

---

# Interaction analysis using PLINK, CASSI, R and BEAM3

---

---

## Overview

---

### Purpose

In this exercise you will be performing association analysis and testing for interaction effects using case/control data.

### Methodology

The methodology used includes chi-squared tests and logistic regression in PLINK and R, as well as variants of these approaches. We shall also use an R package ORMDR that implements a form of multifactor dimensionality reduction (MDR), and a Bayesian Epistasis Association Mapping method implemented in the program BEAM3.

Some of the commands that you will type in order to do the analysis may seem a bit mystifying if you are not familiar with using statistical packages such as R. Don't worry too much if you don't understand all the details! If you decide to regularly use a statistical package such as R to analyse your own data, it will be important to learn how to use that particular package appropriately through attendance on a training course or careful reading of an introductory textbook.

### Program documentation

#### PLINK documentation:

PLINK has an extensive set of documentation including a pdf manual, a web-based tutorial and web-based documentation:

<http://pngu.mgh.harvard.edu/~purcell/plink/>

#### CASSI documentation:

CASSI documentation is available from:

<http://www.staff.ncl.ac.uk/richard.howey/cassi/downloads.html>

#### R documentation:

The R website is at <http://www.r-project.org/>

From within R, one can obtain help on any command `xxxx` by typing ``help(xxxx)'`

#### BEAM documentation:

The (rather sketchy!) documentation provided with the BEAM3 package can be found here [BEAM3-README.txt](#)

---

---

## Exercise

---

### Data overview

The data consists of simulated genotype data at 100 SNP loci, typed in 2000 cases and 2000 controls. The data has been simulated in such a way that the first five SNPs have some relationship with disease, whereas the remaining 95 SNPs have no effect on disease outcome.

The complication with these data is that SNPs 1 and 2 have been simulated in such a way that they show no marginal association with the disease: their association will only be visible when you look at both SNPs in combination. SNPs 3-5 have been simulated to only have an effect on disease when an individual is homozygous at all three of these loci. Although potentially this could lead to marginal effects at the loci, formally this corresponds to a model of pure interaction, with no main effects, at these 3 SNPs.

### Appropriate data

Appropriate data for this exercise is genotype data for a set of linked or unlinked loci typed in a group of unrelated affected individuals (cases) and in a group of unaffected or randomly chosen individuals from the same population (controls).

All the programs will deal with much larger numbers of loci than the 100 SNPs considered here. PLINK, in particular, was specifically designed for the analysis of large numbers of loci e.g. generated as part of a genome-wide association study.

### Special considerations/restrictions (for the programs used here)

All the commands required to run these analyses are given below. However, packages such as R are sophisticated statistical programming packages and have much greater functionality than will be described here. If you intend to use a statistical package to analyse your data, you are strongly encouraged to learn how to use that package appropriately through attendance on a training course or careful reading of an introductory textbook.

---

## Instructions

---

### Data format

The data for the 100 SNPs [simcasecon.ped](#) is in standard linkage pedigree file format, with columns corresponding to family id, subject id (within family), father's id, mother's id, sex (1=m, 2=f), affection status (1=unaffected, 2=affected) and one column for each allele for each locus genotype. Note that since this is case/control rather than family data, there is only one individual per family and everyone's parents are coded as unknown.

PLINK requires an additional map file [simcasecon.map](#) describing the markers (in order) in the pedigree file. The PLINK-format map file contains exactly 4 columns:

```
chromosome (1-22, X, Y or 0 if unplaced)
rs number or snp identifier
Genetic distance (morgans) (not often used - so can be set to 0)
Base-pair position (bp units)
```

Take a look at the data files, and check that you understand how the data is coded. Then save the files as .txt files to the appropriate directory (folder) on your computer.

## Step-by-step instructions

### 1. Analysis in PLINK

Move to the directory where you have saved the data files.

To carry out a basic association analysis in PLINK, type

```
plink --noweb --ped simcasecon.ped --map simcasecon.map --assoc
```

The `--noweb` command tells PLINK to run without bothering to check via the web to see whether there are updated versions of PLINK. The `--ped xxxx` command tells PLINK that the name of the pedigree file is `xxxx` and the `--map yyyy` command tells PLINK that the name of the map file is `yyyy`. The `--assoc` command tells PLINK to perform a basic allele-based chisquared association test.

PLINK outputs some useful messages (you should always read these to make sure they match up with what you expect!) and outputs the results to a file `plink.assoc`.

Take a look at the file `plink.assoc` (e.g. by typing `more plink.assoc`). For each SNP the following columns of results are reported:

<b>CHR</b>	<b>Chromosome</b>
<b>SNP</b>	<b>SNP ID</b>
<b>BP</b>	<b>Physical position (base-pair)</b>
<b>A1</b>	<b>Minor allele name (based on whole sample)</b>
<b>F_A</b>	<b>Frequency of this allele in cases</b>
<b>F_U</b>	<b>Frequency of this allele in controls</b>
<b>A2</b>	<b>Major allele name</b>
<b>CHISQ</b>	<b>Basic allelic test chi-square (1df)</b>
<b>P</b>	<b>Asymptotic p-value for this test</b>
<b>OR</b>	<b>Estimated odds ratio (for A1, i.e. A2 is reference)</b>

Does there appear to be evidence of association at any of the five "true" loci? What about the 95 null loci?

Try performing a genotype-based (rather than an allele-based) analysis in PLINK and take a look at the results by typing the following 3 commands:

```
plink --noweb --ped simcasecon.ped --map simcasecon.map --model
head -1 plink.model
grep GENO plink.model
```

Again, does there appear to be evidence of association at any of the five "true" loci? What about the 95 null loci?

To test for pairwise epistasis in PLINK, the fastest option is to use the `--fast-epistasis` command:

```
plink --noweb --ped simcasecon.ped --map simcasecon.map --fast-epistasis
```

Results can be found in the file `plink.epi.cc`. Only pairwise interaction tests with  $p \leq 0.0001$  are reported (otherwise, for genome-wide studies, there would be too many results to report, given the large number of pairwise tests performed). You should find a very significant interaction between SNPs 1 and 2, and a number of less significant pairwise interactions. Since this is simulated data, we know that all of these less significant results are false positives.

A more powerful test for SNPs that are not in LD with one another (i.e. that are not too

close to one another, in terms of their genomic location) is to additionally use the `--case-only` option:

```
plink --noweb --ped simcasecon.ped --map simcasecon.map --fast-epistasis --case-only
```

Results can be found in the file `plink.epi.co`. Again only pairwise interaction tests with  $p \leq 0.0001$  are reported. You should again find a very significant interaction between SNPs 1 and 2 (even more significant than previously, owing to the increased power with a case-only test), and a number of less significant positive pairwise interactions. Most of these are false positives, but PLINK does appear to have detected the true positive interactions between SNPs 3 and 4, and between SNPs 4 and 5.

A problem with the `--fast-epistasis` test is that it can be affected by LD between the SNPs (although only the case-only test is seriously affected). A more accurate test is to use the slower `--epistasis` command:

```
plink --noweb --ped simcasecon.ped --map simcasecon.map --epistasis
```

Do you find that the program runs more slowly? Results can again be found in the file `plink.epi.cc` (which will now have been overwritten). You can see that now the interaction between SNPs 1 and 2 remains highly significant, together with just one other (false positive) interaction between SNPs 15 and 77.

Since the `--epistasis` option is slower, but most accurate, for genome-wide studies it might be sensible to first to screen for interactions using the `--fast-epistasis` command, but then confirm any findings using the `--epistasis` command on the smaller set of detected SNPs.

Before we finish with PLINK, we will use PLINK to create a differently-formatted data file that will be required later:

```
plink --noweb --ped simcasecon.ped --map simcasecon.map --recodeA --out recoded
```

This creates a new file `recoded.raw` in which the genotype data for each SNP is coded as a single count of the minor allele (0, 1, 2) rather than as two alleles (1 1, 1 2, 2 2 etc). Take a look at the file `recoded.raw` and check you understand this new format.

## 2. Analysis in CASSI

We will also compare our PLINK results with those obtained using the CASSI program, which implements an improved Joint Effects (JE) test of pairwise interaction as described in Ueki and Cordell (2012). First we need to convert our data to PLINK binary format:

```
plink --noweb --ped simcasecon.ped --map simcasecon.map --make-bed --out jeformat
```

This should create PLINK binary format files `jeformat.bed`, `jeformat.bim` `jeformat.fam`. Then we use the CASSI program with the input file `jeformat.bed` to perform pairwise interaction tests at all pairs of loci. (By default, only those pairs of SNPs showing interaction with a P value  $< 0.0001$  are output, though this can be changed if desired):

```
cassi -i jeformat.bed
```

Take a look at the output file `cassi.out`. The most important columns are the first 4 columns (listing the SNP numbers/names) and the last 4 columns listing the case/control and case-only interaction test chi-squareds and p values. It can be quite hard to work out which column is which, so we suggest you start up R by typing

```
R
```

and then read in the results by typing

```
je <-read.table("cassi.out", header=T)
```

and then take a look at them by typing

```
je
```

You can see that SNPs 1 and 2 show a very strong pairwise interaction (Case.Con\_Pval=0; Case.Only\_Pval=0). Interestingly we also detect, at lower significance levels, the pairwise interactions between SNPs 3 and 4 (Case.Con\_Pval=0.018; Case.Only\_Pval=0.000049) and between SNPs 4 and 5 (Case.Con\_Pval=0.00038; Case.Only\_Pval=0.000013). We also detect two false positive interactions, between SNPs 15 and 77, and between SNPs 31 and 100.

### 3. Chi-squared, logistic regression and MDR analysis in R

Stay within the R package. To start with, you will need to read in the original data:

```
simcasecon <- read.table("simcasecon.ped", header=F, na.strings = "0")
```

The "`<-`" operator stands for "is defined as" (or "is assigned as") and is used a lot in R to create new variables, new dataframes or new R objects.

Here this command reads your data into what is called a dataframe, essentially a large matrix with columns corresponding to the different variables. You chose to name the dataframe 'simcasecon' and you can look at the top 6 lines by typing

```
head(simcasecon)
```

Each variable can be accessed by using the name of the dataframe followed by a \$ sign, followed by the variable name. E.g. to look at the column of pedigree names, you just type

```
simcasecon$V1
```

(note that these wrap round rather than looking like a column).

Tell R to automatically look at variables in the simcasecon dataframe (without having to type `simcasecon` every time) by typing:

```
attach(simcasecon)
```

To perform the analysis we will first generate a variable called 'case' that codes affected/unaffected individuals as 1/0 rather than 2/1 (as they are coded in the 'affected' variable V6):

```
case <- V6-1
```

To look at the new case variable, type

```
case
```

To compare with the old 'affected' variable, type

```
V6
```

Now we will generate genotype variables from the two alleles at each of the true loci:

```
g1 <- V7+V8-2
g2 <- V9+V10-2
g3 <- V11+V12-2
g4 <- V13+V14-2
g5 <- V15+V16-2
```

This generates 5 different genotype variables which code for the genotypes at the 5 loci. Can you see how this has worked? The genotype values 0, 1, 2 correspond to the number of copies of the '2' allele that a person has. Take a look at the first of these genotype variables by typing

```
g1
```

To tabulate and perform chi-squared tests on each of the genotype variables (to see if they are associated with disease status), type:

```
table(g1,case)
chisq.test(g1,case)

table(g2,case)
chisq.test(g2,case)

table(g3,case)
chisq.test(g3,case)

table(g4,case)
chisq.test(g4,case)

table(g5,case)
chisq.test(g5,case)
```

Is there any evidence for association at these five loci? Your results should be very similar to what you found using the genotype-based test in PLINK.

To perform logistic regression, we can use the `glm` function in R, however this does not produce the output in a very convenient format. We will therefore read in another function `logistic`, which acts as a wrapper for `glm` but with a nicer output.

First read in the R code for the `logistic` function, from an external file `logregwrapper.R` :

```
source("logregwrapper.R")
```

To perform allele tests at each marker using logistic regression, we use the following commands:

```
logistic (case ~ g1)
logistic (case ~ g2)
logistic (case ~ g3)
logistic (case ~ g4)
logistic (case ~ g5)
```

For genotype analysis at each marker, use the following commands:

```
logistic (case ~ factor(g1))
logistic (case ~ factor(g2))
logistic (case ~ factor(g3))
logistic (case ~ factor(g4))
logistic (case ~ factor(g5))
```

Does there appear to be any evidence for association? Again, your results should be very similar to what you found using the allele- and genotype-based tests respectively in PLINK.

Now let us try analysing SNPs 1 and 2 together:

```
logistic (case ~ factor(g1)*factor(g2))
anova(logistic (case ~ factor(g1)*factor(g2)))
```

These commands first estimate the ORs for main effects (relative to the baseline category) and then for interaction effects (relative to baseline plus main effects). You should find several significant terms. The formal 4 df test of interaction is given by the "deviance" of 701.68 on 4 df. To see the significance of this type:

```
pchisq(701.68,4,lower.tail=F)
```

You should find a highly significant interaction effect. The overall 8 df test of association (allowing for interaction) has a total deviance of  $701.68+0.65+1.49=703.82$  (or equivalently  $5545.2-4841.3=703.9$ ). This is also highly significant, as you can tell by typing:

```
pchisq(703.82,8,lower.tail=F)
```

Let us repeat this type of analysis using SNPs 3, 4 and 5 (considered together, in a 3-way model):

```
logistic (case ~ factor(g3)*factor(g4)*factor(g5))
```

This model has so many terms it is quite difficult to interpret. From the column of ORs you should see the largest and most significant effect ( $OR=33.796$ ;  $p=1.227900e-05$ ) corresponds to when an individual is homozygous at all three loci. This is shown in the two lines (really one line that wraps around) marked

```
factor(g3)2:factor(g4)2:factor(g5)2
```

This result shows that these three loci act largely via a complex 3-way interaction.

To see the significance at these loci, type:

```
anova(logistic (case ~ factor(g3)*factor(g4)*factor(g5)))
```

The significance of any 3-way interaction terms is given by a deviance of 45.6 of 8 df. This is quite significant ( $p=2.8e-07$ ), as you can tell by typing:

```
pchisq(45.6,8,lower.tail=F)
```

The overall test of association, while allowing for interaction, is given by a deviance of  $5545.2-5450.5=94.7$  on 26 df. This is also quite significant ( $p=9.6e-10$ ), as you can tell by typing:

```
pchisq(94.7,26,lower.tail=FALSE)
```

While we are in R, we will try performing a form of MDR analysis that has been made available in an R package "ORMDR". (Note that the original MDR software, which is available as a Java program, is probably a bit more user-friendly than ORMDR). Read in the required ORMDR library:

```
library(ORMDR)
```

First we will read in the data in a form that is compatible with the ORMDR package. We first read in the data in the form given in the file `recoded.raw`, and take a look at the top of it (the first 6 rows) to remind ourselves what it looks like:

```
recoded<-read.table("recoded.raw", header=T)
head(recoded)
```

We then have to select just the 100 columns of SNP data together with the phenotype column, which needs to be coded as 1 and 0 (rather than 2 and 1). For convenience we will place the phenotype column at the end of the SNP data. That can be achieved as follows:

```
newdata<-recoded[7:106]
ormdrdata<-cbind(newdata,recoded$PHENOTYPE-1)
```

Take a look at the top of the new data frame you have created, to check you understand it, by typing:

```
head(ormdrdata)
```

The data in `ormdrdata` consists of 100 SNPs, with the phenotype (case/control status, coded 1/0) in column 101. To analyse this, looking at individual SNPs (i.e. groups of SNPs of size 1), and saving the results in an object "mdr1", type:

```
mdr1<-mdr.c(ormdrdata, colresp=101, cs=1, combi=1, cv.fold = 10)
```



The syntax `colresp=101` tells the program which column the "class" (phenotype) variable is in, and `cs=1` tells the program which code is used for cases. The syntax `combi=1` tells the program to consider single-locus combinations. The syntax `cv.fold = 10` tells the program to perform 10-fold cross validation. This means that the program divides the data into 10 equal parts, uses 9/10 of the data to fit and choose the best combination and then tests how well this combination performs for predicting the remaining 1/10 of the data. The whole process is repeated 10 times, for each of the different 1/10 portions of the data that could be predicted.

To see which was the best SNP (most predictive of disease status) in each of the ten cross validation replicates, type:

```
mdr1$min.comb
```

No SNP comes up consistently as best, although SNP 5 (which is a true associated SNP) comes up in some replicates. To repeat the process looking at all two-locus models (pairwise combinations of SNPs) type:

```
mdr2<-mdr.c(ormdrdata, colresp=101, cs=1, combi=2, cv.fold = 10)
mdr2$min.comb
```

Here we see that SNPs 1 and 2 consistently come up as the best two-locus combination, as might be expected from their strong interaction. To repeat the process looking at all three-locus combinations of SNPs, type:

```
mdr3<-mdr.c(ormdrdata, colresp=101, cs=1, combi=3, cv.fold = 10)
mdr3$min.comb
```

Again SNPs 1 and 2 consistently come up, together with some additional SNPs which are sometimes true positives (SNPS 3-5) and sometimes false positives.

To see how well the best SNP or SNPS in each cross validation sample actually do at predicting the relevant 1/10 of the data, take a look at the error rates:

```
mdr1$test.erate
mdr2$test.erate
mdr3$test.erate
```

The most appropriate number of loci to select for our final model is that number  $n$  which minimizes the average cross validation error. To work out the average cross validation error for each value of  $n$ , type:

```
mdr1mean<-mean(mdr1$test.erate)
mdr2mean<-mean(mdr2$test.erate)
mdr3mean<-mean(mdr3$test.erate)

mdr1mean
mdr2mean
mdr3mean
```

These results suggest that the minimum mean cross validation error occurs when  $n=2$ . To find which is the best 2-way combination, type:

```
mdr2$best.combi
```

This tells us that the best 2-way combination is SNPs 1 and 2. This is what would be expected from the cross-validation results:

```
mdr2$min.comb
```

which showed that SNPs 1 and 2 did best within each cross-validation sample. We therefore have a cross-validation consistency of 10/10. If the minimum cross-validation error had occurred when  $n=3$ , we would need to look at:

```
mdr3$best.combi
mdr3$min.comb
```



in which case the best 3-way combination would be SNPs 1, 2 and 3, but this has a cross validation consistency of around 8/10 (since in several of the cross validation samples, SNPs 1, 2 and 3 were not the best combination).

To get out of R, type `q()` (and reply `n` if it asks you to save the workspace)

---

#### 4. Analysis in BEAM3

Finally, we shall use the program BEAM3 to perform a Bayesian analysis to detect single-locus associations and multi-way interactions. For details of the methodology, please read the BEAM3 paper (Zhang Y (2011) Genet Epid 36: 36-47), or see the (rather sketchy!) BEAM3 documentation (link above).

BEAM3 requires a very different file format from the other programs. Essentially the role of SNPs and people is transposed, so that each line corresponds to a SNP and each column to the genotype (at that SNP) for a different person. The top line of the file gives the case/control status (coded 1/0). For more details see the BEAM3 documentation (link above).

We have prepared a file in the correct format [beam3data.txt](#). To run BEAM with specified input file and default values for the MCMC sampling scheme, type

```
BEAM3 beam3data.txt -o beam3results
```

This should produce an output file `posterior.beam3results` which lists the SNPs (numbered from 0 to 99) together with their posterior probabilities of showing marginal association, interaction association, and total association (= the marginal + interaction association probabilities).

Take a look at this file. Unfortunately BEAM3 does not seem to have picked up any associations. This is because it is hard for BEAM3 to detect SNPs that don't show any marginal association. However, we can use the `-T t` option which runs the program at "temperature" `t` in the first few iterations to try and force the program to jump out of local modes. This can help in finding interactions with no marginal association, although of course it is not guaranteed.

Try using this option with the temperature specified to start at 10 and then gradually drop to 1 as the MCMC procedure proceeds:

```
BEAM3 beam3data.txt -o beam3results -T 10
```

Take a look at the new output file `posterior.beam3results`. You should find that the first two SNPs (numbered 0 and 1) show high posterior probabilities of being involved in interaction associations. Therefore, SNP1 and SNP2 are detected via their interaction effects. No other markers are detected, so it seems as if the 3-way interaction between SNPs 4, 5, 6 is too weak to be detected in comparison to the stronger interaction between SNPs 1 and 2.

---

## Answers

---

### Interpretation of output

Answers and interpretation of the output are described in the step-by-step instructions. Please ask if you need help in understanding the output for any specific test.

---

## Comments

---

### Advantages/disadvantages

PLINK, CASSI and BEAM are designed for genome-wide studies, allowing the inclusion of many thousands of markers.

Analysis in a standard statistical package does not generally allow so many markers, but has the advantage of allowing a lot of extra flexibility with regards to the models and analyses performed e.g. it easy to include additional predictor variables such as environmental factors, gene-environment interactions etc. However, you are required to know or learn how to use the package in order to gain that extra flexibility, and to produce reliable results.

### Study design issues

With case/control data it is relatively easy to obtain large enough sample sizes to detect small genetic effects. However, detection of interactions generally requires much larger sample sizes.

### Other packages

Logistic regression analysis for detection of interactions can be performed in other statistical packages such as Stata, SAS, SPSS, GLIM.

MDR analysis can be performed in the Java MDR package. An alternative Bayesian Epistasis mapping approach is available in the BIA software from Jonathan Marchini. Several packages are available for implementing different data-mining and machine-learning approaches for detecting interactions or detecting association allowing for interaction. See Cordell (2009) (reference below) for more details.

---

## References

---

Cordell HJ (2009) Genome-wide association studies: Detecting gene-gene interactions that underlie human diseases. Nat Rev Genet. 2009 May 12. [Epub ahead of print]

Y Chung and S Y Lee and R C Elston and T Park (2007) Odds ratio based multifactor-dimensionality reduction method for detecting gene-gene interactions. Bioinformatics 23:71-76.

L W Hahn and M D Ritchie and J H Moore (2003) Multifactor dimensionality reduction software for detecting gene-gene and gene-environment interactions Bioinformatics 19:376--382.

Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, Maller J, Sklar P, de Bakker PIW, Daly MJ & Sham PC (2007) PLINK: a toolset for whole-genome association and population-based linkage analysis. American Journal of Human Genetics, 81:559-575.

Ritchie MD, Hahn LW, Roodi N, Bailey LR, Dupont WD, Parl FF and Moore JH (2001) Multifactor-dimensionality reduction reveals high-order interactions among estrogen-

metabolism genes in sporadic breast cancer. Am J Hum Genet 69:138-147.

Ueki M, Cordell HJ (2012) Improved statistics for genome-wide interaction analysis. PLoS Genetics 8(4):e1002625.

Zhang Y, Liu JS (2007) Bayesian inference of epistatic interactions in case-control studies. Nat Genet 39:1167-1173.

Zhang Y (2011) A novel Bayesian graphical model for genome-wide multi-SNP association mapping. Genet Epidemiol 36: 36-47.

---

*Exercises prepared by: Heather Cordell*

*Checked by:*

*Programs used: PLINK, R, DGCgenetics, ORMDR, BEAM*

*Last updated: 01/11/2013 12:24:13*