results

> plot(ans)



Haplotype analysis using haplo.stats

- Prepare model matrix with tag SNPs

```
> datSNP<-setupSNP(SNPs,6:40,sep="")
> tag.SNPs<-c("snp100019", "snp10001", "snp100029")
> geno<-make.geno(datSNP,tag.SNPs)
```

- Estimate haplotype effects

```
> mod<-
haplo.glm(log(protein)~geno,data=SNPs,family=gaussian,locus.label=tag.SNPs,allele.lev=attributes(geno)$unique.alleles,control = haplo.glm.control(haplo.freq.min=0.05))
```

Haplotype analysis using haplo.stats

- Output

```
> mod
Coefficients:
              coef     se  t.stat     pval
(Intercept) 10.6880 0.0985 108.543 0.00e+00
geno.3      -0.3485 0.0859  -4.058 7.86e-05
geno.6      -0.0466 0.0994  -0.469 6.40e-01
geno.rare   -0.2324 0.2429  -0.957 3.40e-01
Haplotypes:
          snp100019 snp10001 snp100029 hap.freq
geno.3            G       C         A   0.2321
geno.6            G       T         G   0.2990
geno.rare         *       *         *   0.0262
haplo.base        C       T         G   0.4427
```

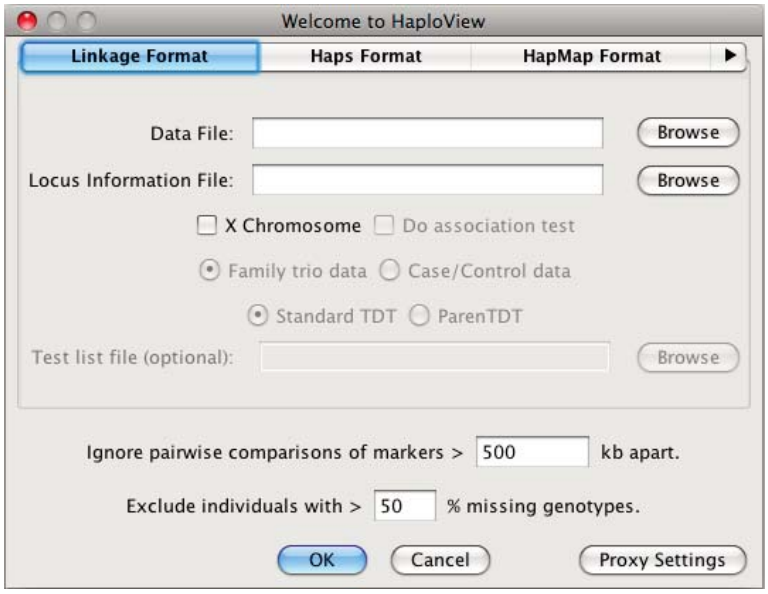
Haplotype analysis using haplo.stats

- Confidence intervals can be obtained

```
> intervals(mod)
              freq   diff   95%   C.I.   P-val
       CTG 0.4427  10.69 Reference haplotype
         G 0.2321  -0.35 (  -0.52 -  -0.18 )  0.0000
         G 0.2990  -0.05 (  -0.24 -   0.15 )  0.6391
      rare 0.0262  -0.23 (  -0.71 -   0.24 )  0.3386
```

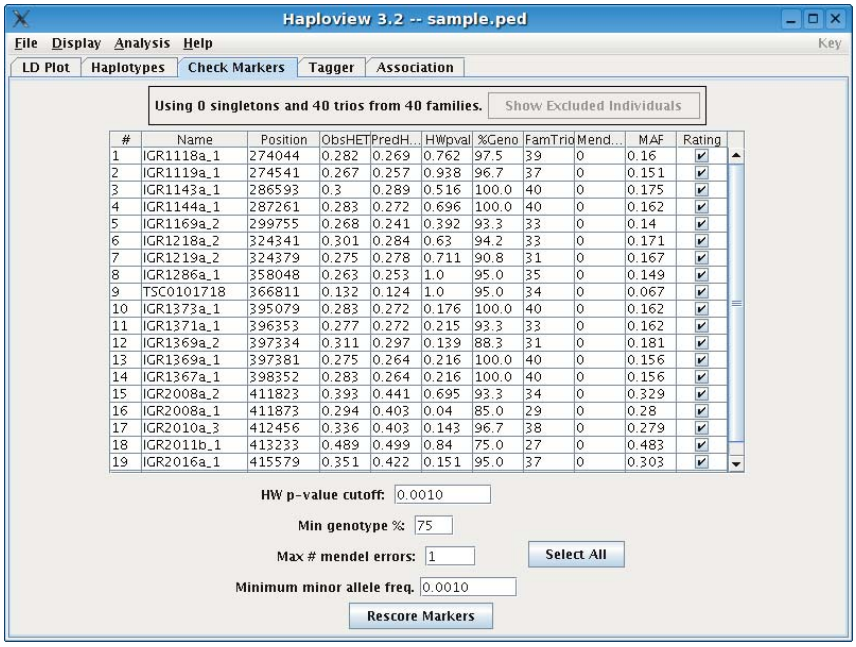
Haploview: load data

- Double click on Java archive Haploview.jar
- Browse to example files sample.txt and sample.info and push OK



Haploview: check markers tab

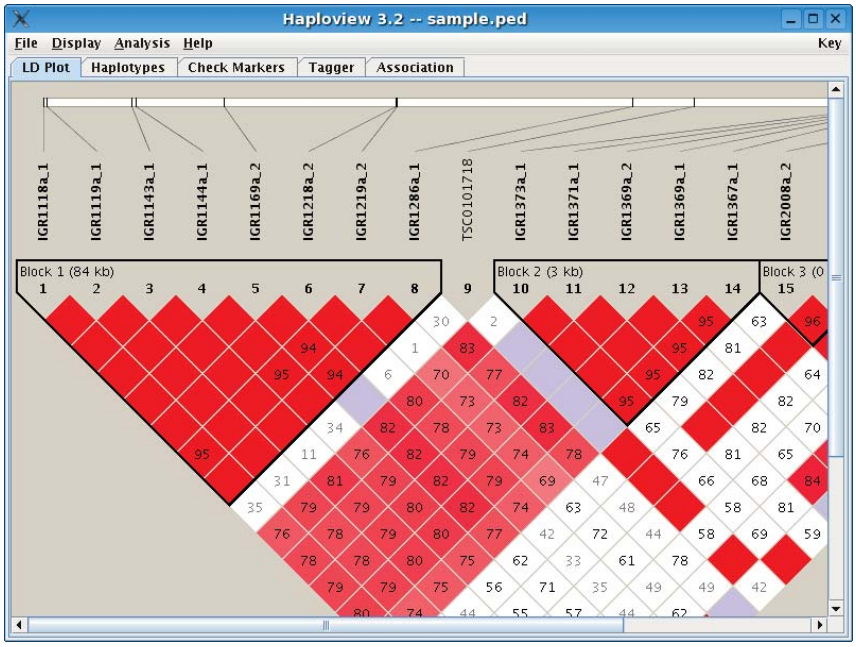
- Marker quality control



#	Name	Position	ObsHET	PredH...	HWpval	%Geno	FamTrio	Mend...	MAF	Rating
1	IGR1118a_1	274044	0.282	0.269	0.762	97.5	39	0	0.16	✔
2	IGR1119a_1	274541	0.267	0.257	0.938	96.7	37	0	0.151	✔
3	IGR1143a_1	286593	0.3	0.289	0.516	100.0	40	0	0.175	✔
4	IGR1144a_1	287261	0.283	0.272	0.696	100.0	40	0	0.162	✔
5	IGR1169a_2	299755	0.268	0.241	0.392	93.3	33	0	0.14	✔
6	IGR1218a_2	324341	0.301	0.284	0.63	94.2	33	0	0.171	✔
7	IGR1219a_2	324379	0.275	0.278	0.711	90.8	31	0	0.167	✔
8	IGR1286a_1	358048	0.263	0.253	1.0	95.0	35	0	0.149	✔
9	TSC0101718	366811	0.132	0.124	1.0	95.0	34	0	0.067	✔
10	IGR1373a_1	395079	0.283	0.272	0.176	100.0	40	0	0.162	✔
11	IGR1371a_1	396353	0.277	0.272	0.215	93.3	33	0	0.162	✔
12	IGR1369a_2	397334	0.311	0.297	0.139	88.3	31	0	0.181	✔
13	IGR1369a_1	397381	0.275	0.264	0.216	100.0	40	0	0.156	✔
14	IGR1367a_1	398352	0.283	0.264	0.216	100.0	40	0	0.156	✔
15	IGR2008a_2	411823	0.393	0.441	0.695	93.3	34	0	0.329	✔
16	IGR2008a_1	411873	0.294	0.403	0.04	85.0	29	0	0.28	✔
17	IGR2010a_3	412456	0.336	0.403	0.143	96.7	38	0	0.279	✔
18	IGR2011b_1	413233	0.489	0.499	0.84	75.0	27	0	0.483	✔
19	IGR2016a_1	415579	0.351	0.422	0.151	95.0	37	0	0.303	✔

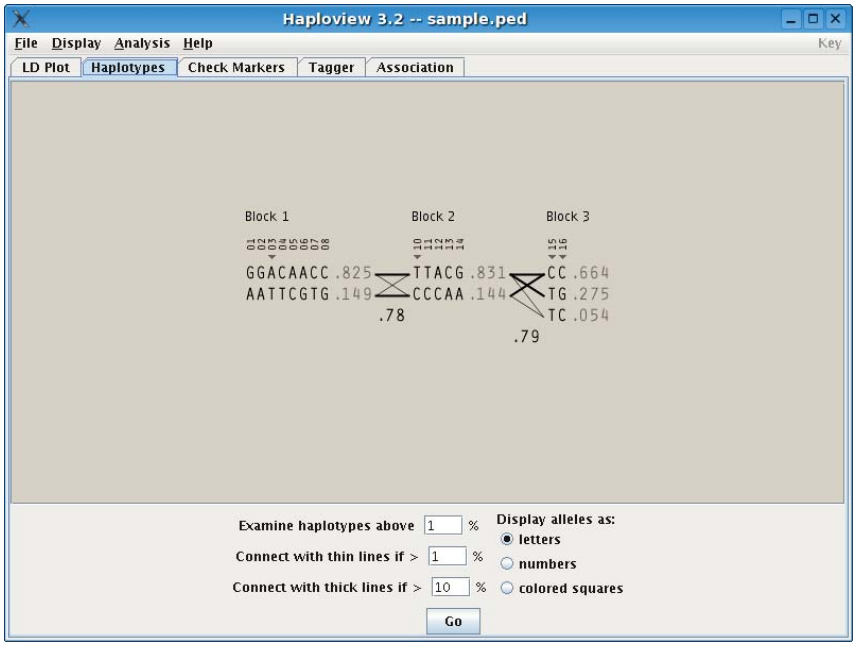
Haploview: LD plot tab

- Pairwise LD (D') and haplotype blocks



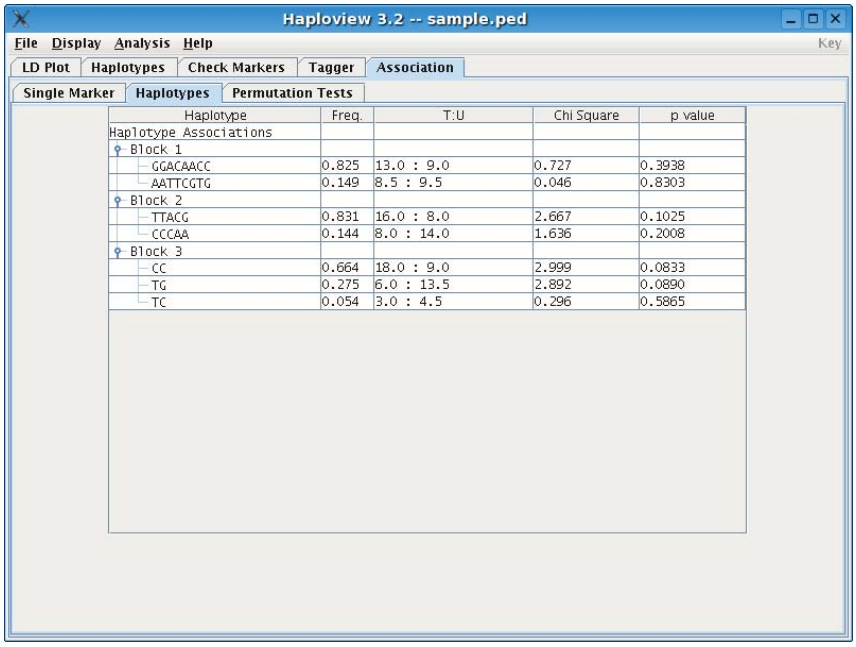
Haploview: haplotypes tab

- Haplotypes, haplotype frequencies and multilocus D’



Haploview: association tab

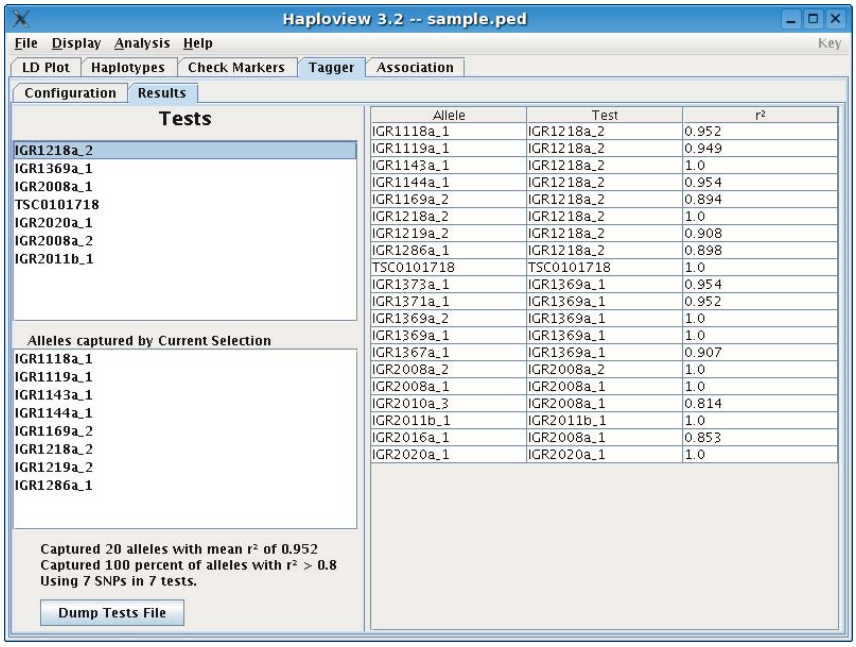
- Haplotype analysis (if indicated at start)



Haplotype	Freq.	T:U	Chi Square	p value
Haplotype Associations				
Block 1				
GGACAACC	0.825	13.0 : 9.0	0.727	0.3938
AATTCGTG	0.149	8.5 : 9.5	0.046	0.8303
Block 2				
TTACG	0.831	16.0 : 8.0	2.667	0.1025
CCCAA	0.144	8.0 : 14.0	1.636	0.2008
Block 3				
CC	0.664	18.0 : 9.0	2.999	0.0833
TG	0.275	6.0 : 13.5	2.892	0.0890
TC	0.054	3.0 : 4.5	0.296	0.5865

Haploview: tagger tab

- Tag SNPs selection



Class exercises

- Q1. Perform a genome-wide scan for HWE for the HapMap dataset.
- Q2. Perform a GWA for arterial blood pressure under the additive genetic model stratified by gender and adjusted for protein level.
- Q3. Determine the median protein level in the SNPs dataset and define a new dichotomous trait protein>median. Perform a GWA for this trait under the dominant genetic model and adjusted for gender.

Multiple testing: Bonferroni

- Recall medium-scale analysis of SNPs data

```
> library(SNPassoc)
> data(SNPs)
> myData<-setupSNP(data=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)
> ans
          comments    codominant dominant recessive overdominant log-additive
snp10004  Monomorphic -          -       -         -            -
…
snp10002  -           0.78525    0.93292  0.48600   0.87267      0.76807
…
```

Multiple testing: Bonferroni

- Bonferroni correction for number of tests performed

```
> Bonferroni.sig(ans, model="log-add", alpha=0.05,include.all.SNPs=FALSE)
number of tests:  21
alpha: 0.05
corrected alpha: 0.002380952
          comments log-additive
snp10001  -        0.001143723
snp100024 -        0.002231790
```

- The corrected alpha equals alpha divided by number of tests

```
> 0.05/21
[1] 0.002380952
```

Multiple testing: false discovery rate

- Recall medium-scale analysis of HapMap data

```
> data(HapMap)

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

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

> 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
chr4	622	6.3	17.7	104	16.7

…

Multiple testing: false discovery rate

- Get p-values and remove monomorphic SNPs

```
> pval<-additive(resHapMap)
> pval<-pval[!is.na(pval)]
```

- Calculate q-values

```
> install.packages('qvalue')
> library(qvalue)
> qobj<-qvalue(pval)
> qobj$qvalues[1:4]
[1] 1.128563e-01 2.309632e-07 2.930540e-10 2.777937e-01
```

- Obtaining the false discovery rate (FDR) for e.g. p-value 0.001

```
> max(qobj$qvalues[qobj$pvalues <= 0.001])
[1] 0.0006046515
```

Multiple testing: multtest package

- Install multtest package

```
> source("http://www.bioconductor.org/biocLite.R")
> biocLite("Biobase")
> install.packages('multtest')
> library(multtest)
```

- Apply several multiple testing strategies

```
> procs<-c("Bonferroni","Holm","Hochberg","SidakSS","SidakSD","BH","BY")
> res2<-mt.rawp2adjp(pval,procs)
> res2$adjp[1:10,]
            rawp   Bonferroni         Holm     Hochberg SidakSS SidakSD         BH         BY
 [1,] 1.740932e-32 1.274362e-28 1.274362e-28 1.274362e-28       0       0 1.274362e-28 1.207541e-27
 [2,] 3.914510e-32 2.865421e-28 2.865030e-28 2.865030e-28       0       0 1.432711e-28 1.357586e-27
...
```

Multiple testing: multtest package

- Obtain number of rejected hypotheses at various significance levels

```
> mt.reject(res2$adjp,seq(0,0.1,0.001))$r
      rawp Bonferroni Holm Hochberg SidakSS SidakSD   BH   BY
0        0          0    0       0      220      220    0    0
0.001 3342       1518 1537    1537     1518     1537 3099 2453
0.002 3549       1591 1650    1650     1591     1650 3322 2642
0.003 3731       1671 1705    1705     1671     1705 3487 2779
0.004 3785       1710 1782    1782     1711     1782 3540 2829
0.005 3845       1751 1811    1811     1751     1812 3611 2875
0.006 3893       1800 1831    1831     1800     1831 3735 2926
0.007 4009       1817 1855    1855     1817     1855 3764 3035
0.008 4045       1831 1873    1873     1832     1874 3801 3051
...
```

Multiple testing: permutation tests

- Permute cases and controls 1000 times

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

> summary(resHapMap.perm)
```

	SNPs (n)	Genot error (%)	Monomorphic (%)	Significant* (n)	(%)
chr1	796	0	18.6	143	18.0
chr2	789	0	13.9	143	18.1
chr3	648	0	13.0	115	17.7
chr4	622	0	17.7	92	14.8
chr5	587	0	14.7	104	17.7
chr6	556	0	16.9	86	15.5

…

Multiple testing: permutation tests

- Perform the actual permutation test calculations

```
> res.perm<- permTest(resHapMap.perm)
> 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
```

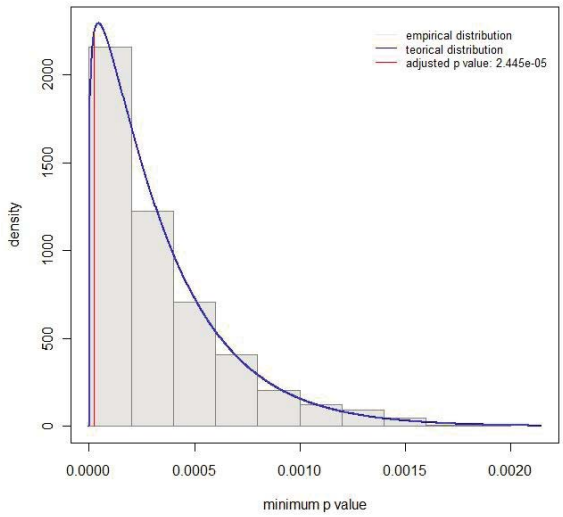
- Get the p-values in the permuted datasets

```
> perms <- attr(resHapMap.perm, "pvalPerm")
> dim(perms)
[1] 9305 1000
```

Multiple testing: permutation tests

- Plot permutation test results

```
> plot(res.perm)
```



Multiple testing: permutation tests

- Rank truncated product [Dudbridge et al. 2006] is also implemented

```
> 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
```

Interaction analysis: GxE

- Analyze SNP interacting with gender

```
> ans<-association(log(protein)~snp10001*sex+blood.pre,data=myData,model="codominant")
> print(ans,dig=2)
      SNP: snp10001  adjusted by: blood.pre
 Interaction
       Male       dif lower upper    Female         dif lower upper
T/T 40   11 0.08  0.00    NA    NA 52   10.6 0.079 -0.026 -0.29  0.24
C/T 27   11 0.10 -0.13 -0.45  0.19 26   10.2 0.184 -0.472 -0.79 -0.15
C/C  8   10 0.35 -0.64 -1.13 -0.14  4    9.8 0.286 -0.887 -1.56 -0.22
p interaction: 0.36051
```

Interaction analysis: GxE

- Analyze SNP interacting with gender: more output

sex within snp10001
T/T

	n	me	se	dif	lower	upper
Male	40	11	0.080	0.000	NA	NA
Female	52	11	0.079	-0.026	-0.29	0.24

C/T

	n	me	se	dif	lower	upper
Male	27	11	0.10	0.00	NA	NA
Female	26	10	0.18	-0.34	-0.69	0.0086

C/C

	n	me	se	dif	lower	upper
Male	8	10.0	0.35	0.00	NA	NA
Female	4	9.8	0.29	-0.25	-1.0	0.53

p trend: 0.26575

snp10001 within sex
Male

	n	me	se	dif	lower	upper
T/T	40	11	0.08	0.00	NA	NA
C/T	27	11	0.10	-0.13	-0.45	0.19
C/C	8	10	0.35	-0.64	-1.13	-0.14

Female

	n	me	se	dif	lower	upper
T/T	52	10.6	0.079	0.00	NA	NA
C/T	26	10.2	0.184	-0.45	-0.75	-0.14
C/C	4	9.8	0.286	-0.86	-1.52	-0.20

p trend: 0.36051

Interaction analysis: GxG

- Analyze two interacting SNPs

```
> ans<-association(log(protein)~snp10001*factor(recessive(snp100019))+blood.pre,data=myData,
model="codominant")

> print(ans,dig=2)

     SNP: snp10001  adjusted by: blood.pre

 Interaction

      G/G-C/G        dif lower  upper    C/C        dif lower upper

T/T 60      11 0.063  0.00    NA     NA 32  11 0.11 -0.038 -0.32  0.24

C/T 53      10 0.106 -0.30 -0.54 -0.053  0   0 0.00     NA    NA    NA

C/C 12      10 0.244 -0.72 -1.13 -0.313  0   0 0.00     NA    NA    NA

p interaction: NA
```

Interaction analysis: GxG

- Analyze two interacting SNPs: more output

factor(recessive(snp100019)) within snp10001
T/T
n me se dif lower upper
G/G-C/G 60 11 0.063 0.000 NA NA
C/C 32 11 0.112 -0.038 -0.32 0.24
C/T
n me se dif lower upper
G/G-C/G 53 10 0.11 0 NA NA
C/C 0 0 0.00 NA NA NA
C/C
n me se dif lower upper
G/G-C/G 12 10 0.24 0 NA NA
C/C 0 0 0.00 NA NA NA
p trend: NA

snp10001 within factor(recessive(snp100019))
G/G-C/G
n me se dif lower upper
T/T 60 11 0.063 0.00 NA NA
C/T 53 10 0.106 -0.30 -0.54 -0.053
C/C 12 10 0.244 -0.72 -1.13 -0.313
C/C
n me se dif lower upper
T/T 32 11 0.11 0 NA NA
C/T 0 0 0.00 NA NA NA
C/C 0 0 0.00 NA NA NA
p trend: NA

Interaction analysis: GxG

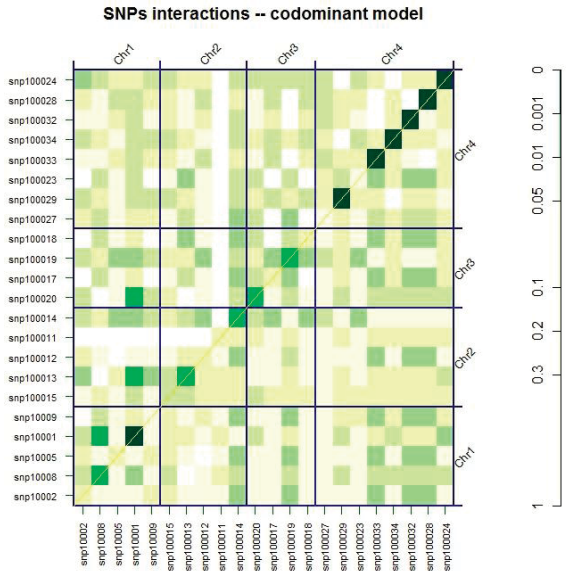
- Study gene-gene interaction

```
> ansCod<-interactionPval(log(protein)~sex, data=myData.o,model="codominant")
> ansCod[1:7,1:7]
          snp10004 snp10007 snp100010  snp10002 snp10003   snp10008  snp10005
snp10004        NA       NA        NA        NA       NA         NA        NA
snp10007        NA       NA        NA        NA       NA         NA        NA
snp100010       NA       NA        NA        NA       NA         NA        NA
snp10002        NA       NA        NA 0.4670088       NA 0.06423172 0.4187811
snp10003        NA       NA        NA        NA       NA         NA        NA
snp10008        NA       NA        NA 0.6488757       NA 0.00577702 0.6412163
snp10005        NA       NA        NA 0.6984826       NA 0.72232141 0.3777925
```

Interaction analysis: GxG

- Plot results of interaction analysis

> plot(ansCod)



Interactions and CART

- Trees allow discovery of a specific form of conditional association
- Trees do not specifically search for statistical interaction
- Consider the situation of a trait **y** with a very strong independent effect of covariate $\mathbf{x}_1$ such that the first split of the tree is on $\mathbf{x}_1$
- Suppose there is a second predictor $\mathbf{x}_2$ and that there is statistical interaction between $\mathbf{x}_1$ and $\mathbf{x}_2$, i.e. there is a difference γ in effect of $\mathbf{x}_2$ for both levels $\mathbf{x}_1 = 0$ and $\mathbf{x}_1 = 1$
- Suppose that there is also a variable $\mathbf{x}_3$ with an independent effect on **y**, regardless of the level of $\mathbf{x}_1$
- Formally we are looking at the linear model

  $\mathbf{y} = \beta_0 + \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 + \gamma \mathbf{x}_1 \mathbf{x}_2 + \beta_3 \mathbf{x}_3 + \varepsilon$

Interactions and CART

- After the initial split on $\mathbf{x}_1$, the model becomes
  - for $\mathbf{x}_1 = 0$: $\mathbf{y} = \beta_0 + \beta_2 \mathbf{x}_2 + \beta_3 \mathbf{x}_3 + \varepsilon$
  - for $\mathbf{x}_1 = 1$: $\mathbf{y} = (\beta_0 + \beta_1) + (\beta_2 + \gamma) \mathbf{x}_2 + \beta_3 \mathbf{x}_3 + \varepsilon$
- The next split within the daughter nodes depends on relative magnitude of the regression coefficients
- E.g. if β_3 is large compared to β_2 and $\beta_2 + \gamma$, it is likely that the next split will be on the variable $\mathbf{x}_3$ in both daughter nodes, although only $\mathbf{x}_1$ and $\mathbf{x}_2$ interact statistically
- Hence, for trees conditional association is more relevant than statistical interaction

Exercises

- Within chromosome 6 of the HapMap data perform an association analysis of the group variable using the dominant model. Correct for multiple testing using different approaches that control the family-wise error rate at 5% (e.g. Bonferroni, permutations), or that control the false discovery rate at 5% (e.g. Benjamini-Hochberg, qvalue approach)
- Investigate gene-environment interaction of snp100025 and sex in determining case-control status in the SNPs dataset, adjusted for protein level
- Visualize gene-gene interactions within chromosome 4 of the SNPs data with respect to the case-control status