DGCgenetics vignette
Some extensions to the CRAN `genetics` package

David Clayton

November 22, 2007

The `DGCgenetics` package

The following exercises on genetic association studies accompanied a short course given by David Clayton, assisted by Chris Wallace, as part of the 2005 and 2007 EEPE programmes (`http://www.eepe.org/`). The exercises use the `genetics` package, written by Greg Warnes and Freidrich Leisch, together with some extensions written by David Clayton. These extensions, together with the datasets used in the exercises, are contained in the package `DGCgenetics`. The course also contained an exercise on the use of the `snpMatrix` package and this exercise is included as a vignette on this package.

To load the libraries and data:

```
> library(DGCgenetics)
```

```
NOTE:

  The R-Genetics project has developed an set of enhanced genetics
  packages that will shortly replace 'genetics'. Please visit the
  project homepage at http://rgenetics.org for more information.
```

Exercise 1: Practical exercise: sibling recurrence risk for leprosy

Introduction

Leprosy is a disease caused by infection with Mycobacterium leprae. Infection is necessary for disease, but it is thought that only about 10% of infections lead to clinical disease which may be manifested across a spectrum from paucibacillary (PB) to multibacillary (MB) disease. MB patients harbour live bacilli and should, therefore, be more infectious than PB patients, whose immune reaction kills the bacilli (while giving rise to distinctive symptoms).

Development of disease depends not only on infection, but also varies according to age (age-specific rates typically peak in teenagers and young adults) and vaccination history (BCG vaccination reduces the incidence of disease by a factor of 2. There is also evidence that host genetics affect the development of disease. Many linkage and association studies have shown the involvement of the HLA region and recently strong evidence has been found, in linkage analyses of sibling pairs from Indian and Vietnamese populations, for susceptibility loci on chromosomes 6, 10 and 20. Estimation of λ_R by usual case-control methods would be expected to give biased results of genetic effects since non-genetic risk factors, particularly exposure to the infectious agent, tend to cluster in families.

The Karonga Prevention Study (KPS) conducted two total population surveys between 1979 and 1989 in Karonga district, Northern Malawi, described by Ponnighaus et al. (1987). Current or past leprosy cases were identified by paramedical leprosy control assistants and data were also collected about familial relationships and non-genetic factors known to affect risk of disease. The data used in this practical concern prevalence of a history of leprosy at one of the cross-sectional surveys. Note that age of onset of disease was not known/recorded. Thus, the data are prevalence data and recurrence risks should be interpreted in terms of prevalence. This is not ideal, but is all that is possible given the cross-sectional natuire of the data. With some important assumptions, relative recurrence risks estimated from prevalence, may not be very different from those estimated using incidence data.

A first look at the data

The data file contains a record for pairs of full siblings identified in the study. Get the data file and carry out some initial tabulations:

```
> data(leprosy)
> summary(leprosy)
```

```
    s1sex                     s1bcg          s1aff        s1pbcon
 male  :7034   negative/uncertain:8271   Min.   :0.0000   No :10844
 female:7363   positive          :6126   1st Qu.:0.0000   Yes: 3553
                                         Median :0.0000
                                         Mean   :0.2100
                                         3rd Qu.:0.0000
                                         Max.   :1.0000
 s1mbcon         s1age         s2sex                       s2bcg
 No :13793   0 - 9  :2424   male  :6962   negative/uncertain:8878
 Yes:  604   10 - 19:5001   female:7435   positive          :5519
             20 - 29:3008
             30 - 44:2145
             45 - 74:1819

     s2aff        s2pbcon      s2mbcon         s2age              wt
 Min.   :0.0000   No :11005   No :13886   0 - 9  :4356   Min.   : 1.00
 1st Qu.:0.0000   Yes: 3392   Yes:  511   10 - 19:3430   1st Qu.: 1.00
 Median :0.0000                           20 - 29:2518   Median :20.00
 Mean   :0.1671                           30 - 44:2255   Mean   :13.10
 3rd Qu.:0.0000                           45 - 74:1838   3rd Qu.:20.00
 Max.   :1.0000                                          Max.   :20.00
     family          size
 Min.   :   1   Min.   : 2.000
 1st Qu.:1907   1st Qu.: 2.000
 Median :3883   Median : 3.000
 Mean   :3926   Mean   : 3.844
 3rd Qu.:5891   3rd Qu.: 5.000
 Max.   :7950   Max.   :12.000
```

```
> attach(leprosy)
> table(s1aff, s2aff)
```

```
     s2aff
s1aff    0    1
    0 9170 2203
    1 2821  203
```

All pairs in which leprosy was observed are included in the dataset , but only a 5% sample of those in which no leprosy was observed. Thus, to estimate population frequencies, leprosy-free pairs in the sample should be given a “case weight” of 20. This is an example of *inverse probability weighting* to correct for unequal sampling — cases are weighted by the reciprocal of the probability of their being selected in the sample. Recalculate the table using these weights:

```
> wtable(s1aff, s2aff, weights = wt)
```

```
       0    1
0 183400 2203
1   2821  203
```

Calculate the prevalence (per 1000): [1]

[1] Note that you can use **R**as a calculating machine; entering 2*2 will result in **R**printing out the answer 4.

1. using data for sib 1 in each pair,
2. using data for sib 2 in each pair, and
3. using all siblings.

You could also calculate the first two prevalences by taking the mean of the 0/1 variables coding disease status:

```
> wmean(s1aff, weights = wt)

[1] 0.01603164

> wmean(s2aff, weights = wt)

[1] 0.01275533
```

Sib's 1 and 2 seem to have different prevalences. In fact, the only difference between the two is that Sib 1 was the one ascertained first when the family was enrolled. Sib 1 is, on average, slightly older than sib 2 and this might account for their higher prevalence.

```
> wtable(s1age, weights = wt)

  0 - 9 10 - 19 20 - 29 30 - 44 45 - 74
  44262   73743   37094   19587   13941

> wtable(s2age, weights = wt)

  0 - 9 10 - 19 20 - 29 30 - 44 45 - 74
  77601   47700   28282   21027   14017
```

Recurrence risks

Taking sib 1 as the proband, the following command calculates prevalences of leprosy in "kin" (*i.e.* sib 2) by disease status of proband (sib 1):

```
> mean.table(s2aff, s1aff, weights = wt)

         0          1
0.01186942 0.06712963
```

What is the recurrence risk (where "risk" here is measured by prevalence)? Calculate the relative recurrence risk. One could also use the odds ratio in the 2×2 table obtained by crossing disease status in the two sibs as a close approximation to the sibling relative recurrence risk. You should check this using the numbers obtained by:

```
> wtable(s1aff, s2aff, weights = wt)
```

```
       0    1
0 183400 2203
1   2821  203
```

You could also calculate this odds ratio using logistic regression. Taking sibling 1 as proband:

```
> logistic(s2aff ~ s1aff, weights = wt)

Logistic regression:  s2aff ~ s1aff

Odds ratios (1 unit change), lower and upper confidence limits, and tests:

            OR    Lower    Upper   z-test       P-value
s1aff 5.990703 5.164016 6.949732 23.62925 1.929156e-123
```

A few notes:

- Odds ratios and prevalence ratios are very close to one another (the “rare disease” scenario)
- The odds ratio would be exactly the same if sib 2 were taken as proband and sib 1 as kin (you should check this is so)
- The odds ratio is, on average, invariant under case–control sampling (comparing risk in kindred of probands with kindred randomly sampled healthy controls)

Confounding

As before, taking sib 1 as proband, we can tabulate prevalence in kindred by age and disease status of proband:

```
> mean.table(s2aff, s2age, s1aff, weights = wt)

                   0          1
0 - 9   0.0008423508 0.02064220
10 - 19 0.0098589155 0.07075472
20 - 29 0.0209739721 0.06196213
30 - 44 0.0281329923 0.07625899
45 - 74 0.0390525448 0.08875740
```

Relative recurrence risks are much stronger at younger ages.

We can also use simple tabulations to look for the possibility of confounding of genetic and environmental effects. For example, BCG vaccination clusters within sibships:

```
> wtable(s1bcg, s2bcg, weights = wt)
```

```
                   negative/uncertain positive
negative/uncertain              64863    32347
positive                        44658    46759
```

The odds ratio is just over 2. To look at prevalence by proband status and by BCG vaccination status of both sibs:

```
> mean.table(s2aff, s1aff, s1bcg, s2bcg, weights = wt)

, , negative/uncertain

  negative/uncertain    positive
0         0.01857561 0.008656737
1         0.08457711 0.048192771

, , positive

  negative/uncertain    positive
0         0.00867488 0.007971656
1         0.04331450 0.051063830
```

You should be able to see that a relative recurrence risk substantially larger than one exists regardless of BCG status of the pair of sibs.

We could measure the relative recurrence risk in terms of odds ratios as before, and use logistic regression to allow for the confounding effect of BCG vaccination:

```
> logistic(s2aff ~ s1aff + s1bcg * s2bcg, weights = wt)

Logistic regression:  s2aff ~ s1aff + s1bcg * s2bcg

Odds ratios (1 unit change), lower and upper confidence limits, and tests:

                                   OR     Lower     Upper     z-test
s1aff                       5.1829316 4.4619571 6.0204030  21.530608
s1bcgpositive               0.4665043 0.4167539 0.5221936 -13.252295
s2bcgpositive               0.4650500 0.4097593 0.5278013 -11.855396
s1bcgpositive:s2bcgpositive 1.9953891 1.6526041 2.4092751   7.183763
                                  P-value
s1aff                       8.047058e-103
s1bcgpositive                4.376155e-40
s2bcgpositive                2.017683e-32
s1bcgpositive:s2bcgpositive  6.781841e-13
```

In this model, the odds of sib 2 is allowed to vary across the four categories of sibship vaccination status, but the odds ratio by proband status is assumed constant. Note that this odds ratio is now somewhat lower than the crude odds ratio we calculated earlier; this is because we have now allowed for the confounding effect of BCG vaccination.

Do not believe the tests and confidence intervals — these take no account of the fact that we only have a 5% sample of disease-free sibships and have counted each one 20 times. But the estimates for recurrence risk dolding BCG status constant should not be misleading.[2]

You might also like to look at the possible confounding effect of PB and MB household contact

Effect modification

We can also use logistic regression to check for *effect modification*. That is, the relative recurrence risk could vary as a function of BCG status of one or both siblings:

```
> logistic(s2aff ~ s1aff * s1bcg * s2bcg, weights = wt)

Logistic regression:  s2aff ~ s1aff * s1bcg * s2bcg

Odds ratios (1 unit change), lower and upper confidence limits, and tests:

                                         OR     Lower     Upper      z-test
s1aff                             4.8814061 4.0573616 5.8728129  16.8060705
s1bcgpositive                     0.4613643 0.4108457 0.5180948 -13.0739656
s2bcgpositive                     0.4623398 0.4052474 0.5274754 -11.4721233
s1aff:s1bcgpositive               1.1878397 0.7232995 1.9507314   0.6801224
s1aff:s2bcgpositive               1.0599169 0.6611413 1.6992189   0.2416497
s1bcgpositive:s2bcgpositive       1.9903667 1.6379152 2.4186599   6.9222254
s1aff:s1bcgpositive:s2bcgpositive 1.0896257 0.4977628 2.3852406   0.2147319
                                       P-value
s1aff                             2.202994e-63
s1bcgpositive                     4.638307e-39
s2bcgpositive                     1.821347e-30
s1aff:s1bcgpositive               4.964270e-01
s1aff:s2bcgpositive               8.090516e-01
s1bcgpositive:s2bcgpositive       4.446031e-12
s1aff:s1bcgpositive:s2bcgpositive 8.299764e-01
```

in this output, the parameter `s1aff:s1bcgpositive` represents the (multiplicative) effect of sibling 1 being BCG positive on the relative recurrence risk. You should again check that the results of this analysis are not affected by interchanging sibs 1 and 2.

Acknowledgement

Thanks are due to Prof. Paul Fine, coworkers, and collaborators in the Malawi leprosy study for allowing me to use these data.

[2] There are ways of obtaining correct confidence intervals in these circumstances, but we will leave these for another time.

Reference

Wallace C., Clayton D., and Fine P. (2003) “Estimating the relative recurrence risk ratio for leprosy in Karonga District, Malawi”, *Leprosy Reviews* **74**:133–40.

Exercise 2: Linkage analysis by affected relative pairs

Prior IBD probabilities

What is the “prior” probability that the doubly framed members of each of the pedigrees shown in Table 1 share 0, 1, or 2 copies of a gene identically by descent (IBD). You should assume that founders are not inbred.

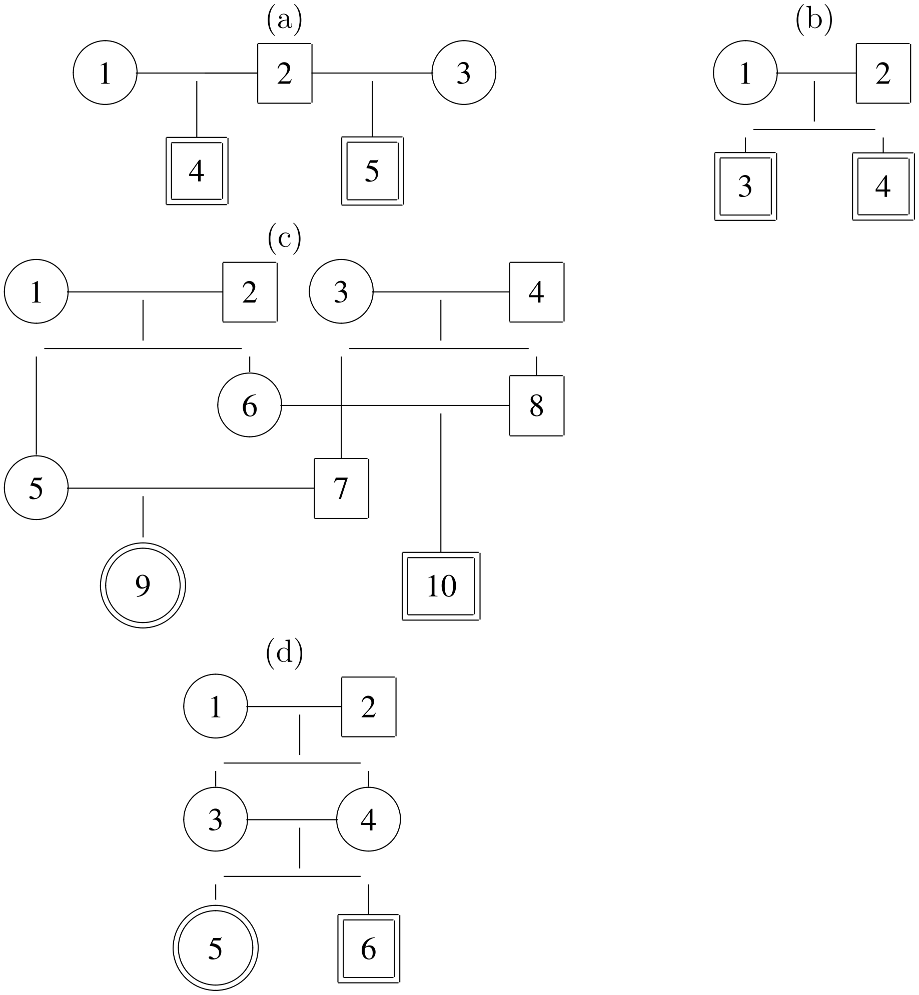


Table 1: Four pedigrees. calculate IBD probabilities for indicated relative pairs

Posterior IBD probabilities

All except one of the pedigree members shown in Table 2 have been typed at a 4–allele marker locus. The missing typing is shown as 0/0. What is the “posterior” probability that the doubly framed members of each of the following pedigrees share 0, 1, or 2 copies of the marker identically by descent (IBD). Again you should assume that founders are not inbred and you should also assume that the marker is not linked to any disease-susceptibility locus. You may need to know that the marker allele frequencies are $1 = 0.5$, $2 = 0.2$, $3 = 0.2$ and $4 = 0.1$.[3]

IBD sharing scores and NPL statistics

1. Calculate the S_{Pairs} statistic for the affected relative pairs in Table 2. Each pair contributes

$$1 \times (\text{Posterior probability 1-IBD}) + 2 \times (\text{Posterior probability 2-IBD})$$

2. Calculate its expected value under the null hypthesis (no linkage) by substituting the "prior" probabilities in the above

The z-score, sometimes called the NPL ("Non-Parametric Linkage") score, is obtained by dividing the observed minus "expected" S_{Pairs} statistic by the square root of its variance. Until recently, the variance was calculated in available programs by treating the S_{Pairs} statistic as if it was a true count — *i.e.* as if all posterior IBD assignments were certain. This gives too large a variance and, therefore, the procedure is conservative. More recent software corrects the error.

The Haseman–Elston method for quantitative traits

In this practical exercise we will carry out Haseman–Elston quantitative trait linkage analyses of a dataset concerning bone mineral density (BMD) in sibling pairs. We shall be using **R**to calculate the regressions of trait similarity on IBD status. Posterior IBD probabilities given marker data must first be calculated using a standard program such as GENEHUNTER or MERLIN, but I have done this in advance. These programs estimate IBD status at several marker loci and, possibly, at points in between. To load the data:

```
> data(bmd.sibs)
> summary(bmd.sibs)
```

```
      pos             ped              mem.1           mem.2
 Min.   : 0.0   Min.   :  1.00   Min.   :3.000   Min.   : 4.000
 1st Qu.: 0.0   1st Qu.: 22.00   1st Qu.:3.000   1st Qu.: 5.000
 Median : 7.5   Median : 49.00   Median :4.000   Median : 6.000
 Mean   :10.6   Mean   : 55.26   Mean   :4.143   Mean   : 6.258
 3rd Qu.:24.3   3rd Qu.: 86.00   3rd Qu.:5.000   3rd Qu.: 8.000
 Max.   :24.3   Max.   :118.00   Max.   :9.000   Max.   :10.000

     sex.1           sex.2             z0               z1
```

[3] Hint for pedigree (f): The mother must be either homozygous (1/1) or heterozygous (1/x). Work out the probability of the pedigree under each of these possibilities by first calculating the probabilities of the founder genotypes, then multiplying by the probability of descendant genotypes conditional on founders. These two numbers give the relative weight we should give to the two possibilities when calculating the posterior IBD sharing probabilities.

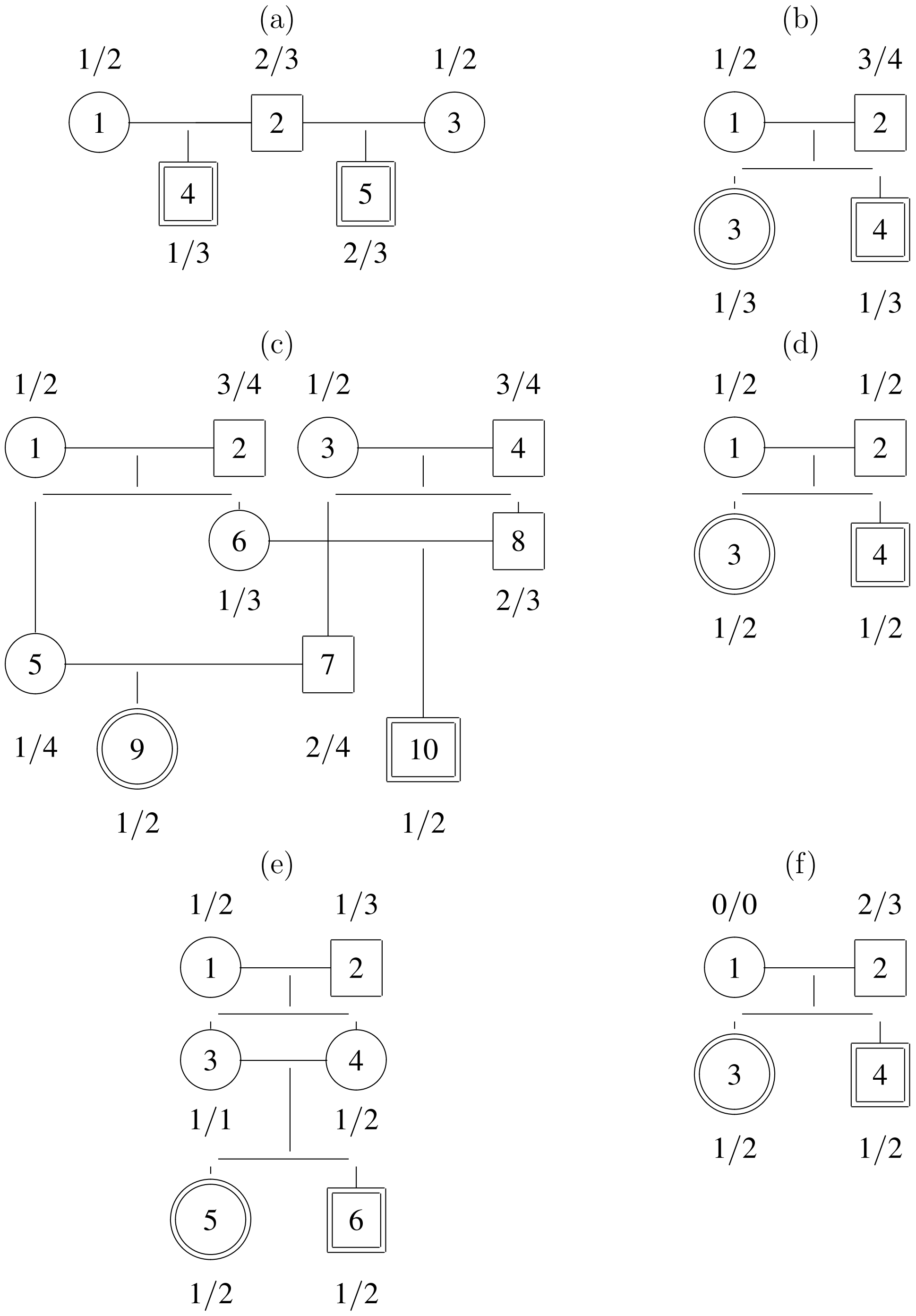


Table 2: Marker genotypes in six pedigrees; calculate posterior IBD probabilities for affected relative pairs

```
Min.   :1.000   Min.   :1.000   Min.   :0.00000   Min.   :0.0000
1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.00000   1st Qu.:0.2067
Median :2.000   Median :2.000   Median :0.07946   Median :0.5270
Mean   :1.519   Mean   :1.508   Mean   :0.24274   Mean   :0.5168
3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:0.35297   3rd Qu.:0.8427
Max.   :2.000   Max.   :2.000   Max.   :1.00000   Max.   :1.0000

      z2             bmd.1             bmd.2
Min.   :0.0000   Min.   : 0.5540   Min.   :0.5440
1st Qu.:0.0000   1st Qu.: 0.7920   1st Qu.:0.7640
Median :0.0000   Median : 0.8825   Median :0.8650
Mean   :0.2404   Mean   : 0.8875   Mean   :0.8632
3rd Qu.:0.4353   3rd Qu.: 0.9700   3rd Qu.:0.9630
Max.   :1.0000   Max.   : 1.3370   Max.   :1.3340
                 NA's   :39.0000   NA's   :9.0000
```

This datset concerns IBD status at three loci. We shall only look at the VNTR marker locus, which is at position 7.5. First extract this subset of the data and attach it:

```
> at.vntr <- subset(bmd.sibs, pos == 7.5)
> attach(at.vntr)
```

The first step is to calculate the squared differences between BMD values for eachpair:

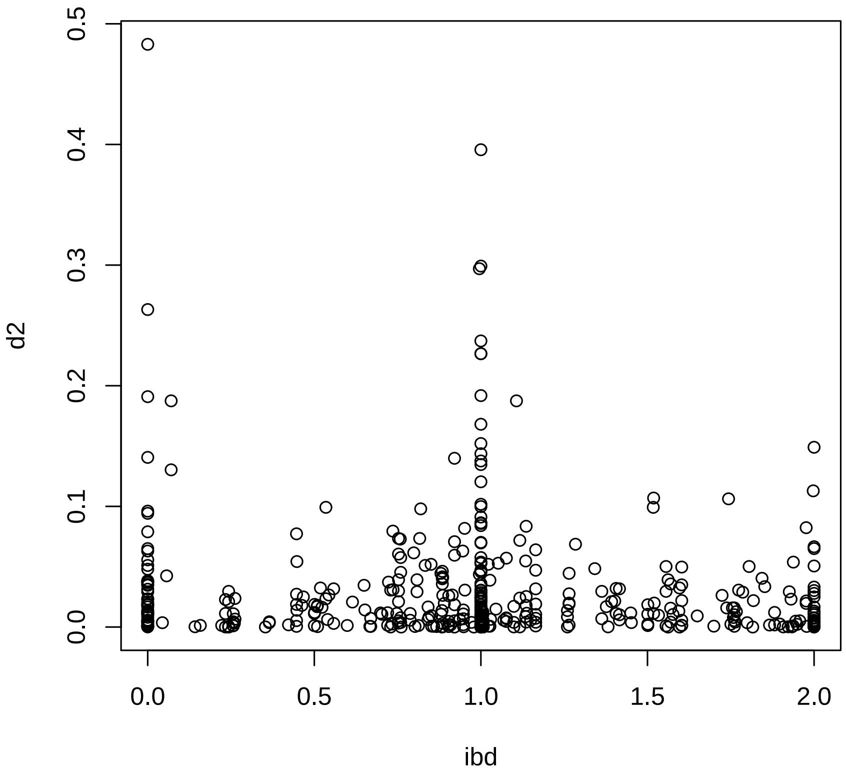
```
> d2 <- (bmd.1 - bmd.2)^2
```

The next step is to calculate the expected number of copies shared by each sib pair *a posteriori*:

```
> ibd <- 2 * z2 + z1
```

The Haseman-Elston method looks for correlation beyween these two quantities:

```
> plot(ibd, d2)
```



```
> cor.test(ibd, d2)

        Pearson's product-moment correlation

data:  ibd and d2
t = -2.2994, df = 431, p-value = 0.02196
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.20222662 -0.01601595
sample estimates:
       cor
-0.1100872
```

On the face of it, there is significant linkage at this locus, albeit falling a long way short of "genome-wide" significance. However, squaring differences can lead to very influential outliers, as is demonstrated in the plot. It would be better to use a rank correlation test:

```
> cor.test(ibd, d2, method = "spearman")

        Spearman's rank correlation rho
```

```
data:  ibd and d2
S = 14683858, p-value = 0.07639
alternative hypothesis: true rho is not equal to 0
sample estimates:
        rho
-0.08525066
```

The “significance” is lost! An additional problem with this analysis is that it uses all sib-pair comparisons for sibships of size > 2. These do not represent independent data points. The analysis can be corrected to allow for this correlation.

Haseman–Elston revisited

Haseman and Elston later suggested a modification of their methodwhich uses a different measure of trait similarity — namely the product of deviations of trait values from their mean. First we must calculate the mean value:

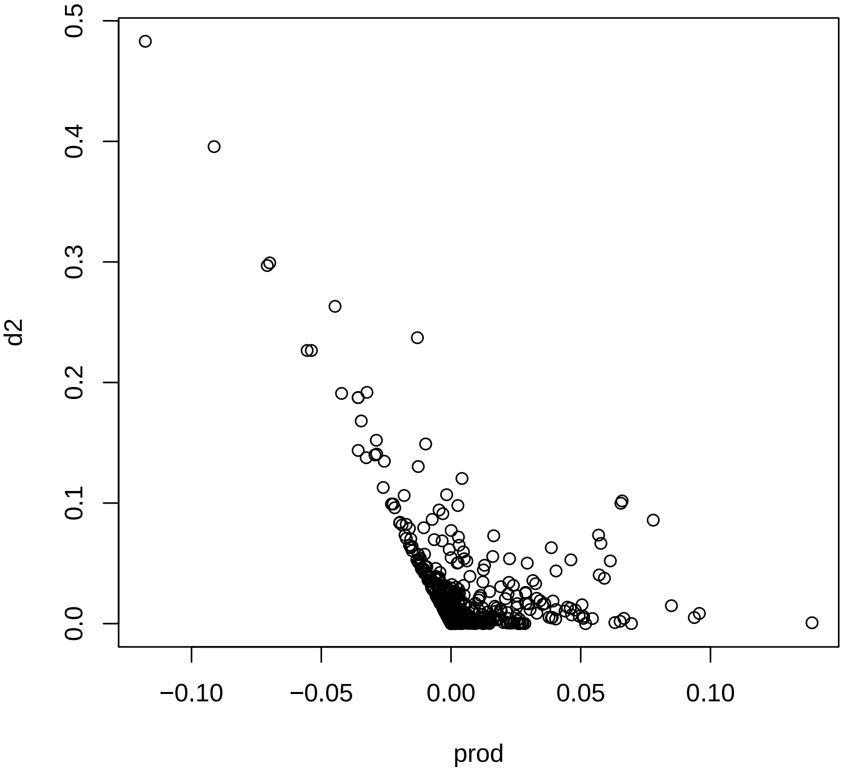
```
> bmd.m <- mean(c(bmd.1, bmd.2), na.rm = TRUE)
```

(The argument `na.rm=TRUE` is an annoying feature of **R**. Without it, any missing values in the argument(s) results in the mean also being regarded as missing — a somewhat pedantic form of behaviour). We now calculate the products of differences from the mean:

```
> prod <- (bmd.1 - bmd.m) * (bmd.2 - bmd.m)
```

To see how these relate to the squared distances, try

```
> plot(prod, d2)
```



Why do you think that it might be better to use products rather than squared differences? Using products, you can carry out the Haseman–Elston test (or the non-parametric alternative) as before:

```
> cor.test(ibd, prod)

        Pearson's product-moment correlation

data:  ibd and prod
t = 1.5032, df = 431, p-value = 0.1335
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.02217234  0.16532821
sample estimates:
       cor
0.07221595

> cor.test(ibd, prod, method = "spearman")

        Spearman's rank correlation rho

data:  ibd and prod
S = 12637565, p-value = 0.1705
```

```
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
0.06598621
```

The two Haseman-Elston methods turn out to use rather different parts of the information and it has been shown that the optimal test is a weighted compromise between the two methods. Special purpose software is available for this.

Exercise 3: Linkage disequilibrium

Susan Service wrote the first version of this exercise for the MSc course at Erasmus University, Rotterdam. David Clayton made minor alterations and converted it to LaTeX and **R**. Thanks to the Diabetes and Inflammation laboratory for use of the insulin dataset.

Calculating D'

The table below shows the frequency of haplotypes of two loci in 300 unrelated subjects (although, in practice, you would not be able to observe these frequencies directly). Calculate D' between the two loci. What is the assumed order in which these alleles arose in the population history?

Locus 1	Locus 2 allele		
allele	1	2	Total
1	172	41	213
2	67	320	387
Total	239	361	600

Examining LD in different populations

Data for this exercise are taken from a paper by Gabriel et al. (2002). The data come from a 200 kb region on chromosome 2[4]. In the paper it is noted that populations of African descent have less extensive LD than populations of European or Asian ancestry. For this exercise we will use genotype data for 14 SNP markers from 50 African Americans and 42 Asians.

In the exercise below, we shall use **R**functions from the `genetics` package `pwld` which calculate various measures of LD, after first resolving phase using the EM algorithm.

Start **R**and get the data:

```
> data(Gabriel.etal)
> summary(AfAm)

      id            s641676         s495166         s586708
 Length:50          1/1 :12 (24%)   1/1 :33 (66%)   1/1: 1 ( 2%)
 Class :character   2/1 :18 (36%)   1/2 :13 (26%)   2/1:13 (26%)
 Mode  :character   2/2 :18 (36%)   2/2 : 1 ( 2%)   2/2:36 (72%)
                    NA's: 2 ( 4%)   NA's: 3 ( 6%)
 s641662        s785900         s266619        s1077797       s474197
 1/1: 5 (10%)   1/1 : 1 ( 2%)   1/1: 7 (14%)   1/1:36 (72%)   1/1 : 3 ( 6%)
 2/1:18 (36%)   2/1 :17 (34%)   2/1:23 (46%)   1/2:13 (26%)   2/1 :16 (32%)
 2/2:27 (54%)   2/2 :30 (60%)   2/2:20 (40%)   2/2: 1 ( 2%)   2/2 :24 (48%)
                NA's: 2 ( 4%)                                 NA's: 7 (14%)
 s474196        s376142         s785897          s51791         s785879
```

[4] They can be found at: `http://www-genome.wi.mit.edu/mpg/hapmap/hapstruc.html#data`

```
 1/1:27 (54%)   1/1 :29 (58%)   1/1 : 2 ( 4%)   1/1 :23 (46%)   1/1 : 2 ( 4%)
 1/2:18 (36%)   1/2 :17 (34%)   2/1 :13 (26%)   1/2 :23 (46%)   2/1 :15 (30%)
 2/2: 5 (10%)   2/2 : 2 ( 4%)   2/2 :33 (66%)   2/2 : 3 ( 6%)   2/2 :29 (58%)
                NA's: 2 ( 4%)   NA's: 2 ( 4%)   NA's: 1 ( 2%)   NA's: 4 ( 8%)
 s462060
 1/1 : 2 ( 4%)
 2/1 :16 (32%)
 2/2 :29 (58%)
 NA's: 3 ( 6%)
```

```
> summary(Asian)
```

```
      id             s641676          s495166          s586708
 Length:42          1/1 :12 (29%)   1/1 : 6 (14%)   1/1 :13 (31%)
 Class :character   1/2 :21 (50%)   2/1 :20 (48%)   1/2 :22 (52%)
 Mode  :character   2/2 : 7 (17%)   2/2 :13 (31%)   2/2 : 6 (14%)
                    NA's: 2 ( 5%)   NA's: 3 ( 7%)   NA's: 1 ( 2%)
 s641662         s785900          s266619          s1077797
 1/1 : 1 ( 2%)   2/1 :11 (26%)   1/1 :13 (31%)   1/1 : 8 (19%)
 2/1 :14 (33%)   2/2 :30 (71%)   1/2 :22 (52%)   2/1 :20 (48%)
 2/2 :26 (62%)   NA's: 1 ( 2%)   2/2 : 6 (14%)   2/2 :13 (31%)
 NA's: 1 ( 2%)                   NA's: 1 ( 2%)   NA's: 1 ( 2%)
 s474197         s474196          s376142          s785897          s51791
 2/1 :11 (26%)   1/1 :26 (62%)   1/1 :24 (57%)   1/1 : 2 ( 5%)   1/1:18 (43%)
 2/2 :25 (60%)   1/2 :14 (33%)   1/2 :11 (26%)   2/1 :15 (36%)   1/2:21 (50%)
 NA's: 6 (14%)   2/2 : 1 ( 2%)   2/2 : 2 ( 5%)   2/2 :23 (55%)   2/2: 3 ( 7%)
                 NA's: 1 ( 2%)   NA's: 5 (12%)   NA's: 2 ( 5%)
 s785879        s462060
 1/1: 6 (14%)   1/1: 8 (19%)
 2/1:18 (43%)   2/1:15 (36%)
 2/2:18 (43%)   2/2:19 (45%)
```

```
> summary(positions)
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
86290000 86350000 86360000 86370000 86380000 86470000
```

```
> summary(distances)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    338   10610   28770   47810   85410  176900
```

There are 14 markers in each dataset so that, with 14 loci there are a total of 91 pairwise measures of LD to calculate. The variable `positions` holds the marker locations (in base pairs) on chromosome 2 and `distances` holds the inter-marker spacings. To calculate several measures of LD, we use the function `LD()`. For the African Americans:

```
> ldAfAm <- LD(AfAm)
```

To list D' values:

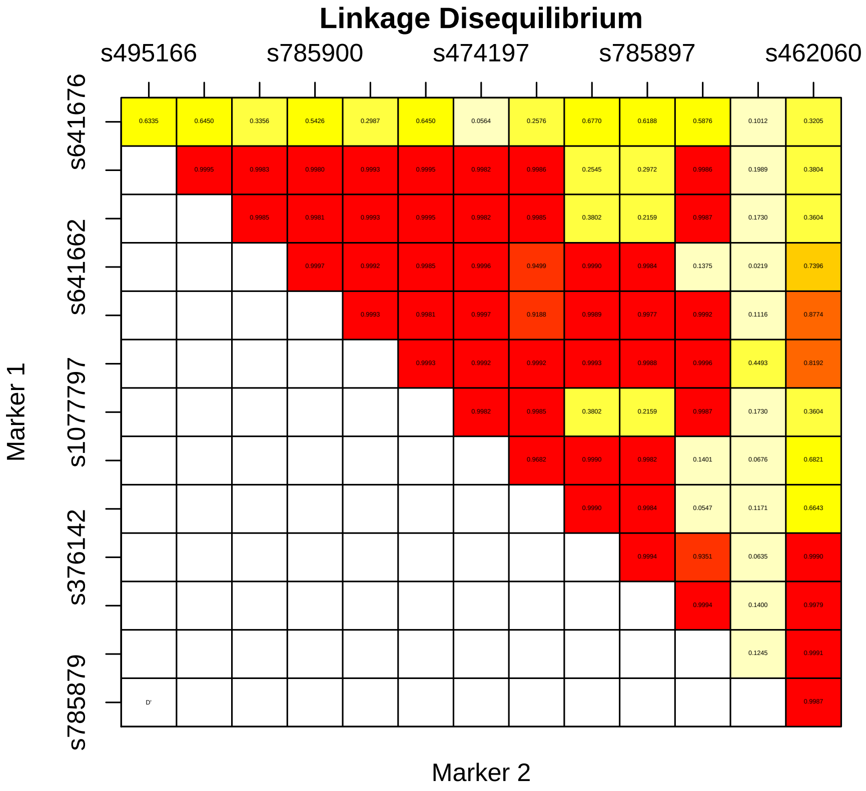
```
> summary(ldAfAm, which = "D'")
```

```
Pairwise LD
-----------
               s495166   s586708   s641662   s785900   s266619  s1077797
s641676  D' 0.6335219 0.6450313 0.3355714 0.5426163 0.2987093 0.6450313
s495166  D'           0.9994609 0.9982947 0.9979989 0.9992808 0.9994609
s586708  D'                     0.9985480 0.9980888 0.9992808 0.9994669
s641662  D'                               0.9996563 0.9992432 0.9985480
s785900  D'                                         0.9993311 0.9980888
s266619  D'                                                   0.9992808
s1077797 D'
s474197  D'
s474196  D'
s376142  D'
s785897  D'
s51791   D'
s785879  D'
               s474197   s474196   s376142   s785897    s51791   s785879
s641676  D' 0.0564148 0.2576432 0.6770152 0.6187540 0.5875759 0.1011929
s495166  D' 0.9982288 0.9986317 0.2544834 0.2971593 0.9985684 0.1989184
s586708  D' 0.9982288 0.9985480 0.3802254 0.2159466 0.9986805 0.1729565
s641662  D' 0.9996111 0.9499408 0.9990462 0.9983818 0.1374880 0.0218593
s785900  D' 0.9996675 0.9187777 0.9988687 0.9977106 0.9991637 0.1115945
s266619  D' 0.9992432 0.9992432 0.9992782 0.9987754 0.9996148 0.4493375
s1077797 D' 0.9982288 0.9985480 0.3802254 0.2159466 0.9986805 0.1729565
s474197  D'           0.9682083 0.9989561 0.9982288 0.1401305 0.0676433
s474196  D'                     0.9990462 0.9983818 0.0547397 0.1170825
s376142  D'                               0.9994200 0.9351441 0.0635456
s785897  D'                                         0.9993565 0.1399910
s51791   D'                                                   0.1244568
s785879  D'
               s462060
s641676  D' 0.3205039
s495166  D' 0.3803745
s586708  D' 0.3604253
s641662  D' 0.7396193
s785900  D' 0.8773779
s266619  D' 0.8191677
s1077797 D' 0.3604253
s474197  D' 0.6820881
s474196  D' 0.6642623
s376142  D' 0.9990037
```

```
s785897  D' 0.9978704
s51791   D' 0.9991152
s785879  D' 0.9986524
```

You can also see a matrix of correlation coefficients values by giving `r` instead of `D'` in the above command. Other (less useful) measures are available. A useful graphical display is a colour coded table:

```
> LDtable(ldAfAm, which = "D'")
```



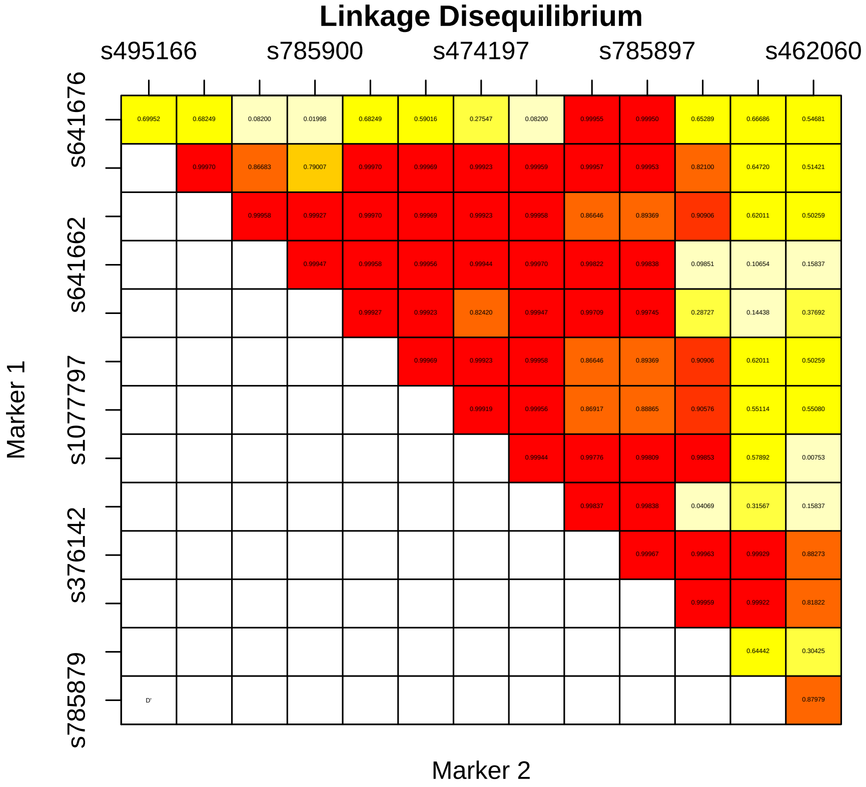
This shows D' by default but can show other indices.

Let's see how the LD pattern looks for the Asian sample.

```
> ldAsian <- LD(Asian)
> summary(ldAsian, which = "D'")
```

```
Pairwise LD
-----------
                s495166   s586708   s641662   s785900   s266619  s1077797
s641676  D' 0.6995223 0.6824916 0.0820022 0.0199831 0.6824916 0.5901584
s495166  D'           0.9997043 0.8668312 0.7900676 0.9997043 0.9996915
s586708  D'                     0.9995823 0.9992659 0.9997043 0.9996915
s641662  D'                               0.9994661 0.9995823 0.9995641
```

```
s785900  D'                                                   0.9992659 0.9992339
s266619  D'                                                             0.9996915
s1077797 D'
s474197  D'
s474196  D'
s376142  D'
s785897  D'
s51791   D'
s785879  D'
              s474197   s474196   s376142   s785897    s51791   s785879
s641676  D' 0.2754686 0.0820022 0.9995514 0.9995022 0.6528869 0.6668566
s495166  D' 0.9992317 0.9995854 0.9995721 0.9995252 0.8209953 0.6472038
s586708  D' 0.9992259 0.9995823 0.8664553 0.8936853 0.9090600 0.6201138
s641662  D' 0.9994371 0.9996962 0.9982248 0.9983823 0.0985087 0.1065445
s785900  D' 0.8241965 0.9994661 0.9970929 0.9974529 0.2872698 0.1443847
s266619  D' 0.9992259 0.9995823 0.8664553 0.8936853 0.9090600 0.6201138
s1077797 D' 0.9991923 0.9995641 0.8691663 0.8886525 0.9057579 0.5511399
s474197  D'           0.9994371 0.9977647 0.9980922 0.9985274 0.5789169
s474196  D'                     0.9983749 0.9983823 0.0406933 0.3156654
s376142  D'                               0.9996691 0.9996282 0.9992935
s785897  D'                                         0.9995873 0.9992159
s51791   D'                                                   0.6444150
s785879  D'
              s462060
s641676  D' 0.5468145
s495166  D' 0.5142088
s586708  D' 0.5025932
s641662  D' 0.1583654
s785900  D' 0.3769195
s266619  D' 0.5025932
s1077797 D' 0.5507966
s474197  D' 0.0075276
s474196  D' 0.1583654
s376142  D' 0.8827266
s785897  D' 0.8182224
s51791   D' 0.3042517
s785879  D' 0.8797896

> LDtable(ldAsian, which = "D'")
```



Does it look like there is more or less LD than in the African American sample? It's hard to see from these graphs. To look more carefully, we'll capture the D' values:

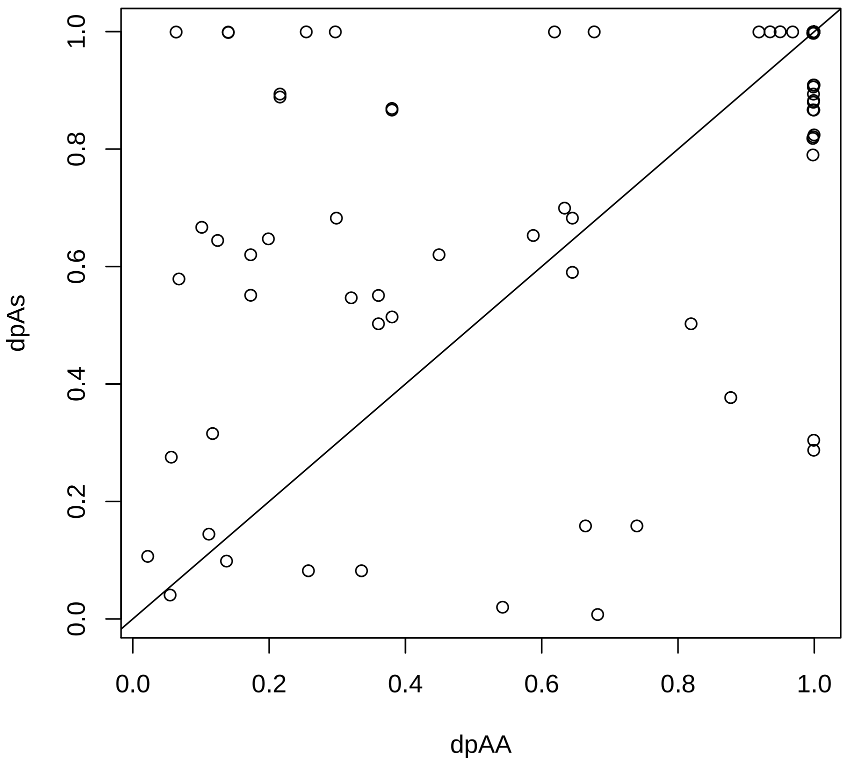
```
> dpAA <- summary(ldAfAm, which = "D'")
> dpAs <- summary(ldAsian, which = "D'")
```

If you look at the contents of these arrays, you will see that only half the matrix has values — the rest are missing (`NA`). To get rid of these:

```
> dpAA <- dpAA[!is.na(dpAA)]
> dpAs <- dpAs[!is.na(dpAs)]
```

We'll now create a scatter plot with a line of equality:

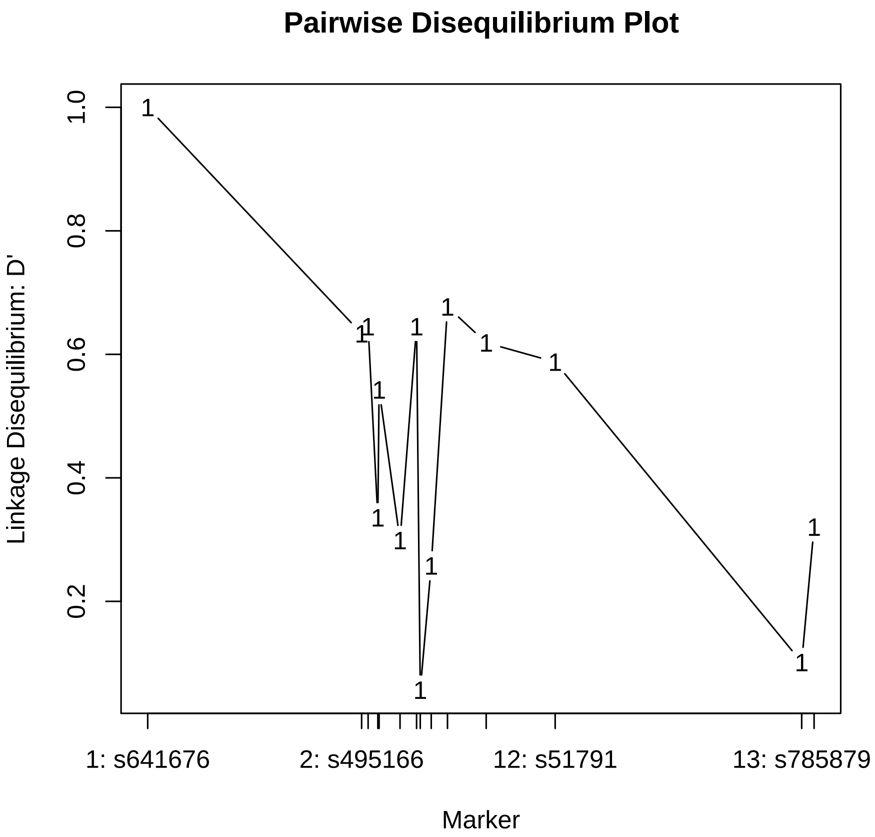
```
> plot(dpAA, dpAs)
> abline(0, 1)
```



Is there a suggestion that more or less points lie above the line? What is the interpretation of this, and is this what you would have expected to find? It is interesting to note that the correlation in this plot is not strong — pairs of markers taht are in strong LD in one population are not necessarily in strong LD in the other.

To plot the values of D' against the inter-marker distances, the `genetics` package has the function `LDplot`. To plot D' values against distances from the first marker, and then from the second marker and so on:

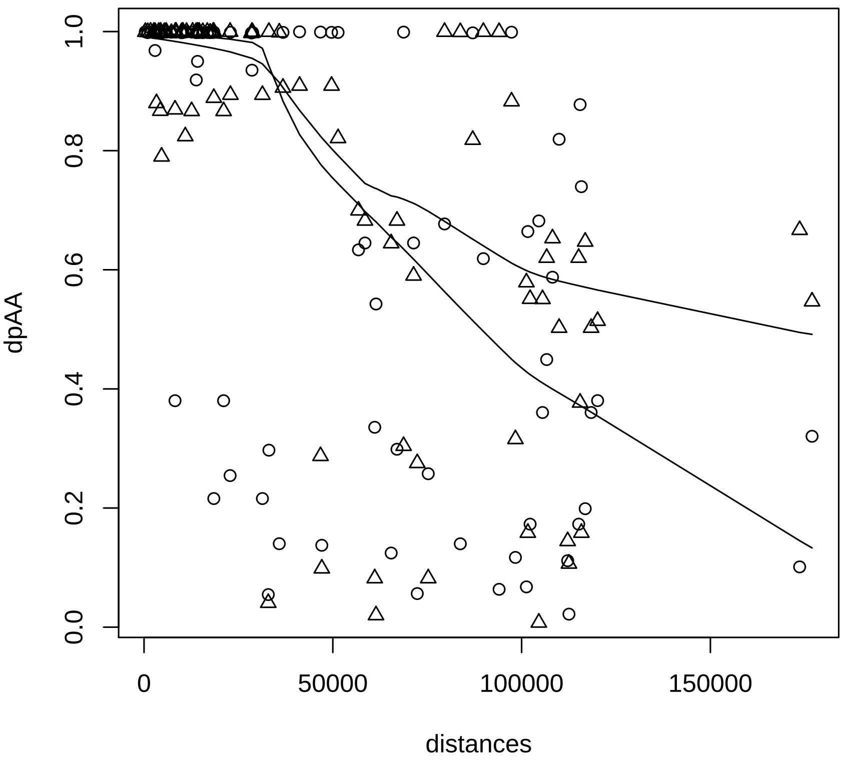
```
> LDplot(ldAfAm, distance = positions, which = "D'", marker = 1)
> LDplot(ldAfAm, distance = positions, which = "D'", marker = 2)
> LDplot(ldAfAm, distance = positions, which = "D'", marker = 3)
```



and so on. It can be seen that, whereas D' does fall away with distance, it does so in a highly variable manner. Although this is partly due to the small sample size, it is mainly due to the random nature of the recombination history. You should try running these commands for the Asian data also. Is there any suggestion that the extent of LD differs between these populations?

By dropping the `marker=` argument in the above commands you will get all plots superimposed. You can try it but almost certainly you won't find it very useful. Instead let's plot D' values against the inter-marker distances, using different plotting symbols. This is still not very clear, so we will superimpose *lowess* smoothed lines:

```
> plot(distances, dpAA)
> points(distances, dpAs, pch = 2)
> lines(lowess(distances, dpAA))
> lines(lowess(distances, dpAs))
```



The difference between the populations is now apparent! The "half-life" of LD has been defined as the distance at which the mean level of D' drops below 0.5. What is the approximate half-life of LD in these two populations?

Examining many loci in one population

Now we'll look at the pattern of LD in a 70kb region around the insulin gene. Approximately 70 SNP markers are typed in 868 subjects (not all subjects have data for all SNPs). First fetch the data:

```
> data(insulin)
> summary(insulin)
```

```
    familyid          member      father          mother              sex
 Min.   :  1.0   Min.   :1.0   Mode:logical   Mode:logical   Min.   :1.0
 1st Qu.:123.0   1st Qu.:1.0   NA's:868       NA's:868       1st Qu.:1.0
 Median :254.5   Median :1.5                                 Median :1.5
 Mean   :253.7   Mean   :1.5                                 Mean   :1.5
 3rd Qu.:381.0   3rd Qu.:2.0                                 3rd Qu.:2.0
 Max.   :523.0   Max.   :2.0                                 Max.   :2.0

      t1d            l1                l2                l3
 Min.   :1.000   1/1 :572 (66%)   1/1 :191 (22%)   1/1 :182 (21%)
```

```
1st Qu.:1.000   1/2 :240 (28%)   1/2 :397 (46%)   2/1 :408 (47%)
Median :1.000   2/2 : 21 ( 2%)   2/2 :174 (20%)   2/2 :182 (21%)
Mean   :1.063   NA's: 35 ( 4%)   NA's:106 (12%)   NA's: 96 (11%)
3rd Qu.:1.000
Max.   :2.000

   l4               l5               l6               l7
1/1 : 33 ( 4%)   1/1 :362 (42%)   1/1 :191 (22%)   1/1 :142 (16%)
2/1 :291 (34%)   1/2 :335 (39%)   2/1 :444 (51%)   1/2 :331 (38%)
2/2 :494 (57%)   2/2 : 87 (10%)   2/2 :191 (22%)   2/2 :136 (16%)
NA's: 50 ( 6%)   NA's: 84 (10%)   NA's: 42 ( 5%)   NA's:259 (30%)


   l8               l9              l10              l11
1/1 :  7 ( 1%)   1/1 :365 (42%)   1/1 :152 (18%)   1/1 :632 (73%)
2/1 :167 (19%)   1/2 :380 (44%)   2/1 :404 (47%)   1/2 :159 (18%)
2/2 :667 (77%)   2/2 : 73 ( 8%)   2/2 :248 (29%)   2/2 : 10 ( 1%)
NA's: 27 ( 3%)   NA's: 50 ( 6%)   NA's: 64 ( 7%)   NA's: 67 ( 8%)


  l12              l13              l14              l15
1/1 :180 (21%)   1/1 : 23 ( 3%)   1/1 :251 (29%)   1/1 : 37 ( 4%)
2/1 :433 (50%)   2/1 :222 (26%)   1/2 :399 (46%)   2/1 :248 (29%)
2/2 :187 (22%)   2/2 :571 (66%)   2/2 :144 (17%)   2/2 :523 (60%)
NA's: 68 ( 8%)   NA's: 52 ( 6%)   NA's: 74 ( 9%)   NA's: 60 ( 7%)


  l16              l17              l18              l19
1/1 :460 (53%)   1/1 : 28 ( 3%)   1/1 :459 (53%)   1/1 :461 (53%)
1/2 :179 (21%)   2/1 :175 (20%)   1/2 :176 (20%)   1/2 :179 (21%)
2/2 : 30 ( 3%)   2/2 :459 (53%)   2/2 : 28 ( 3%)   2/2 : 15 ( 2%)
NA's:199 (23%)   NA's:206 (24%)   NA's:205 (24%)   NA's:213 (25%)


  l20              l21              l22              l23
1/1 :268 (31%)   1/1 : 96 (11%)   1/1 :277 (32%)   1/1 :285 (33%)
1/2 :284 (33%)   2/1 :293 (34%)   1/2 :264 (30%)   1/2 :280 (32%)
2/2 : 94 (11%)   2/2 :260 (30%)   2/2 : 76 ( 9%)   2/2 : 74 ( 9%)
NA's:222 (26%)   NA's:219 (25%)   NA's:251 (29%)   NA's:229 (26%)
```

```
  l24                 l25              l26              l27
1/1 : 15 ( 2%)   122/106:111 (13%)    1/1 :118 (14%)   1/1 :139 (16%)
2/1 :177 (20%)   122/110: 85 (10%)    2/1 :307 (35%)   2/1 :305 (35%)
2/2 :473 (54%)   122/118: 72 ( 8%)    2/2 :222 (26%)   2/2 :157 (18%)
NA's:203 (23%)   106/110: 71 ( 8%)    NA's:221 (25%)   NA's:267 (31%)
                 122/122: 59 ( 7%)
                 (Other):334 (38%)
                 NA's   :136 (16%)
  l28              l29              l30               l31
1/1 :564 (65%)   1/1 :  1 ( 0%)   1/1 :654 (75%)    1/1 :432 (50%)
1/2 :101 (12%)   2/1 : 19 ( 2%)   1/2 : 17 ( 2%)    1/2 :278 (32%)
2/2 :  7 ( 1%)   2/2 :642 (74%)   NA's:197 (23%)    2/2 : 30 ( 3%)
NA's:196 (23%)   NA's:206 (24%)                     NA's:128 (15%)



  l32              l33              l34               l35
1/1 :525 (60%)   1/1 :536 (62%)   1/1 :523 (60%)    1/1 :618 (71%)
1/2 :196 (23%)   1/2 :276 (32%)   1/2 :200 (23%)    1/2 : 62 ( 7%)
2/2 : 14 ( 2%)   2/2 : 32 ( 4%)   2/2 : 14 ( 2%)    NA's:188 (22%)
NA's:133 (15%)   NA's: 24 ( 3%)   NA's:131 (15%)



  l36              l37              l38               l39
1/1 :541 (62%)   1/1 : 21 ( 2%)   1/1 :480 (55%)    1/1 :457 (53%)
1/2 :201 (23%)   2/1 :233 (27%)   1/2 :177 (20%)    1/2 :178 (21%)
2/2 : 15 ( 2%)   2/2 :407 (47%)   2/2 : 11 ( 1%)    2/2 :  9 ( 1%)
NA's:111 (13%)   NA's:207 (24%)   NA's:200 (23%)    NA's:224 (26%)



  l40              l41              l42               l43
1/1 :614 (71%)   2/1 : 56 ( 6%)   1/1 : 45 ( 5%)    1/1 :641 (74%)
1/2 :228 (26%)   2/2 :636 (73%)   2/1 :256 (29%)    1/2 : 40 ( 5%)
2/2 : 19 ( 2%)   NA's:176 (20%)   2/2 :376 (43%)    2/2 :  1 ( 0%)
NA's:  7 ( 1%)                    NA's:191 (22%)    NA's:186 (21%)



  l44              l45              l46               l47
1/1 : 10 ( 1%)   1/1 :141 (16%)   1/1 :227 (26%)    1/1 :  1 ( 0%)
2/1 :145 (17%)   2/1 :366 (42%)   1/2 :282 (32%)    2/1 : 46 ( 5%)
2/2 :510 (59%)   2/2 :296 (34%)   2/2 :106 (12%)    2/2 :617 (71%)
NA's:203 (23%)   NA's: 65 ( 7%)   NA's:253 (29%)    NA's:204 (24%)
```

```
  l148             l149             l150             l151       
 1/1 :622 (72%)   1/1 :  2 ( 0%)   1/1 :289 (33%)   1/1 :108 (12%)  
 1/2 : 54 ( 6%)   2/1 : 29 ( 3%)   1/2 :306 (35%)   1/2 :256 (29%)  
 NA's:192 (22%)   2/2 :607 (70%)   2/2 : 78 ( 9%)   2/2 :103 (12%)  
                  NA's:230 (26%)   NA's:195 (22%)   NA's:401 (46%)  



  l152             l153             l154             l155       
 1/1 :140 (16%)   1/1 : 99 (11%)   2/1 :  5 ( 1%)   1/1 :575 (66%)  
 1/2 :315 (36%)   2/1 :301 (35%)   2/2 :680 (78%)   1/2 :101 (12%)  
 2/2 :126 (15%)   2/2 :273 (31%)   NA's:183 (21%)   2/2 :  3 ( 0%)  
 NA's:287 (33%)   NA's:195 (22%)                    NA's:189 (22%)  



  l156             l157             l158             l159       
 1/1 :567 (65%)   1/1 : 43 ( 5%)   2/1 : 28 ( 3%)   1/1 :571 (66%)  
 1/2 :101 (12%)   2/1 :271 (31%)   2/2 :651 (75%)   1/2 :104 (12%)  
 2/2 :  4 ( 0%)   2/2 :361 (42%)   NA's:189 (22%)   2/2 :  6 ( 1%)  
 NA's:196 (23%)   NA's:193 (22%)                    NA's:187 (22%)  



  l160             l161             l162             l163       
 1/1 : 42 ( 5%)   1/1 :  6 ( 1%)   1/1 : 17 ( 2%)   1/1 : 13 ( 1%)  
 2/1 :287 (33%)   2/1 : 92 (11%)   2/1 :112 (13%)   2/1 :102 (12%)  
 2/2 :344 (40%)   2/2 :548 (63%)   2/2 :200 (23%)   2/2 :206 (24%)  
 NA's:195 (22%)   NA's:222 (26%)   NA's:539 (62%)   NA's:547 (63%)  



  l164             l165             l166             l167       
 1/1 : 27 ( 3%)   1/1 :152 (18%)   1/1 :178 (21%)   1/1 :593 (68%)  
 2/1 :132 (15%)   2/1 :306 (35%)   1/2 :325 (37%)   1/2 : 51 ( 6%)  
 2/2 :160 (18%)   2/2 :164 (19%)   2/2 :163 (19%)   2/2 :  1 ( 0%)  
 NA's:549 (63%)   NA's:246 (28%)   NA's:202 (23%)   NA's:223 (26%)  



  l168             l169             l170             l171       
 1/1 :171 (20%)   1/1 :170 (20%)   1/1 :182 (21%)   1/1 :168 (19%)  
 1/2 :314 (36%)   1/2 :318 (37%)   1/2 :319 (37%)   2/1 :332 (38%)  
 2/2 :158 (18%)   2/2 :162 (19%)   2/2 :159 (18%)   2/2 :172 (20%)  
 NA's:225 (26%)   NA's:218 (25%)   NA's:208 (24%)   NA's:196 (23%)
```

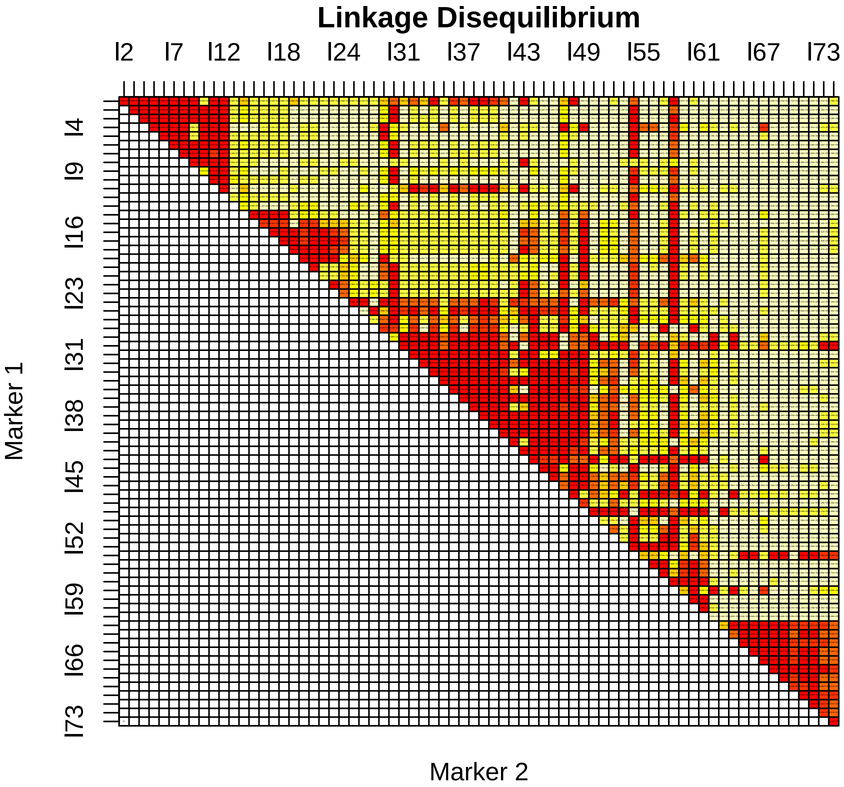
```
   l72              l73              l74
 1/1 :168 (19%)   1/1 : 50 ( 6%)   1/1 : 49 ( 6%)
 2/1 :329 (38%)   2/1 :161 (19%)   2/1 :158 (18%)
 2/2 :170 (20%)   2/2 :106 (12%)   2/2 :108 (12%)
 NA's:201 (23%)   NA's:551 (63%)   NA's:553 (64%)
```

Now calculate the LD measures (this could take quite a long time unless your computer is very fast):

```
> ldins <- LD(insulin)
```

Finally we'll plot the colour-coded D' table:

```
> LDtable(ldins, which = "D'")
```



The plot shows distinct regions where many markers are nearly in complete LD, separated by regions of low LD. there is a suggestion that there are five LD (or haplotype) "blocks", although the middle three have some degree of LD between them.

References

Gabriel et al. (2002) The Structure of Haplotype Blocks in the Human Genome. *Science*, **296**:2225–2229.

Exercise 4: Population–based association studies

The data

The data for this practical exercise concern a population–based case–control study of the association between a disease and four closely linked single nucleotide polymorphisms (SNPs). The data are in the GE03.2005 package as an **R**dataframe (`popn`). You can load and print a brief summary of the dataframe contents as follows:

```
> data(popn)
> summary(popn)

     sex          affected        A                 B                 C
 Male  : 383   Control:864   1/1 :541 (35%)   1/1 :531 (35%)   1/1 :507 (33%)
 Female:1037   Case   :672   1/2 :704 (46%)   1/2 :734 (48%)   1/2 :696 (45%)
 NA's  : 116                 2/2 :244 (16%)   2/2 :234 (15%)   2/2 :278 (18%)
                             NA's: 47 ( 3%)   NA's: 37 ( 2%)   NA's: 55 ( 4%)


    D               subject
 1/1 :296 (19%)   Min.   :   1.0
 2/1 :691 (45%)   1st Qu.: 384.8
 2/2 :454 (30%)   Median : 768.5
 NA's: 95 ( 6%)   Mean   : 768.5
                  3rd Qu.:1152.2
                  Max.   :1536.0
```

Each of the loci *A*, *B*, *C* and *D* are held as *genotype* variables. The file also contains variables coding sex and case/control status. We'll first attach the dataframe and do some simple tabulations:

```
> attach(popn)
> table(affected)

affected
Control    Case
    864     672

> table(sex)

sex
  Male Female
   383   1037
```

The following command does a chi-squared test for association between disease status and sex

```
> chisq.test(sex, affected)
```

```
        Pearson's Chi-squared test with Yates' continuity correction

data:  sex and affected
X-squared = 80.1396, df = 1, p-value < 2.2e-16
```

There is a highly significant association between disease and sex (the Pearson chi-squared test is the "score" test for association). Alternatively, we can create a table of disease status by sex and apply the `chisq.test()` function to the table:

```
> sbyd <- table(sex, affected)
> sbyd

        affected
sex      Control Case
  Male       277  106
  Female     471  566

> chisq.test(sbyd)

        Pearson's Chi-squared test with Yates' continuity correction

data:  sbyd
X-squared = 80.1396, df = 1, p-value < 2.2e-16
```

Genotype counting and allele counting

There are two simple ways of testing for association between disease and a genetic variant. The first is to simply count genotypes in cases and controls and compare them using a chi-squared test. For marker *A*,

```
> abyd <- table(A, affected)
> abyd

     affected
A      Control Case
  1/1      261  280
  1/2      406  298
  2/2      161   83

> chisq.test(abyd)

        Pearson's Chi-squared test

data:  abyd
X-squared = 23.7385, df = 2, p-value = 7.003e-06
```

The association is highly significant. Note that this is a two degree of freedom test reflecting the fact that there are two dimensions in which the genotype distributions could differ.

Another commonly used analysis counts alleles (or chromosomes) rather than genotypes (people). The function `allele.table()` expects its first argument to be a genotype variable and counts alleles — otherwise it is the same as `table()`:

```
> abyd <- allele.table(A, affected)
> abyd

   affected
A    Control Case
  1      928  858
  2      728  464

> chisq.test(abyd)

        Pearson's Chi-squared test with Yates' continuity correction

data:  abyd
X-squared = 23.6881, df = 1, p-value = 1.133e-06
```

The chi-squared test now has one degree of freedom. This test is powerful against the more restrictive alternative hypothesis in which the effect (broadly defined) on risk of genotype 1/2 vs 1/1 is the same as for genotype 2/2 vs 1/2 — the model of *generalized additive* effects of alleles. This strategy of counting alleles and treating them as independent samples from a population also assumes Hardy-Weinberg equilibrium (HWE)

This seems an appropriate point to mention that you could test for HWE by:

```
> HWE.chisq(A)

        Pearson's Chi-squared test with simulated p-value (based on 10000
        replicates)

data:  tab
X-squared = 0.3449, df = NA, p-value = 0.5833
```

However, it would usually be more appropriate to test for HWE only in controls. This brings us to an important mechanism for selecting *subsets* of data. We first generate an object, `control`, which contains either `TRUE` or `FALSE` according to whether or not the subject is a control. We then use square brackets, as in `A[control]`, to select the genotypes for controls only:

```
> control <- (affected == "Control")
> table(control)
```

```
control
FALSE  TRUE 
  672   864 
```

```
> HWE.chisq(A[control])

        Pearson's Chi-squared test with simulated p-value (based on 10000
        replicates)

data:  tab 
X-squared = 0.0191, df = NA, p-value = 0.9417
```

You might like to try some of these commands on the other marker loci in these data.

Logistic regression

Analysis at the genotype (person level) is safer, since there is no need to assume HWE. A flexible way of carrying out such tests is by use of logistic regression. An additional bonus of this approach is that it provides estimates of the size of genotype effects.

The general approach is to carry out a logistic regression which (rather counter-intuitively) treats disease status as the outcome variable and genotype as an explanatory variable. However, there are several ways in which the genotype can be entered into the regression. This can be controlled by setting an *attribute* of the genotype variable:

```
> gcontrasts(A) <- "additive"
```

This attribute causes the genotype variable A[5] to be entered into the logistic regression as a single indicator variable coded 0, 1 or 2 (the number of copies of allele "2"). This is the model of generalized additive allelic effects and, for the logistic model, corresponds to a multiplicative model for the odds ratio case:control. In a case/control study the measure of effect is then equivalent to the relative risk for each copy of allele "2". To fit this model[6]

```
> logistic(affected ~ A)

Logistic regression:  affected ~ A 

Odds ratios (1 unit change), lower and upper confidence limits, and tests:

             OR    Lower     Upper    z-test      P-value
A:a:2 0.6911957 0.595188 0.8026901 -4.840603 1.294457e-06
```

[5] Actually this command makes a *copy* of `A` in the global environment (position 1) and it is this copy that has the attribute set. The *original* version remains in the attached dataframe (position 2) and is unchanged.

[6] Those with some previous knowledge of **R**should be aware that `logistic` is a simple wrapper for the generalized linear model function, `glm`, which fits a variety of regression models.

The relative risk bestowed by each copy of the "2" allele is 0.69.

In order to test for deviation, we need to include a "dominance" indicator in the model. Its effect here is to allow the risk for the 1/2 genotype to differ from the (geometric) mean of that for the two homozygous genotypes (1/1 and 1/2). We do this as follows:

```
> gcontrasts(A) <- "dominance"
> logistic(affected ~ A)

Logistic regression:  affected ~ A

Odds ratios (1 unit change), lower and upper confidence limits, and tests:

               OR     Lower     Upper     z-test      P-value
A:a:2   0.6932140 0.5924985 0.8110496 -4.5746506 4.770154e-06
A:d:1:2 0.9869732 0.7946085 1.2259069 -0.1185474 9.056339e-01
```

There is now an extra coefficient (`A:d:1:2`) which tests this. It is not significantly different from its value under the null hypothesis (1.0). However, if it had been significant, this parametrisation would have been hard to interpret. Instead we might prefer to report the genotype relative risks. To obtain these,

```
> gcontrasts(A) <- "genotype"
> logistic(affected ~ A)

Logistic regression:  affected ~ A

Odds ratios (1 unit change), lower and upper confidence limits, and tests:

            OR     Lower     Upper    z-test      P-value
A1/1 1.4615958 1.1666750 1.8310690  3.300675 0.0009645264
A2/2 0.7023636 0.5181763 0.9520208 -2.276865 0.0227942791
```

This output gives the genotype relative risks with the most common genotype as baseline. If we wanted to use a different baseline, say "2/2":

```
> gcontrasts(A, base = "2/2") <- "genotype"
> logistic(affected ~ A)

Logistic regression:  affected ~ A

Odds ratios (1 unit change), lower and upper confidence limits, and tests:

           OR    Lower    Upper   z-test      P-value
A1/1 2.080968 1.520216 2.848561 4.574651 4.770154e-06
A1/2 1.423764 1.050397 1.929845 2.276865 2.279428e-02
```

More advanced analyses

Incorporating extraneous variables

There are two reasons why you might wish to include an extraneous variable in such analyses:

1. you may be concerned that such a variable could *confound* the association between genotype and disease, and/or
2. you may wonder if a variable could *modify* the association.

An important example of confounding might be region of residence in a national study. In the United Kingdom, for example, there are known to be geographical variations of allele frequencies for a range of variants, variation in the population ancestry. In such studies we would wish to compare cases and controls *within* geographical regions — an approach which is sometimes called *post-stratification.* This can be simply carried out in logistic regression simply by including the stratification in the model. In our example, taking `sex` as such a variable (although we would not expect it to confound the association, since allele frequencies should not vary by sex), we would carry out the analysis as follows

```
> gcontrasts(A) <- "additive"
> logistic(affected ~ sex + A)

Logistic regression:  affected ~ sex + A

Odds ratios (1 unit change), lower and upper confidence limits, and tests:

                 OR     Lower     Upper    z-test      P-value
sexFemale 3.2764342 2.5221434 4.2563088  8.890011 6.110275e-19
A:a:2     0.6782981 0.5786404 0.7951196 -4.787809 1.686118e-06
```

You will see that the odds ratio for the genetic association is scarcely changed; as, expected, sex did not confound the association.

If we had needed to apply a two degree of freedom test, it is a little more complicated. Firts we can select appropriate contrasts (`dominance` or `genotype`):

```
> gcontrasts(A) <- "genotype"
> logistic(affected ~ sex + A)

Logistic regression:  affected ~ sex + A

Odds ratios (1 unit change), lower and upper confidence limits, and tests:

                 OR     Lower     Upper    z-test      P-value
sexFemale 3.2760877 2.5219032 4.2558139  8.889576 6.134263e-19
A1/1      1.4552493 1.1426498 1.8533679  3.040796 2.359540e-03
A2/2      0.6649649 0.4824297 0.9165652 -2.492128 1.269804e-02
```

But this gives us two one degree of freedom tests. We can't simply add the chi-squared values since they are not independent. The solution is to save the logistic regression results in a variable, `rA2` say, and carry out an *analysis of deviance*:

```
> rA2 <- logistic(affected ~ sex + A)
> anova(rA2)

Analysis of Deviance Table

Model: binomial, link: logit

Response: affected

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev
NULL                    1373    1902.80
sex     1    84.87      1372    1817.93
A       2    23.38      1370    1794.55
```

The table obtained here shows the changes in the *deviance* (basically twice the log likelihood) obtained by adding successive terms to the model. These changes in deviance are, in large samples and under the hypothesis that the added term really has no effect, distributed as chi-squared with the indicated degrees of freedom. The first test considers the effect of adding `sex` to the null model. This is highly significant ($X^2 = 84.87$ on 1 df), indicating strong association between sex and disease status (as we learnt earlier). This could be due to real differences in disease rates between males and females or to a *selection bias*. The second test considers addition of `A` to the model which includes `sex`. This is the stratified test and remains significant ($X^2 = 23.36$ on 2 df). Note that the sequence of tests depends on the order in which you entered the terms in the model; if you had specified the model as `A + sex` you would have performed a test for the effect of sex stratified by genotype — not a very sensible thing to do. Incidentally, you can calculate P-value corresponding to the deviance chi-squared as follows:

```
1 - pchisq(23.38, 2)
```

To test whether a variable such as sex *modifies* the affect of the genetic variant, we must include terms called *statistical interactions*. For example:

```
> gcontrasts(A) <- "additive"
> logistic(affected ~ sex + A + sex:A)

Logistic regression:  affected ~ sex + A + sex:A

Odds ratios (1 unit change), lower and upper confidence limits, and tests:
```

```
                       OR    Lower     Upper    z-test      P-value
sexFemale       2.2958663 1.573498 3.3498631  4.311637 1.620504e-05
A:a:2           0.4552335 0.317843 0.6520122 -4.293388 1.759675e-05
sexFemale:A:a:2 1.6591061 1.110567 2.4785844  2.472062 1.343364e-02
```

The last coefficient is the statistical interaction term and it is statistically significant at a modest level. The precise interpretation of the coefficient is that the (multiplicative) effect of the "2" allele is 1.66 stronger in females than in males. This could be of considerable interest.

This is the way in which we study *gene–environment interaction*. However we must take care about the interpretation of statistical interactions. They have a precise but limited mathematical interpretation, which may be some distance from biological interaction in the mechanistic sense.

Multiple marker analyses

Logistic regression can also be used to examine further the nature of the relationship between disease and genotype. All four markers in this dataset are associated with disease and this is surely due to the fact that there is extremely strong linkage disequilibrium (LD) between them. The causal variant may be one of these markers or, it may be a quite different variant which is in LD with all of them. To print common measures of LD in the popn dataset:

```
> LD(popn)

Pairwise LD
-----------
                       B            C            D
A D           -0.1570897    0.2255614    0.2168532
A D'           0.9788660    0.9761177    0.9764719
A Corr.       -0.6542138    0.9319561    0.8905700
A X^2       1260.0193694 2539.6175093 2261.9636245
A P-value < 2.2204e-16 < 2.2204e-16 < 2.2204e-16
A n                 1472         1462         1426

B D                        -0.1690608   -0.1768417
B D'                        0.9975868    0.9907845
B Corr.                    -0.6983191   -0.7260514
B X^2                    1427.8377394 1503.4334970
B P-value                < 2.2204e-16 < 2.2204e-16
B n                              1464         1426

C D                                      0.2338012
C D'                                     0.9969490
C Corr.                                  0.9523311
C X^2                                 2568.4383385
C P-value                             < 2.2204e-16
C n                                           1416
```

It can be seen that Lewontin's D' measure is universally high, reflecting rather rare ancestral recombination in the region. The correlation coefficients are also high, indicating that the mutations which created the variants occurred on closely related haplotypes.

If these four markers are "tags" which, hopefully, reflect causal variants(s) in the same region of strong LD, a good method for testing for a causal variant somewhere in the region is to test the effect of including *all* of them in the logistic regression:

```
> gcontrasts(A) <- "additive"
> gcontrasts(B) <- "additive"
> gcontrasts(C) <- "additive"
> gcontrasts(D) <- "additive"
> anova(logistic(affected ~ A + B + C + D))

Analysis of Deviance Table

Model: binomial, link: logit

Response: affected

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev
NULL                    1385    1908.43
A       1    22.42      1384    1886.01
B       1     1.01      1383    1885.01
C       1     6.39      1382    1878.61
D       1     0.74      1381    1877.87
```

The test for adding all four SNPs is $X^2 = 22.42 + 1.01 + 6.39 + 0.74 = 30.56$ on four degrees of freedom. This example is a poor one, however; the correlation between these four loci is so strong that all of them could not rationally have been chosen as "tags" — they mostly carry the same information.

A rather different use of logistic regression is for fine mapping, when we are attempting to discriminate between possibly causal variants. For example, if A and C were both likely causal variants, what do you think would be the implication of the following:

```
> anova(logistic(affected ~ A + C))

Analysis of Deviance Table

Model: binomial, link: logit

Response: affected
```

```
Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev
NULL                    1461    2008.77
A       1    24.03      1460    1984.75
C       1     7.60      1459    1977.15

> anova(logistic(affected ~ C + A))

Analysis of Deviance Table

Model: binomial, link: logit

Response: affected

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev
NULL                    1461    2008.77
C       1    30.88      1460    1977.90
A       1     0.75      1459    1977.15
```

Exercise 5: Design of indirect association studies

The two most important design decisions in an indirect association study are choice of markers and choice of sample size. We shall start with the first of these.

Tagging

The following data were generated by sequencing the CD25/IL2RA region (a 60kb candidate region for type 1 diabetes) in a small panel of CEPH subjects.

```
> data(CD25)
> summary(CD25)
```

```
            boxp        wellp           familyID      member
 CEPHSE32V0204:32   Min.   : 1.00   CEPH1340: 4   Min.   : 9.00
                    1st Qu.: 8.75   CEPH1362: 4   1st Qu.:11.00
                    Median :16.50   CEPH1454: 4   Median :12.00
                    Mean   :16.50   CEPH1459: 4   Mean   :12.38
                    3rd Qu.:24.25   CEPH1341: 2   3rd Qu.:13.25
                    Max.   :32.00   CEPH1344: 2   Max.   :16.00
                                    (Other) :12
  father         mother             sex           t1d    DIL8204
 Mode:logical   Mode:logical   Min.   :1.0   Min.   :1   1/1 :26 (81%)
 NA's:32        NA's:32        1st Qu.:1.0   1st Qu.:1   1/2 : 4 (12%)
                               Median :1.5   Median :1   NA's: 2 ( 6%)
                               Mean   :1.5   Mean   :1
                               3rd Qu.:2.0   3rd Qu.:1
                               Max.   :2.0   Max.   :1

 DIL4613        DIL4612        DIL4611        DIL4610          DIL4609
 1/1:17 (53%)   1/1:25 (78%)   1/1:28 (88%)   1/1 : 7 (22%)   1/1: 8 (25%)
 1/2:13 (41%)   1/2: 6 (19%)   1/2: 4 (12%)   2/1 :11 (34%)   1/2:17 (53%)
 2/2: 2 ( 6%)   2/2: 1 ( 3%)                  2/2 :12 (38%)   2/2: 7 (22%)
                                              NA's: 2 ( 6%)



 DIL8203        DIL4608        DIL4607        DIL8202        DIL4606
 2/1: 1 ( 3%)   1/1: 1 ( 3%)   2/1: 4 (12%)   2/1: 3 ( 9%)   1/1: 2 ( 6%)
 2/2:31 (97%)   2/1: 4 (12%)   2/2:28 (88%)   2/2:29 (91%)   2/1: 4 (12%)
                2/2:27 (84%)                                 2/2:26 (81%)



 DIL8201         DIL8200         DIL8199         DIL4605        DIL8198
 2/1 : 5 (16%)   2/1 : 3 ( 9%)   1/1 :28 (88%)   1/1:17 (53%)   1/1:28 (88%)
```

```
2/2 :26 (81%)   2/2 :25 (78%)   1/2 : 3 ( 9%)   1/2:13 (41%)   1/2: 4 (12%)
NA's: 1 ( 3%)   NA's: 4 (12%)   NA's: 1 ( 3%)   2/2: 2 ( 6%)



DIL4604         DIL8195         DIL8194         DIL4603        DIL4602
1/1 : 1 ( 3%)   1/1 : 1 ( 3%)   1/1 :28 (88%)   2/1: 8 (25%)   1/1:30 (94%)
2/1 : 4 (12%)   2/1 : 5 (16%)   1/2 : 1 ( 3%)   2/2:24 (75%)   1/2: 2 ( 6%)
2/2 :12 (38%)   2/2 :19 (59%)   NA's: 3 ( 9%)
NA's:15 (47%)   NA's: 7 (22%)



DIL4601        DIL4600         DIL4599         DIL8193         DIL4593
1/1: 2 ( 6%)   1/1 :21 (66%)   1/1 :19 (59%)   2/1 : 1 ( 3%)   1/1: 1 ( 3%)
2/1:14 (44%)   1/2 : 2 ( 6%)   1/2 : 2 ( 6%)   2/2 :21 (66%)   2/1: 6 (19%)
2/2:16 (50%)   NA's: 9 (28%)   2/2 : 1 ( 3%)   NA's:10 (31%)   2/2:25 (78%)
                               NA's:10 (31%)



DIL4589        DIL4588        DIL8187        DIL8186        DIL8176
2/1: 5 (16%)   2/1: 5 (16%)   1/1:29 (91%)   2/1: 5 (16%)   1/1 : 1 ( 3%)
2/2:27 (84%)   2/2:27 (84%)   1/2: 3 ( 9%)   2/2:27 (84%)   2/1 : 5 (16%)
                                                            2/2 :19 (59%)
                                                            NA's: 7 (22%)



DIL8175         DIL4585        DIL4584        DIL8166        DIL8165
1/1 :26 (81%)   1/1:12 (38%)   1/1: 2 ( 6%)   1/1:30 (94%)   2/1: 2 ( 6%)
1/2 : 3 ( 9%)   1/2:11 (34%)   2/1:11 (34%)   1/2: 2 ( 6%)   2/2:30 (94%)
NA's: 3 ( 9%)   2/2: 9 (28%)   2/2:19 (59%)



DIL8164        DIL8163         DIL8162        DIL4583        DIL8161
1/1:23 (72%)   2/1 : 2 ( 6%)   1/1:19 (59%)   1/1: 9 (28%)   2/1: 1 ( 3%)
1/2: 8 (25%)   2/2 :29 (91%)   1/2:11 (34%)   2/1:11 (34%)   2/2:31 (97%)
2/2: 1 ( 3%)   NA's: 1 ( 3%)   2/2: 2 ( 6%)   2/2:12 (38%)
```

```
DIL4581         DIL8144         DIL8112        DIL8098         DIL4580
1/1 :10 (31%)   1/1 :28 (88%)   1/1: 1 ( 3%)   2/1 : 3 ( 9%)   1/1 :12 (38%)
1/2 :10 (31%)   1/2 : 1 ( 3%)   2/1: 1 ( 3%)   2/2 :27 (84%)   1/2 :15 (47%)
2/2 : 9 (28%)   NA's: 3 ( 9%)   2/2:30 (94%)   NA's: 2 ( 6%)   2/2 : 2 ( 6%)
NA's: 3 ( 9%)                                                  NA's: 3 ( 9%)
```

```
DIL8091         DIL4579         DIL8090        DIL8089        DIL8088
1/1 : 1 ( 3%)   2/1 : 2 ( 6%)   1/1:11 (34%)   1/1:27 (84%)   2/1: 2 ( 6%)
2/1 : 4 (12%)   2/2 :29 (91%)   1/2:17 (53%)   1/2: 4 (12%)   2/2:30 (94%)
2/2 :26 (81%)   NA's: 1 ( 3%)   2/2: 4 (12%)   2/2: 1 ( 3%)
NA's: 1 ( 3%)
```

```
DIL4575        DIL4574        DIL8087         DIL4573
1/1:10 (31%)   1/1:11 (34%)   1/1 : 1 ( 3%)   1/1 : 2 ( 6%)
1/2:13 (41%)   1/2:17 (53%)   2/1 : 2 ( 6%)   2/1 :16 (50%)
2/2: 9 (28%)   2/2: 4 (12%)   2/2 :16 (50%)   2/2 :12 (38%)
                              NA's:13 (41%)   NA's: 2 ( 6%)
```

We start by calculating LD statistics and displaying the D' patterns.

```
> LDtable(LD(CD25), which = "D'")
```

It can be seen that LD is strong. However, many of the SNPs identified have low frequency. We shall restrict attention to the subset of SNPs with minor allele frequencies exceeding 5%. To create this restricted dataset:

```
> CD25.fr <- mamerge(CD25, maf = 0.05)
```

The warning messages just note the dropping of SNPs with MAF $< 5\%$. We now recalculate the LD statistics and inspect the D' pattern:

```
> ld <- LD(CD25.fr)
> LDtable(ld, which = "D'")
```

This is not such a simple LD structure — there are not clearly differentiated "blocks" but, equally, LD is not unbroken across the whole region.

Whereas D' tells us about the history of recombinations, the important measure which dictates the power to detect at an effect at one locus by typing another is the value of the squared correlation coefficient between them, r^2. We can extract this from the LS summary very easily:

```
> r2 <- (ld$r)^2
> r2
```

```
        DIL8204    DIL4613    DIL4612     DIL4611    DIL4610    DIL4609
DIL8204      NA 0.02560562 0.01001133 0.004593949 0.09959370 0.07568741
DIL4613      NA         NA 0.39451975 0.001137101 0.25801150 0.33938802
DIL4612      NA         NA         NA 0.058218722 0.10184129 0.13397339
DIL4611      NA         NA         NA          NA 0.04734829 0.07064158
DIL4610      NA         NA         NA          NA         NA 0.65916216
DIL4609      NA         NA         NA          NA         NA         NA
DIL4608      NA         NA         NA          NA         NA         NA
DIL4607      NA         NA         NA          NA         NA         NA
DIL4606      NA         NA         NA          NA         NA         NA
DIL8201      NA         NA         NA          NA         NA         NA
DIL8200      NA         NA         NA          NA         NA         NA
DIL4605      NA         NA         NA          NA         NA         NA
DIL8198      NA         NA         NA          NA         NA         NA
DIL4604      NA         NA         NA          NA         NA         NA
DIL8195      NA         NA         NA          NA         NA         NA
DIL4603      NA         NA         NA          NA         NA         NA
DIL4601      NA         NA         NA          NA         NA         NA
DIL4599      NA         NA         NA          NA         NA         NA
DIL4593      NA         NA         NA          NA         NA         NA
DIL4589      NA         NA         NA          NA         NA         NA
DIL4588      NA         NA         NA          NA         NA         NA
DIL8186      NA         NA         NA          NA         NA         NA
DIL8176      NA         NA         NA          NA         NA         NA
DIL8175      NA         NA         NA          NA         NA         NA
DIL4585      NA         NA         NA          NA         NA         NA
DIL4584      NA         NA         NA          NA         NA         NA
DIL8164      NA         NA         NA          NA         NA         NA
DIL8162      NA         NA         NA          NA         NA         NA
DIL4583      NA         NA         NA          NA         NA         NA
DIL4581      NA         NA         NA          NA         NA         NA
DIL8098      NA         NA         NA          NA         NA         NA
DIL4580      NA         NA         NA          NA         NA         NA
DIL8091      NA         NA         NA          NA         NA         NA
DIL8090      NA         NA         NA          NA         NA         NA
DIL8089      NA         NA         NA          NA         NA         NA
DIL4575      NA         NA         NA          NA         NA         NA
DIL4574      NA         NA         NA          NA         NA         NA
DIL8087      NA         NA         NA          NA         NA         NA
DIL4573      NA         NA         NA          NA         NA         NA
            DIL4608     DIL4607     DIL4606     DIL8201     DIL8200     DIL4605
DIL8204 0.002808320 0.930962358 0.010011329 0.006082692 0.003903699 0.025605623
DIL4613 0.037211199 0.023898582 0.051513296 0.031565644 0.155998951 0.999237501
DIL4612 0.014605481 0.009343907 0.020275310 0.012395052 0.005578746 0.394519754
DIL4611 0.006723128 0.004277336 0.009343907 0.005677179 0.003634859 0.001137101
DIL4610 0.144463774 0.092954122 0.199720637 0.122520206 0.040206520 0.258011504
```

```
DIL4609 0.109809428 0.070641582 0.151833427 0.093131902 0.052918930 0.339388023
DIL4608          NA 0.004433244 0.014605481 0.008942954 0.005711604 0.037211199
DIL4607          NA          NA 0.009343907 0.005677179 0.003634859 0.023898582
DIL4606          NA          NA          NA 0.613078938 0.007936920 0.051513296
DIL8201          NA          NA          NA          NA 0.004823480 0.031565644
DIL8200          NA          NA          NA          NA          NA 0.155998951
DIL4605          NA          NA          NA          NA          NA          NA
DIL8198          NA          NA          NA          NA          NA          NA
DIL4604          NA          NA          NA          NA          NA          NA
DIL8195          NA          NA          NA          NA          NA          NA
DIL4603          NA          NA          NA          NA          NA          NA
DIL4601          NA          NA          NA          NA          NA          NA
DIL4599          NA          NA          NA          NA          NA          NA
DIL4593          NA          NA          NA          NA          NA          NA
DIL4589          NA          NA          NA          NA          NA          NA
DIL4588          NA          NA          NA          NA          NA          NA
DIL8186          NA          NA          NA          NA          NA          NA
DIL8176          NA          NA          NA          NA          NA          NA
DIL8175          NA          NA          NA          NA          NA          NA
DIL4585          NA          NA          NA          NA          NA          NA
DIL4584          NA          NA          NA          NA          NA          NA
DIL8164          NA          NA          NA          NA          NA          NA
DIL8162          NA          NA          NA          NA          NA          NA
DIL4583          NA          NA          NA          NA          NA          NA
DIL4581          NA          NA          NA          NA          NA          NA
DIL8098          NA          NA          NA          NA          NA          NA
DIL4580          NA          NA          NA          NA          NA          NA
DIL8091          NA          NA          NA          NA          NA          NA
DIL8090          NA          NA          NA          NA          NA          NA
DIL8089          NA          NA          NA          NA          NA          NA
DIL4575          NA          NA          NA          NA          NA          NA
DIL4574          NA          NA          NA          NA          NA          NA
DIL8087          NA          NA          NA          NA          NA          NA
DIL4573          NA          NA          NA          NA          NA          NA
            DIL8198      DIL4604      DIL8195      DIL4603     DIL4601
DIL8204 0.930962358 0.0151010375 0.1398035757 1.001133e-02 0.027715033
DIL4613 0.023898582 0.4608730963 0.0586857557 5.146797e-02 0.923629747
DIL4612 0.009343907 0.3792336488 0.0231206193 2.473606e-09 0.364665483
DIL4611 0.004277336 0.0140943017 0.0106695790 9.343907e-03 0.005173307
DIL4610 0.092954122 0.0458457588 0.0363811828 1.997206e-01 0.279127535
DIL4609 0.070641582 0.0537416241 0.0119556109 1.518334e-01 0.258750855
DIL4608 0.004433244 0.0219836830 0.0166645025 1.260873e-02 0.040268997
DIL4607 0.997470953 0.0140943017 0.1400954774 9.343907e-03 0.025867364
DIL4606 0.009343907 0.0005697671 0.0618506139 3.090607e-01 0.055738733
DIL8201 0.005677179 0.0186520812 0.0204937180 6.130789e-01 0.034158782
DIL8200 0.003634859 0.2634075721 0.0321664938 7.936920e-03 0.144187695
```

```
DIL4605 0.023898582 0.4608730963 0.0586857557 5.146797e-02 0.923629747
DIL8198          NA 0.0140943017 0.1400954774 9.343907e-03 0.025867364
DIL4604          NA           NA 0.0001587816 3.044238e-02 0.423842797
DIL8195          NA           NA           NA 9.096365e-05 0.063500451
DIL4603          NA           NA           NA           NA 0.055643106
DIL4601          NA           NA           NA           NA          NA
DIL4599          NA           NA           NA           NA          NA
DIL4593          NA           NA           NA           NA          NA
DIL4589          NA           NA           NA           NA          NA
DIL4588          NA           NA           NA           NA          NA
DIL8186          NA           NA           NA           NA          NA
DIL8176          NA           NA           NA           NA          NA
DIL8175          NA           NA           NA           NA          NA
DIL4585          NA           NA           NA           NA          NA
DIL4584          NA           NA           NA           NA          NA
DIL8164          NA           NA           NA           NA          NA
DIL8162          NA           NA           NA           NA          NA
DIL4583          NA           NA           NA           NA          NA
DIL4581          NA           NA           NA           NA          NA
DIL8098          NA           NA           NA           NA          NA
DIL4580          NA           NA           NA           NA          NA
DIL8091          NA           NA           NA           NA          NA
DIL8090          NA           NA           NA           NA          NA
DIL8089          NA           NA           NA           NA          NA
DIL4575          NA           NA           NA           NA          NA
DIL4574          NA           NA           NA           NA          NA
DIL8087          NA           NA           NA           NA          NA
DIL4573          NA           NA           NA           NA          NA
            DIL4599      DIL4593      DIL4589      DIL4588      DIL8186
DIL8204 0.024789916 1.001133e-02 0.0058708400 0.0058708400 0.0058708400
DIL4613 0.035970825 5.151330e-02 0.0304993697 0.0304993697 0.0304165631
DIL4612 0.014118631 2.027531e-02 0.0119780173 0.0119780173 0.0034472410
DIL4611 0.006493820 7.217374e-09 0.0054794507 0.0054794507 0.0054794507
DIL4610 0.139648315 1.018413e-01 0.1183735930 0.1183735930 0.0434222637
DIL4609 0.106149113 1.339734e-01 0.0899805777 0.0899805777 0.0251994367
DIL4608 0.785141782 1.260873e-02 0.0002460259 0.0002460259 0.0002460259
DIL4607 0.027172125 9.343907e-03 0.0054794507 0.0054794507 0.0054794507
DIL4606 0.014118631 2.029827e-02 0.3410283023 0.3410283023 0.1657759966
DIL8201 0.008640899 1.069345e-03 0.6022606542 0.6022606542 0.0923047272
DIL8200 0.005516896 7.936920e-03 0.0046555877 0.0046555877 0.0419701478
DIL4605 0.035970825 5.151330e-02 0.0304993697 0.0304993697 0.0304165631
DIL8198 0.027172125 9.343907e-03 0.0054794507 0.0054794507 0.0054794507
DIL4604 0.021250894 3.047102e-02 0.0180231453 0.0180231453 0.0516660468
DIL8195 0.016109019 2.312062e-02 0.0011861506 0.0011861506 0.2412415293
DIL4603 0.014118631 2.029827e-02 0.5923186601 0.5923186601 0.1447620480
DIL4601 0.038926697 5.573873e-02 0.0330046871 0.0330046871 0.0329690538
```

```
DIL4599          NA 9.310817e-04 0.0083510005 0.0083510005 0.0083510005
DIL4593          NA           NA 0.0034472410 0.0034472410 0.0119780173
DIL4589          NA           NA           NA 0.9985573052 0.0960067566
DIL4588          NA           NA           NA           NA 0.0960067566
DIL8186          NA           NA           NA           NA           NA
DIL8176          NA           NA           NA           NA           NA
DIL8175          NA           NA           NA           NA           NA
DIL4585          NA           NA           NA           NA           NA
DIL4584          NA           NA           NA           NA           NA
DIL8164          NA           NA           NA           NA           NA
DIL8162          NA           NA           NA           NA           NA
DIL4583          NA           NA           NA           NA           NA
DIL4581          NA           NA           NA           NA           NA
DIL8098          NA           NA           NA           NA           NA
DIL4580          NA           NA           NA           NA           NA
DIL8091          NA           NA           NA           NA           NA
DIL8090          NA           NA           NA           NA           NA
DIL8089          NA           NA           NA           NA           NA
DIL4575          NA           NA           NA           NA           NA
DIL4574          NA           NA           NA           NA           NA
DIL8087          NA           NA           NA           NA           NA
DIL4573          NA           NA           NA           NA           NA
             DIL8176     DIL8175     DIL4585      DIL4584      DIL8164
DIL8204 0.0114316917 0.003764864 0.058874204 0.0874844796 0.0130274305
DIL4613 0.0586857557 0.019561332 0.436090098 0.1104922784 0.0667535547
DIL4612 0.0231206193 0.007651657 0.172156911 0.0435798257 0.0263172065
DIL4611 0.0106695790 0.003505782 0.080111094 0.0202021025 0.0037258035
DIL4610 0.0442932818 0.076067406 0.081269945 0.0406636478 0.0424193639
DIL4609 0.0540147297 0.057809722 0.019819568 0.0124679271 0.0187563290
DIL4608 0.0053139164 0.005507136 0.085437146 0.1907616000 0.0007002324
DIL4607 0.0106695790 0.003505782 0.054949257 0.0807682280 0.0121589352
DIL4606 0.0231206193 0.380830500 0.009311843 0.0435798257 0.0263172065
DIL8201 0.0097279696 0.210245307 0.005084462 0.0266968985 0.0123301414
DIL8200 0.0090625533 0.002958326 0.068025475 0.0217202683 0.0103271705
DIL4605 0.0586857557 0.019561332 0.436090098 0.1104922784 0.0667535547
DIL8198 0.0106695790 0.003505782 0.054949257 0.0807682280 0.0121589352
DIL4604 0.0347090295 0.036691851 0.258192316 0.0653442803 0.0394790244
DIL8195 0.0263335843 0.334183083 0.010703766 0.1363763319 0.0299759491
DIL4603 0.0231206193 0.106005756 0.002469593 0.0435798257 0.0263172065
DIL4601 0.0635004512 0.021172329 0.471780336 0.1195491125 0.0722304727
DIL4599 0.0006326678 0.005319497 0.082589241 0.3260413556 0.0009448029
DIL4593 0.8767195560 0.007651657 0.118154510 0.0350906071 0.7706835792
DIL4589 0.0093513085 0.022760893 0.070011869 0.0257955127 0.0155135956
DIL4588 0.0093513085 0.022760893 0.070011869 0.0257955127 0.0155135956
DIL8186 0.0136650220 0.642056040 0.021240804 0.0257955127 0.0155603072
DIL8176           NA 0.008736420 0.134620437 0.0005077704 0.8781481626
```

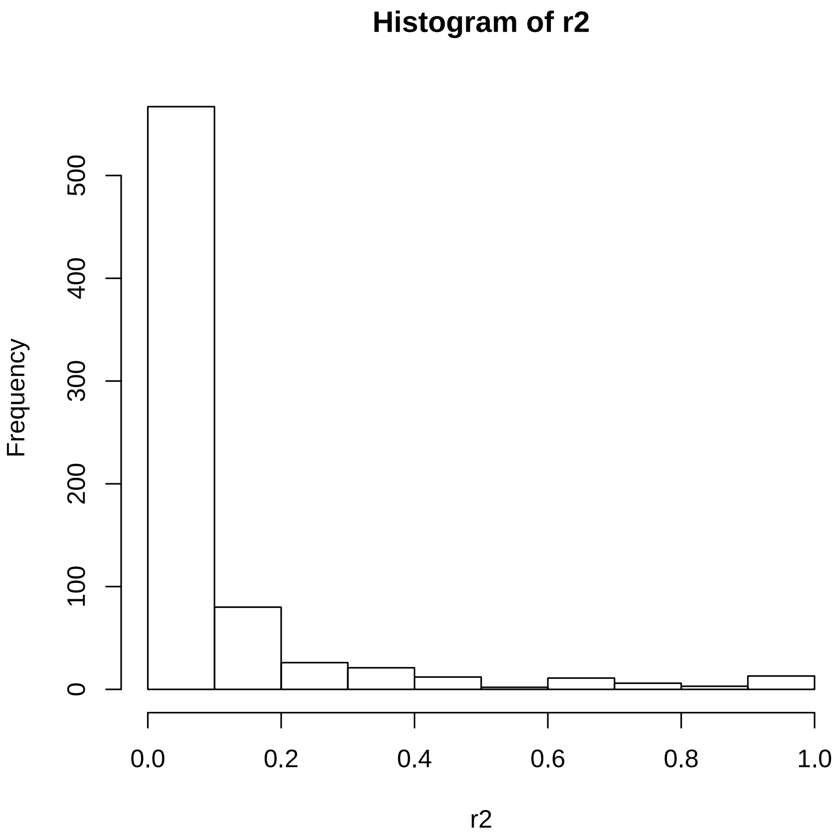
```
DIL8175           NA          NA 0.065558340 0.0165366128 0.0099551197
DIL4585           NA          NA          NA 0.2533315321 0.1531324429
DIL4584           NA          NA          NA           NA 0.0334930416
DIL8164           NA          NA          NA           NA           NA
DIL8162           NA          NA          NA           NA           NA
DIL4583           NA          NA          NA           NA           NA
DIL4581           NA          NA          NA           NA           NA
DIL8098           NA          NA          NA           NA           NA
DIL4580           NA          NA          NA           NA           NA
DIL8091           NA          NA          NA           NA           NA
DIL8090           NA          NA          NA           NA           NA
DIL8089           NA          NA          NA           NA           NA
DIL4575           NA          NA          NA           NA           NA
DIL4574           NA          NA          NA           NA           NA
DIL8087           NA          NA          NA           NA           NA
DIL4573           NA          NA          NA           NA           NA
             DIL8162     DIL4583      DIL4581      DIL8098     DIL4580
DIL8204 0.0874844796 0.058874204 6.633944e-02 0.0036357725 0.035458729
DIL4613 0.1104922784 0.436090098 2.154360e-01 0.0265281728 0.064221941
DIL4612 0.0435798257 0.172156911 8.341581e-02 0.0073864128 0.042957424
DIL4611 0.0202021025 0.080111094 7.110132e-02 0.0033857641 0.136359710
DIL4610 0.0406636478 0.081269945 4.077470e-03 0.0024475386 0.010469772
DIL4609 0.0124679271 0.019819568 2.422068e-03 0.0005504595 0.105810954
DIL4608 0.1907616000 0.085437146 9.625869e-02 0.0053170171 0.211878416
DIL4607 0.0807682280 0.054949257 6.191681e-02 0.0033857641 0.033338159
DIL4606 0.0435798257 0.009311843 2.284138e-02 0.1207575208 0.069424005
DIL8201 0.0266968985 0.005084462 9.224707e-06 0.0228648190 0.042557399
DIL8200 0.0217202683 0.068025475 5.257667e-02 0.0028574801 0.027381303
DIL4605 0.1104922784 0.436090098 2.154360e-01 0.0265281728 0.064221941
DIL8198 0.0807682280 0.054949257 6.191681e-02 0.0033857641 0.033338159
DIL4604 0.0653442803 0.258192316 1.309320e-01 0.2449551572 0.104107013
DIL8195 0.1363763319 0.010703766 1.396678e-03 0.0998935541 0.003173409
DIL4603 0.0435798257 0.002469593 5.517347e-02 0.0030781422 0.069424005
DIL4601 0.1195491125 0.471780336 2.285799e-01 0.0263252832 0.067926007
DIL4599 0.3260413556 0.082589241 9.305007e-02 0.0051359515 0.204815802
DIL4593 0.0350906071 0.118154510 1.331083e-01 0.0073864128 0.069429246
DIL4589 0.0257955127 0.070011869 7.887818e-02 0.0246225248 0.041118870
DIL4588 0.0257955127 0.070011869 7.887818e-02 0.0246225248 0.041118870
DIL8186 0.0257955127 0.021240804 1.431644e-02 0.2184465121 0.041118870
DIL8176 0.0005077704 0.134620437 1.516596e-01 0.0084331736 0.079047746
DIL8175 0.0165366128 0.065558340 5.818591e-02 0.4078238507 0.026390006
DIL4585 0.2533315321 0.999367157 6.257820e-01 0.0632643377 0.027436459
DIL4584 0.9992701569 0.253331532 1.922497e-01 0.0159600936 0.250903752
DIL8164 0.0334930416 0.153132443 1.725152e-01 0.0096091783 0.089965795
DIL8162           NA 0.253331532 1.922497e-01 0.0159600936 0.250903752
DIL4583           NA          NA 6.257820e-01 0.0632643377 0.027436459
```

```
DIL4581           NA          NA           NA 0.0561502030 0.007591774
DIL8098           NA          NA           NA           NA 0.025468274
DIL4580           NA          NA           NA           NA          NA
DIL8091           NA          NA           NA           NA          NA
DIL8090           NA          NA           NA           NA          NA
DIL8089           NA          NA           NA           NA          NA
DIL4575           NA          NA           NA           NA          NA
DIL4574           NA          NA           NA           NA          NA
DIL8087           NA          NA           NA           NA          NA
DIL4573           NA          NA           NA           NA          NA
             DIL8091      DIL8090      DIL8089     DIL4575      DIL4574
DIL8204 3.144728e-01 0.0207843042 0.3216023347 0.066771314 0.0207843042
DIL4613 3.854017e-02 0.0412893494 0.0372111986 0.104254206 0.0412893494
DIL4612 1.512711e-02 0.0914196105 0.0146054808 0.034634014 0.0914196105
DIL4611 3.594206e-03 0.1035954904 0.0044332444 0.062319893 0.1035954904
DIL4610 5.663831e-02 0.0026181516 0.0587951384 0.053745321 0.0026181516
DIL4609 1.137312e-01 0.0916052903 0.1098094275 0.178088595 0.0916052903
DIL4608 1.453162e-03 0.1609912339 0.0008728294 0.096884732 0.1609912339
DIL4607 3.152677e-01 0.0136861613 0.3236243520 0.062319893 0.0136861613
DIL4606 3.911336e-02 0.0350527170 0.0410189798 0.004731064 0.0350527170
DIL8201 9.642594e-03 0.0560342624 0.0104306344 0.026199925 0.0560342624
DIL8200 5.920221e-03 0.0360693496 0.0057116038 0.010410483 0.0360693496
DIL4605 3.854017e-02 0.0412893494 0.0372111986 0.104254206 0.0412893494
DIL8198 3.152677e-01 0.0136861613 0.3236243520 0.062319893 0.0136861613
DIL4604 2.276881e-02 0.0070590746 0.0219836830 0.025611971 0.0070590746
DIL8195 4.477273e-02 0.0455589952 0.0437513888 0.067293679 0.0455589952
DIL4603 1.308147e-03 0.0913841014 0.0016339778 0.010434521 0.0913841014
DIL4601 4.170718e-02 0.0441643501 0.0402689971 0.122291010 0.0441643501
DIL4599 6.766615e-03 0.1556248594 0.0089957565 0.093655241 0.1556248594
DIL4593 7.523288e-03 0.0913841014 0.0126087347 0.064741647 0.0913841014
DIL4589 8.955497e-03 0.0541393161 0.0086428260 0.025199437 0.0541393161
DIL4588 8.955497e-03 0.0541393161 0.0086428260 0.025199437 0.0541393161
DIL8186 1.039728e-02 0.0021990729 0.0112147123 0.079391107 0.0021990729
DIL8176 1.725966e-02 0.1041167530 0.0166645025 0.172996058 0.1041167530
DIL8175 4.139905e-02 0.0076450196 0.0436591314 0.051000332 0.0076450196
DIL4585 1.125145e-03 0.0063188947 0.0022735587 0.006383743 0.0063188947
DIL4584 3.258274e-02 0.1609780519 0.0315455250 0.050182006 0.1609780519
DIL8164 1.972124e-02 0.1184332210 0.0190276625 0.093572802 0.1184332210
DIL8162 3.258274e-02 0.1609780519 0.0315455250 0.050182006 0.1609780519
DIL4583 1.125145e-03 0.0063188947 0.0022735587 0.006383743 0.0063188947
DIL4581 1.404462e-06 0.0002022256 0.0001953281 0.002405364 0.0002022256
DIL8098 5.511018e-03 0.0818076584 0.0053170171 0.049216372 0.0818076584
DIL4580 5.203600e-02 0.7596114436 0.0501473968 0.457359070 0.7596114436
DIL8091           NA 0.0684237540 0.9639504802 0.100344901 0.0684237540
DIL8090           NA           NA 0.0660643142 0.601745357 0.9994320552
DIL8089           NA           NA           NA 0.096884732 0.0660643142
```

```
DIL4575           NA           NA           NA          NA 0.6017453567
DIL4574           NA           NA           NA          NA           NA
DIL8087           NA           NA           NA          NA           NA
DIL4573           NA           NA           NA          NA           NA
            DIL8087    DIL4573
DIL8204 0.290718581 0.04304362
DIL4613 0.042318618 0.06819161
DIL4612 0.016610155 0.03558727
DIL4611 0.077300664 0.13285926
DIL4610 0.056042549 0.01473747
DIL4609 0.124881310 0.02903990
DIL4608 0.082734872 0.20644175
DIL4607 0.370579059 0.03099322
DIL4606 0.093142200 0.07125401
DIL8201 0.019070194 0.04368047
DIL8200 0.006513356 0.02810531
DIL4605 0.042318618 0.06819161
DIL8198 0.370579059 0.03099322
DIL4604 0.025001051 0.10685177
DIL8195 0.067976308 0.00905028
DIL4603 0.002421989 0.07125401
DIL4601 0.045796114 0.07156556
DIL4599 0.632163810 0.19956036
DIL4593 0.024751902 0.07115846
DIL4589 0.009844468 0.04220391
DIL4588 0.009844468 0.04220391
DIL8186 0.020673635 0.04220391
DIL8176 0.019007820 0.08114282
DIL8175 0.109558273 0.02708772
DIL4585 0.004712566 0.01787636
DIL4584 0.010091278 0.25951288
DIL8164 0.003928730 0.09242631
DIL8162 0.010091278 0.25951288
DIL4583 0.004712566 0.01787636
DIL4581 0.004225256 0.00363220
DIL8098 0.006062595 0.02614154
DIL4580 0.001897940 0.97390364
DIL8091 0.909222439 0.05330344
DIL8090 0.075264735 0.77959574
DIL8089 0.877869941 0.05149917
DIL4575 0.110182636 0.46939068
DIL4574 0.075264735 0.77959574
DIL8087          NA 0.05867867
DIL4573          NA         NA
```

(The brackets are not necessary but are included to make it clear exactly what has been done). It can be seen that the values of r^2 are much smaller than the values of D'. This is even clearer if we plot a histogram of these values

```
> hist(r2)
```



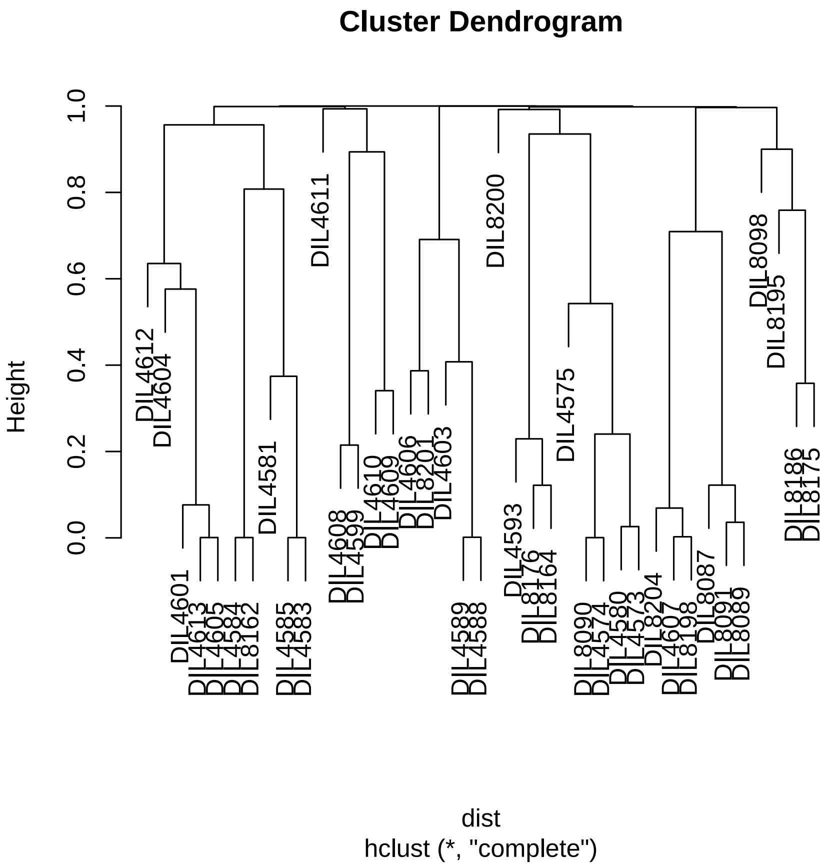
This is somewhat depressing, suggesting that the phylogeny (the "family tree") of the observed haplotypes is rather fragmented and that many "tags" may be necessary to mark al the variation in this region.

The most commonly used methods of selecting tag SNPs are based, lloosely, on cluster analysis. We first gather the SNPs into groups, or clusters, within which the r^2 values are small. We then select a representative member of each cluster to use as a tag. The most widely cited example of this is due to Carlson et al. (2004). It has been pointed out recently (Rinaldo et al., 2005) that this method is closely related to the standard statistical method of *complete linkage hierarchical cluster analysis*, and that is how the calculations will be approached here. We start by expressing the "distance" between SNPs as $(1-r^2)$:

```
> dist <- as.dist(t(1 - r2))
```

The use of the `t()` function in the above needs a word of explanation. This swaps the rows and columns of a matrix and is necessary here because the `as.dist()` function (which prepares a distance matrix for cluster analysis) annoyingly requires the diagonally opposite part of the matrix from that which is calculated by LD(). We now calculate, and plot, the hierachical cluster analysis using the complete linkage method:

```
> hier <- hclust(dist, method = "complete")
> plot(hier)
```



You might like to widen the graphics window to see things better. To understand what this graph means it is helpful to look at a tiny part of it comprising three SNPs:

```
> three <- c("DIL4581", "DIL4585", "DIL4583")
```

The axis labelled "height" in the plot shows distances, $(1 - r^2)$ and we can look at the distances between these SNPs by simply typing

```
> 1 - r2[three, three]
```

```
         DIL4581 DIL4585      DIL4583
DIL4581       NA      NA           NA
DIL4585 0.374218      NA 0.0006328426
DIL4583 0.374218      NA           NA
```

We divide the total collection of SNPs into clusters, in effect, by drawing a horizontal line across the plot. If we draw a line at height 0.5, this will create clusters in which the maximum distance between SNPs is 0.5, so that the minimum r^2 is also 0.5. We can simultaneously draw the clusters on the graph and save them for later processing with the following command:

```
> clusters <- rect.hclust(hier, h = 0.5)
```

Our final task to complete the process of choosing tags is to select a "central" representative of each cluster. this is computed using the clusters we have just calculated and the distance matrix:

```
> tags <- representative(clusters, dist)
```

The `tags` object so computed is a list with three elements:

1. `$rep`: the representatives or, here, the tags chosen
2. `$is.rep`: a logical array saying whether each of the original SNPs was or was not chosen, and
3. `$dist`: the mean distance between other SNPs within the cluster and the tag.

You should look at the solution by typing `tags` and identifying the selected tags on the plot. Note that we have chosen half the SNPs as tags while only guaranteeing $r^2 > 0.5$.

Multiple tagging

In this section we shall use some fancy **R**to show how much better tagging can be achieved by using the tags together rather than one-at-a-time. We'll start by creating two lists of SNPs — those chosen as tags and those not chosen. To make things hard for ourselves, we'll start by randomly choosing only half of the tags chosen by our previous approach, by an electronic flip of a coin:

```
> choose <- tags$is.rep & (runif(39) < 0.5)
> snps <- hier$labels
> chosen <- snps[choose]
> chosen
```

```
 [1] "DIL4612" "DIL4611" "DIL4610" "DIL8200" "DIL4605" "DIL4604" "DIL4585"
 [8] "DIL4584" "DIL4575" "DIL4574"
```

```
> not <- snps[!choose]
> not
```

```
 [1] "DIL8204" "DIL4613" "DIL4609" "DIL4608" "DIL4607" "DIL4606" "DIL8201"
 [8] "DIL8198" "DIL8195" "DIL4603" "DIL4601" "DIL4599" "DIL4593" "DIL4589"
[15] "DIL4588" "DIL8186" "DIL8176" "DIL8175" "DIL8164" "DIL8162" "DIL4583"
[22] "DIL4581" "DIL8098" "DIL4580" "DIL8091" "DIL8090" "DIL8089" "DIL8087"
[29] "DIL4573"
```

(The first line says that a SNP will only be chosen if it was chosen as a cluster representative and if a random variable, evenly distributed over the region (0,1) is less than 0.5). We now want to demonstrate how well all the chosen SNPs, taken together, predict those not chosen. For this prediction we will use simple linear regression, coding each SNP genotype as 0, 1 or 2. For a justification of this, see Clayton et al. (2004).

We start by creating the prediction matrix adding one chosen SNP at a time

```
> mat <- NULL
> for (snp in chosen) {
+     mat <- cbind(mat, allele.count(CD25[[snp]], 2))
+ }
> colnames(mat) <- chosen
```

The function `cbind` adds a column to mat, and the function `allele.count` constructs the 0, 1, 2 scoring of genotypes by counting the "2" allele. You can look at the first 5 rows of mat to see what you have achieved by typing

```
> mat[1:5, ]
```

```
     DIL4612 DIL4611 DIL4610 DIL8200 DIL4605 DIL4604 DIL4585 DIL4584 DIL4575
[1,]       0       0       1       2       0       2       0       1       1
[2,]       0       0       2       2       0       2       0       1       1
[3,]       0       0       2       2       0       2       0       2       2
[4,]       0       0       2       2       2      NA       2       2       0
[5,]       0       0       0      NA       0       2       0       1       2
     DIL4574
[1,]       1
[2,]       1
[3,]       0
[4,]       2
[5,]       0
```

We'll now code each of the remaining SNPs in turn similarly, predict it by multiple regression on `mat`, and print the R^2. But first we have to force **R**to only use subjects without missing values in these computations:

```
> options(na.action = na.omit)
> for (snp in not) {
+     gt <- allele.count(CD25[[snp]], 2)
+     reg <- lm(gt ~ mat)
+     print(summary(reg)$r.squared)
+ }
```

```
[1] 0.6893491
[1] 1
[1] 1
```

```
[1] 0.6725572
[1] 0.6893491
[1] 0.9220374
[1] 0.8011364
[1] 0.6893491
[1] 0.7461538
[1] 0.8397436
[1] 1
[1] 1
[1] 0.932916
[1] 0.7781065
[1] 0.7781065
[1] 1
[1] 0.9318182
[1] 1
[1] 0.932916
[1] 1
[1] 1
[1] 0.9275148
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
[1] 1
```

It usually works pretty well — certainly much better than single tagging. Note, however, that omitting subjects with anything missing could lead to multiple tagging being a very costly strategy indeed. For an alternative approach based on imputation of missing values, see Vella et al. (2005). This paper describes the end result of the CD25 study.

Power

The `DGCgenetics` package contains a number of functions for power calculations. These assume a generalized codominant mode of action for the causal variant. Thus, for a disease trait, each copy of the risk allele is assumed to multiply the risk by the same amount, $\boldsymbol{\theta}$ say (`theta`) and, for a quantitative trait, each copy of the allele adds the same amount to the trait mean. For case-control studies and quantitative trait studies, the power is calculated by `htPower.cc` and `htPower.qtl` respectively. Sample size calculations are carried out using the corresponding `htSampleSize` functions. Consult the help page for details.

How big must a case-control study be in order to detect a causal variant in a candidate gene with frequency 0.1 (10%) and $\boldsymbol{\theta} = 1.5$. Even for a candidate gene current opinion would suggest that you should aim for a significance level no larger than $\alpha = 10^{-4}$. You might like to experiment with different values for these parameters.

For quantitative traits, it might be reasonable to try and detect variants responsible for only 1% of total variance of the trait. How big do such studies need to be?

References

Carlson et al. (2004) Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium. *Am.J.Hum.Genet.*, **61**:525.

Clayton et al. (2004) The use of unphased multilocus genotype data in indirect association studies *Genetic Epidemiology*, **27**:415–428.

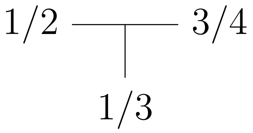
Vella et al. (2005) Localization of a type 1 diabetes locus in the IL2RA/CD25 region by use of tag single-nucleotide polymorphisms. *Am.J.Hum.Genet.*, **76**:773–779.

Rinaldo et al. (2005) Characterization of Multilocus Linkage Disequilibrium *Genetic Epidemiology*, **28**:193–206.

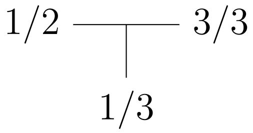
Exercise 6: Transmission/disequilibrium: the TDT and some extensions

Informative transmissions

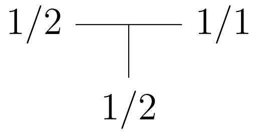
The following represent trios of an affected offspring and both parents. Alleles are coded 1–4 and unknown genotypes are denoted by ?/?.



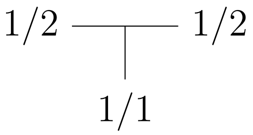
(1)



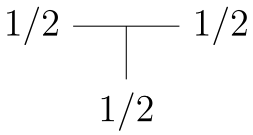
(2)



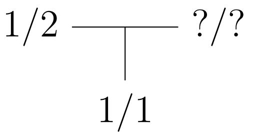
(3)



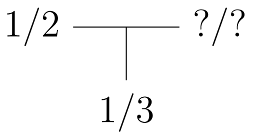
(4)



(5)



(6)



(7)

Determine, in each case, how many informative transmissions are provided by the family. Using only these informative transmissions, make a table of how many times each allele was either transmitted or not transmitted.

Preparing computer files for pedigree data

The most commonly used method for entering family data is in the standard "pre-ped" format which was introduced in the LINKAGE package. This has one data record per family member, and there are six standard fields. These, together with their default names for my **R**package, are as follows:

1. `pedigree`: the code identifying different families in the dataset.

2. `id`: the code identifying an individual within the family.

3. `id.father`: the identifier for the subject's father.

4. `id.mother`: the identifier for the subject's mother. This and the previous field need only be filled in if the parent is present in the file. Note that sometimes it is necessary to supply records for parents even when no data are available for them, in order to make clear the relationship between subjects for whom we do have data.

5. `sex`: this can be numerically coded, as 1 for male and 2 for female. Alternatively it can be stored as a "factor" (the name for a categorical variable in **R**),

providing male is the first level and female is the second level (this ensures the correct underlying numerical codes).

6. `affected`: the disease status. The usual coding is 1 for not affected and 2 for affected. But this can be stored as a factor in the same way as `sex`.

The dataframe `tdt.exercise` contains a template of the datafile for the exercise you have just completed, but the marker genotype is missing. You can load and browse the dataframe, and enter the marker data by:

```
data(tdt.exercise)
fix(tdt.exercise)
```

(or, in *MS Windows*, you can initiate this from the drop-down menus.). You should enter the genotypes as they are written in the exercise — as character strings such as 1/2. When you have finished, quit the data editor. You might like to try `summary(tdt.exercise)` to check that all is well. But `marker` is still stored as a character string rather than as a genotype variable. The function `makeGenotypes` is useful for this purpose. It can simultaneously change the storage mode of many variables within a dataframe in this way, but the following command converts the variable type of the single marker, making a new copy of the dataframe:

```
tdtex <- makeGenotypes(tdt.exercise, convert="marker")
```

```
> data(tdt.solution)
> tdtex <- tdt.solution
```

The dataframe should look like this:

```
> show(tdtex)
```

```
   pedigree id id.father id.mother sex affected marker
1         1  1        NA        NA   1       NA    1/2
2         1  2        NA        NA   2       NA    3/4
3         1  3         1         2  NA        2    1/3
4         2  1        NA        NA   1       NA    1/2
5         2  2        NA        NA   2       NA    3/3
6         2  3         1         2  NA        2    1/3
7         3  1        NA        NA   1       NA    1/2
8         3  2        NA        NA   2       NA    1/1
9         3  3         1         2  NA        2    1/2
10        4  1        NA        NA   1       NA    1/2
11        4  2        NA        NA   2       NA    1/2
12        4  3         1         2  NA        2    1/1
13        5  1        NA        NA   1       NA    1/2
14        5  2        NA        NA   2       NA    1/2
15        5  3         1         2  NA        2    1/2
16        6  1        NA        NA   1       NA    1/2
```

```
17        6  2        NA        NA   2       NA   <NA>
18        6  3         1         2  NA        2    1/1
19        7  1        NA        NA   1       NA    1/2
20        7  2        NA        NA   2       NA   <NA>
21        7  3         1         2  NA        2    1/3
```

You can now carry out the TDT calculations:

```
> attach(tdtex)

        The following object(s) are masked from popn :

         affected sex

> tdt(marker)

        Transmission/disequilibrium test
Data:        marker

Untransmitted allele frequencies, informative transmissions
and exact P-values

Allele        Frequency     Transmitted   Untransmitted P-value
1                   0.27273 6             2             0.289
2                   0.54545 2             6             0.289
3                   0.09091 1             0             1.000
4                   0.09091 0             1             1.000

Global chi-squared test = 3 on 2 df. Asymptotic P-value = 0.223
```

Here the tdt function has assumed the standard names and coding for the six standard variables in pre-ped files. If different names were given you would have had to supply these.

Check that the computer agrees with your answer to the first part of the exercise.

A real study

First, we will save tdtex to a disk file (seven_trios) for later use, and clear the decks.

```
> save(tdtex, file = "seven_trios")
> clear()
```

The next exercise concerns the same markers discussed in the exercise on population–based studies, but here measured in a family–based study of another disease. The data are stored in the dataframe fmly. First load and attach the new dataframe, and do a TDT test:

```
> data(fmly)
> summary(fmly)

   pedigree              id          id.father          id.mother     
 Length:15591       Min.   : 1.00   Min.   :   1.000   Min.   :   1.000  
 Class  :character   1st Qu.: 2.00   1st Qu.:   1.000   1st Qu.:   2.000  
 Mode   :character   Median : 2.00   Median :   1.000   Median :   2.000  
                    Mean   : 2.65   Mean   :   1.103   Mean   :   1.926  
                    3rd Qu.: 4.00   3rd Qu.:   1.000   3rd Qu.:   2.000  
                    Max.   :18.00   Max.   :  13.000   Max.   :  10.000  
                                    NA's   :7818.000   NA's   :7816.000  
      sex            affected        A                 B        
 Min.   : 1.000   Min.   :1.000   1/1 :5804 (37%)   1/1 :3720 (24%)  
 1st Qu.: 1.000   1st Qu.:1.000   1/2 :6397 (41%)   1/2 :5898 (38%)  
 Median : 1.000   Median :1.000   2/2 :1879 (12%)   2/2 :2676 (17%)  
 Mean   : 1.494   Mean   :1.331   NA's:1511 (10%)   NA's:3297 (21%)  
 3rd Qu.: 2.000   3rd Qu.:2.000                                      
 Max.   : 2.000   Max.   :2.000                                      
 NA's   :35.000                                                      
    C                 D            
 1/1 :5649 (36%)   1/1 :1909 (12%)  
 1/2 :6570 (42%)   2/1 :5581 (36%)  
 2/2 :2076 (13%)   2/2 :4469 (29%)  
 NA's:1296 ( 8%)   NA's:3632 (23%)  

> attach(fmly)
> tdt(A)

        Transmission/disequilibrium test
Data:        A

Untransmitted allele frequencies, informative transmissions
and exact P-values

Allele        Frequency     Transmitted   Untransmitted P-value
2                    0.3821 1821          2081          3.36e-05
```

Don't worry about the warning messages — I said it was a real study!

Because some families have more than one affected offspring, these tests are not strictly valid in the presence of linkage, since the transmissions to two affected siblings are not independent of one another. This independence assumption can be avoided using the `robust` option:

```
> tdt(A, robust = TRUE)

        Transmission/disequilibrium test
Data:        A
```

```
Untransmitted allele frequencies, informative transmissions
and asymptotic P-values

Allele        Frequency     Transmitted   Untransmitted P-value
2                    0.3821 1821          2081          3.20e-05
```

In fact linkage is very weak in this region and its effect is very small. The asymptotic properties of the `robust` test are not as good as the standard test. This author would only advocate use of the `robust` option in the presence of strong linkage, but this would not be universally accepted by all journal editors and reviewers.

As with allele counting methods for analysis of case–control data, the TDT test is most powerful against the alternative model in which effects of alleles are multiplicative. In this case, the ratio of transmitted to untransmitted alleles provides an estimate of the multiplicative effect of each allele, expressed relative to all the remaining alleles. You should calculate the ratio of the number of times allele "2" of marker *A* was transmitted to the number of times it was not and make a note of it to compare with later results.

Case/pseudo-control studies

More complicated, genotype–based analyses may be carried out by considering the transmitted pair of alleles as the "case" and the other three possible pairs of transmitted alleles as "pseudo-controls" in a *matched* case/control study. The tool for transforming the data in this way is the function `pseudocc()`. We shall experiment with it using the data for the seven trios of the first exercise. We shall start by clearing the global environment and restoring the dataframe `tdtex`:

```
> clear()
> load("seven_trios")
> summary(tdtex)

    pedigree       id      id.father    id.mother       sex         affected
 Min.   :1   Min.   :1   Min.   : 1   Min.   : 2   Min.   :1.0   Min.   : 2
 1st Qu.:2   1st Qu.:1   1st Qu.: 1   1st Qu.: 2   1st Qu.:1.0   1st Qu.: 2
 Median :4   Median :2   Median : 1   Median : 2   Median :1.5   Median : 2
 Mean   :4   Mean   :2   Mean   : 1   Mean   : 2   Mean   :1.5   Mean   : 2
 3rd Qu.:6   3rd Qu.:3   3rd Qu.: 1   3rd Qu.: 2   3rd Qu.:2.0   3rd Qu.: 2
 Max.   :7   Max.   :3   Max.   : 1   Max.   : 2   Max.   :2.0   Max.   : 2
                         NA's   :14   NA's   :14   NA's   :7.0   NA's   :14
  marker
 1/1 : 3 (14%)
 1/2 :11 (52%)
 1/3 : 3 (14%)
 3/3 : 1 ( 5%)
 3/4 : 1 ( 5%)
 NA's: 2 (10%)
```

The case-control dataset can be created as follows:

```
> pscc <- pseudocc(marker, data = tdtex)
```

The result, pscc is a new dataframe of cases and pseudo-controls. To inspect its contents:

```
> show(pscc)
```

```
   set cc pedigree id id.father id.mother marker marker.mother marker.father
1    1  1        1  3         1         2    1/3           3/4           1/2
2    1  0        1  3         1         2    1/4           3/4           1/2
3    1  0        1  3         1         2    2/3           3/4           1/2
4    1  0        1  3         1         2    2/4           3/4           1/2
5    2  1        2  3         1         2    1/3           3/3           1/2
6    2  0        2  3         1         2    1/3           3/3           1/2
7    2  0        2  3         1         2    2/3           3/3           1/2
8    2  0        2  3         1         2    2/3           3/3           1/2
9    3  1        3  3         1         2    1/2           1/1           1/2
10   3  0        3  3         1         2    1/2           1/1           1/2
11   3  0        3  3         1         2    1/1           1/1           1/2
12   3  0        3  3         1         2    1/1           1/1           1/2
13   4  1        4  3         1         2    1/1           1/2           1/2
14   4  0        4  3         1         2    1/2           1/2           1/2
15   4  0        4  3         1         2    1/2           1/2           1/2
16   4  0        4  3         1         2    2/2           1/2           1/2
17   5  1        5  3         1         2    1/2           1/2           1/2
18   5  0        5  3         1         2    1/1           1/2           1/2
19   5  0        5  3         1         2    2/2           1/2           1/2
20   5  0        5  3         1         2    1/2           1/2           1/2
```

Has the program produced the case-control sets you would expect?

We'll now go through the same process in order to estimate the effect of the *A* genotype on our larger family-based study:

```
> clear()
> data(fmly)
> pscc.fmly <- pseudocc(A, data = fmly)
> summary(pscc.fmly)
```

```
      set              cc            pedigree           id        
 Min.   :   1   Min.   :0.00   625     :   20   Min.   : 3.000  
 1st Qu.:1191   1st Qu.:0.00   3699    :   16   1st Qu.: 3.000  
 Median :2324   Median :0.00   3776    :   16   Median : 3.000  
 Mean   :2364   Mean   :0.25   1172    :   12   Mean   : 3.618  
 3rd Qu.:3523   3rd Qu.:0.25   1814    :   12   3rd Qu.: 4.000  
 Max.   :4856   Max.   :1.00   2356    :   12   Max.   :18.000  
```

```
                                 (Other):16632
   id.father         id.mother        A                  A.mother
 Min.   : 1.000   Min.   :1.000   1/1:6788 (41%)   1/1:6732 (40%)
 1st Qu.: 1.000   1st Qu.:2.000   1/2:7588 (45%)   1/2:7712 (46%)
 Median : 1.000   Median :2.000   2/2:2344 (14%)   2/2:2276 (14%)
 Mean   : 1.148   Mean   :1.882
 3rd Qu.: 1.000   3rd Qu.:2.000
 Max.   :13.000   Max.   :9.000

 A.father
 1/1:6628 (40%)
 1/2:7896 (47%)
 2/2:2196 (13%)
```

```
> attach(pscc.fmly)
```

We can now fit various relative risk models by use of the conditional logistic regression function `clogit()`. To indicate that the variable `set` labels matched case-control sets we include the term `strata(set)` in the model. For example, the model of multiplicative allelic effects is fitted by

```
> gcontrasts(A) <- "additive"
> clogit(cc ~ A + strata(set))

Call:
clogit(cc ~ A + strata(set))


        coef exp(coef) se(coef)     z       p
A:a:2 -0.133     0.875   0.0321 -4.16 3.2e-05

Likelihood ratio test=17.3  on 1 df, p=3.13e-05  n= 16720
```

In the output from this command, the relative risk parameters are labelled `exp(coef)`. How does this result compare with the transmitted:untransmitted ratio obtained with the TDT?

You could test for deviation from the multiplicative model by adding a dominance effect:

```
> gcontrasts(A) <- "dominance"
> clogit(cc ~ A + strata(set))

Call:
clogit(cc ~ A + strata(set))


           coef exp(coef) se(coef)      z       p
```

```
A:a:2   -0.1431     0.867   0.0342 -4.181 2.9e-05
A:d:1:2  0.0329     1.033   0.0391  0.842 4.0e-01

Likelihood ratio test=18.1  on 2 df, p=0.000121  n= 16720
```

There is no such suggestion!. If you wished to estimate genotype relative risks:

```
> gcontrasts(A) <- "genotype"
> clogit(cc ~ A + strata(set))

Call:
clogit(cc ~ A + strata(set))


       coef exp(coef) se(coef)     z      p
A1/1  0.110     1.116   0.0424  2.60 0.0094
A2/2 -0.176     0.839   0.0601 -2.93 0.0034

Likelihood ratio test=18.1  on 2 df, p=0.000121  n= 16720
```

As with the TDT, in the presence of linkage it is technically incorrect to assume that transmissions in different trios from the same family are independent. The following command forces a "robust" variance estimate for the effects, which does not depend on this assumption

```
> clogit(cc ~ A + strata(set) + cluster(pedigree), method = "approximate")

Call:
clogit(cc ~ A + strata(set) + cluster(pedigree), method = "approximate")


       coef exp(coef) se(coef) robust se     z      p
A1/1  0.110     1.116   0.0424    0.0426  2.59 0.0096
A2/2 -0.176     0.839   0.0601    0.0599 -2.94 0.0033

Likelihood ratio test=18.1  on 2 df, p=0.000121  n= 16720
```

(the method is only approximate in a technical sense not relevant here, but this argument to the function is nevertheless required).

Several loci

It is possible to extend case/pseudo-control analysis with several loci. There are then two ways we can make the case-control study, according to whether or not we require *haplotype phase* to be known for cases and controls.[7] We will first calculate the phased case-control sets:

[7] Another option requires parent-of-origin to be inferred. This results in further loss of information, since this is not possible for all trios, even when phase can be inferred. This is needed for analyses of parent-of-origin effects.

```
> clear()
> data(fmly)
> pscc <- pseudocc(A, C, data = fmly)
> summary(pscc)

      set             cc             pedigree            id        
 Min.   :   1   Min.   :0.0000   3776    :   16   Min.   : 3.000  
 1st Qu.:1232   1st Qu.:0.0000   1172    :   12   1st Qu.: 3.000  
 Median :2364   Median :0.0000   1814    :   12   Median : 3.000  
 Mean   :2378   Mean   :0.2688   2356    :   12   Mean   : 3.629  
 3rd Qu.:3515   3rd Qu.:1.0000   2589    :   12   3rd Qu.: 4.000  
 Max.   :4856   Max.   :1.0000   2590    :   12   Max.   :18.000  
                                 (Other):13034                    
   id.father        id.mother           A.C                A           
 Min.   : 1.000   Min.   :1.000   1:1/1:1:5711 (44%)   1/1:5962 (45%)  
 1st Qu.: 1.000   1st Qu.:2.000   1:1/2:2:2567 (20%)   1/2:2698 (21%)  
 Median : 1.000   Median :2.000   2:2/1:1:2537 (19%)   2/1:2666 (20%)  
 Mean   : 1.145   Mean   :1.876   2:2/2:2:1732 (13%)   2/2:1784 (14%)  
 3rd Qu.: 1.000   3rd Qu.:2.000   1:1/1:2: 124 ( 1%)                   
 Max.   :13.000   Max.   :9.000   1:2/1:1: 119 ( 1%)                   
                                  (Other): 320 ( 2%)                   
   C              A.mother          C.mother          A.father        
 1/1:5813 (44%)   1/1:6402 (49%)   1/1:6228 (48%)   1/1:6232 (48%)  
 1/2:2717 (21%)   1/2:4536 (35%)   1/2:4624 (35%)   1/2:4772 (36%)  
 2/1:2685 (20%)   2/2:2172 (17%)   2/2:2258 (17%)   2/2:2106 (16%)  
 2/2:1895 (14%)                                                       



 C.father      
 1/1:6100 (47%)  
 1/2:4776 (36%)  
 2/2:2234 (17%)  
```

The dataset created now contains a two-locus haplotype variable, `A.C`. We'll first attach the dataframe and examine the the counts of haplotypes for cases and pseudo-controls,

```
> attach(pscc)
> allele.table(A.C)

  1:1   1:2   2:1   2:2 
16867   421   161  8771 
```

Note the extremely strong LD. First let us fit a model for multiplicative haplotype effects:

```
> gcontrasts(A.C) <- "additive"
> clogit(cc ~ A.C + strata(set))

Call:
clogit(cc ~ A.C + strata(set))


            coef exp(coef) se(coef)     z       p
A.C:a:1:2 -0.296     0.744   0.1339 -2.21 2.7e-02
A.C:a:2:2 -0.137     0.872   0.0336 -4.08 4.5e-05
A.C:a:2:1 -0.620     0.538   0.2168 -2.86 4.3e-03

Likelihood ratio test=26  on 3 df, p=9.49e-06  n= 13110
```

You can investigate the effect of using "robust" standard error estimates by adding the `cluster(pedigree)` term to the model as demonstrated above. Next, consider an unrestricted model for genotype relative risks . . .

```
> gcontrasts(A.C) <- "dominance"
> clogit(cc ~ A.C + strata(set))

Call:
clogit(cc ~ A.C + strata(set))


                 coef exp(coef) se(coef)      z       p
A.C:a:1:2     -0.2269     0.797   0.4791 -0.474 0.64000
A.C:a:2:2     -0.1397     0.870   0.0364 -3.840 0.00012
A.C:a:2:1      0.2402     1.272   0.6953  0.346 0.73000
A.C:d:1:2:1:1  0.0702     1.073   0.4981  0.141 0.89000
A.C:d:1:2:2:2 -0.2166     0.805   0.5137 -0.422 0.67000
A.C:d:1:2:2:1 -0.9939     0.370   1.2748 -0.780 0.44000
A.C:d:1:1:2:2  0.0358     1.036   0.0504  0.711 0.48000
A.C:d:1:1:2:1 -0.8579     0.424   0.7371 -1.164 0.24000
A.C:d:2:2:2:1 -1.0676     0.344   0.7880 -1.355 0.18000

Likelihood ratio test=30  on 9 df, p=0.000437  n= 13110
```

There is some suggestion that there is a haplotype effect, or *cis interaction*, since the 2.1 haplotype has a markedly different risk from the other three. A cis interaction would suggest either, if ***A*** and ***C*** are indeed causal that both changes must occur on the same chromosome for the causal effect to be observed, or that neither is causal but are both in LD with the true causal variant

To test whether the effect of the ***A.C*** haplotype is greater than the combined (multiplicative) effects of ***A*** and ***C***, we can carry out the following sequence of commands:

```
> gcontrasts(A) <- "additive"
> gcontrasts(C) <- "additive"
> gcontrasts(A.C) <- "additive"
> clogit(cc ~ A + C + A.C + strata(set))

Call:
clogit(cc ~ A + C + A.C + strata(set))


             coef exp(coef) se(coef)     z      p
A:a:2      -0.620     0.538    0.217 -2.86 0.0043
C:a:2       0.482     1.620    0.217  2.22 0.0270
A.C:a:1:2 -0.778     0.459    0.275 -2.83 0.0047
A.C:a:2:2     NA        NA    0.000    NA     NA
A.C:a:2:1     NA        NA    0.000    NA     NA

Likelihood ratio test=26  on 3 df, p=9.49e-06  n= 13110
```

The haplotype (cis interaction) effect is not significant when the two main effects are in the model. Closer inspection reveals that the *effect* is quite large; it is not significant because the LD is so strong that there is little information about haplotype effects. This is an important lesson when using multiple markers in a region of very strong LD — there is usually little point in considering haplotype effects unless one is interested in very rare variants — and these are only detectable if they have very large effects indeed.

If we only wish to fit the model `A + C`, with alleles of A and C both acting multiplicatively, then phase is irrelevant, since all alleles would act multiplicatively and, for example, the genotype 1.2/2.1 would predict the same risk as 1.1/2.2. phase becomes irelevant.There is then a more efficient way of forming the case-control dataframe. This was originally suggested by Falk and Rubinstein. The idea is to treat the pair of untransmitted alleles at each locus as an unphased genotype. This allows us to use many more trios, and there is no loss in including many markers at the same time:

```
> clear()
> data(fmly)
> pscc <- pseudocc(A, B, C, D, data = fmly, phase = FALSE)
> summary(pscc)

      set             cc          pedigree          id            id.father
 Min.   :   1   Min.   :0.0   3795    :  12   Min.   : 2.000   Min.   : 1.000
 1st Qu.:1215   1st Qu.:0.0   625     :  10   1st Qu.: 3.000   1st Qu.: 1.000
 Median :2428   Median :0.5   3699    :   8   Median : 3.000   Median : 1.000
 Mean   :2428   Mean   :0.5   3776    :   8   Mean   : 3.632   Mean   : 1.163
 3rd Qu.:3642   3rd Qu.:1.0   1172    :   6   3rd Qu.: 4.000   3rd Qu.: 1.000
 Max.   :4856   Max.   :1.0   1304    :   6   Max.   :18.000   Max.   :13.000
                              (Other):9662
```

```
   id.mother          A                  B                  C
 Min.   : 1.000   1/1 :2351 (24%)   1/1 :1786 (18%)   1/1 :2281 (23%)
 1st Qu.: 2.000   1/2 :2658 (27%)   1/2 :2775 (29%)   1/2 :2670 (27%)
 Median : 2.000   2/2 : 801 ( 8%)   2/2 :1249 (13%)   2/2 : 859 ( 9%)
 Mean   : 1.882   NA's:3902 (40%)   NA's:3902 (40%)   NA's:3902 (40%)
 3rd Qu.: 2.000
 Max.   :10.000

    D               A.mother          B.mother          C.mother
 1/1 : 950 (10%)   1/1 :3552 (37%)   1/1 :2494 (26%)   1/1 :3442 (35%)
 2/1 :2723 (28%)   1/2 :4032 (42%)   1/2 :3762 (39%)   1/2 :4192 (43%)
 2/2 :2137 (22%)   2/2 :1242 (13%)   2/2 :1640 (17%)   2/2 :1338 (14%)
 NA's:3902 (40%)   NA's: 886 ( 9%)   NA's:1816 (19%)   NA's: 740 ( 8%)



 D.mother          A.father          B.father          C.father
 1/1 :1228 (13%)   1/1 :3516 (36%)   1/1 :2414 (25%)   1/1 :3432 (35%)
 2/1 :3612 (37%)   1/2 :4162 (43%)   1/2 :3784 (39%)   1/2 :4240 (44%)
 2/2 :2828 (29%)   2/2 :1154 (12%)   2/2 :1650 (17%)   2/2 :1302 (13%)
 NA's:2044 (21%)   NA's: 880 ( 9%)   NA's:1864 (19%)   NA's: 738 ( 8%)



 D.father
 1/1 :1268 (13%)
 2/1 :3662 (38%)
 2/2 :2736 (28%)
 NA's:2046 (21%)
```

The A+C model can now be fitted by:

```
> attach(pscc)
> gcontrasts(A) <- "additive"
> gcontrasts(C) <- "additive"
> clogit(cc ~ A + C + strata(set))

Call:
clogit(cc ~ A + C + strata(set))


         coef exp(coef) se(coef)      z    p
A:a:2 -0.1284     0.879    0.113 -1.134 0.26
C:a:2 -0.0336     0.967    0.112 -0.299 0.77

Likelihood ratio test=17.0  on 2 df, p=0.000201  n=5810 (3902 observations deleted
```

This analysis forms the basis of an efficient method of analysing “tagged” regions or candidate genes, although some extension is necessary to better detect dominance effects at a causal locus. **R**tools for such analyses are under development.

Exercise 7: Parent-of-origin effects

The next exercise studies effects of parent-of-origin of an associated allele. We shall use some data concerning insulin dependent diabetes mellitus (IDDM) and a set of three closely spaced markers in the MHC class 3 region. The data are a very small subset of a much larger study. The markers (bat2, bat3 and ng36 are separated by 20 kb and 260 kb respectively. These cover a region strongly implicated in IDDM by linkage analysis. These data form the dataframe mhc3iddm:

```
> data(mhc3iddm)
> summary(mhc3iddm)

   pedigree            id          id.father     id.mother        sex
 Min.   :  3.00   Min.   :1.00   Min.   :  1   Min.   :  2   Min.   :1.000
 1st Qu.: 45.50   1st Qu.:1.75   1st Qu.:  1   1st Qu.:  2   1st Qu.:1.000
 Median : 89.50   Median :2.50   Median :  1   Median :  2   Median :1.000
 Mean   : 89.47   Mean   :2.50   Mean   :  1   Mean   :  2   Mean   :1.474
 3rd Qu.:134.25   3rd Qu.:3.25   3rd Qu.:  1   3rd Qu.:  2   3rd Qu.:2.000
 Max.   :180.00   Max.   :4.00   Max.   :  1   Max.   :  2   Max.   :2.000
                                 NA's   :192   NA's   :192
   affected     bat2              bat3              ng36
 Min.   :1.0   1/1 : 98 (26%)   1/1 : 78 (20%)   1/1 :198 (52%)
 1st Qu.:1.0   2/1 :179 (47%)   2/1 :171 (45%)   1/2 :126 (33%)
 Median :1.5   2/2 :101 (26%)   2/2 :126 (33%)   2/2 : 47 (12%)
 Mean   :1.5   NA's:  6 ( 2%)   NA's:  9 ( 2%)   NA's: 13 ( 3%)
 3rd Qu.:2.0
 Max.   :2.0

> attach(mhc3iddm)

        The following object(s) are masked from pscc :

         id id.father id.mother pedigree
```

To test for association with the ba2 locus:

```
> tdt(bat2)

        Transmission/disequilibrium test
Data:        bat2

Untransmitted allele frequencies, informative transmissions
and exact P-values

Allele        Frequency     Transmitted   Untransmitted P-value
2                   0.4220 107            66            0.00226
```

Note, however, that these data concern affected sib pairs and there is very strong linkage in the region. This means that transmissions to the two siblings cannot be regarded as independent. To allow for this, you could use the `robust` option with the `tdt` function, e.g.

```
> tdt(bat2, robust = TRUE)

        Transmission/disequilibrium test
Data:        bat2

Untransmitted allele frequencies, informative transmissions
and asymptotic P-values

Allele        Frequency     Transmitted   Untransmitted P-value
2                    0.4220 107           66            0.0174
```

You can look at transmission of maternal and paternal alleles separately using the `tdt` function e.g.

```
> tdt(bat2, parent = "mother")

        Transmission/disequilibrium test
Data:        bat2

Untransmitted allele frequencies, informative transmissions
and exact P-values

Allele        Frequency     Transmitted   Untransmitted P-value
2                    0.4242 42            27            0.0912

> tdt(bat2, parent = "father")

        Transmission/disequilibrium test
Data:        bat2

Untransmitted allele frequencies, informative transmissions
and exact P-values

Allele        Frequency     Transmitted   Untransmitted P-value
2                    0.3975 42            16            0.000862
```

These results are suggestive of a difference. It has been commonplace to test for this by combining the data into a 2×2 table and doing a simple chi-squared test:

```
> tab <- matrix(c(42, 27, 42, 16), nrow = 2)
> tab
```

```
     [,1] [,2]
[1,]   42   42
[2,]   27   16
```

```
> chisq.test(tab)

        Pearson's Chi-squared test with Yates' continuity correction

data:  tab
X-squared = 1.3952, df = 1, p-value = 0.2375
```

although this is not a valid test for reasons explained in the lecture. (In the above sequence, the function `c()` is the *concatenation* function — a general function to group simple objects together. The function `matrix()` establishes the 2×2 structure.)

An alternative analysis is to create a case–control dataset in which the parent of origin of each allele is tracked:

```
> clear()
> data(mhc3iddm)
> ccmhc <- pseudocc(bat2, data = mhc3iddm, parent.of.origin = TRUE)
> summary(ccmhc)
```

```
      set               cc            pedigree            id          id.father
 Min.   :  1.00   Min.   :0.0000   Min.   :  3.00   Min.   :3.000   Min.   :1  
 1st Qu.: 44.00   1st Qu.:0.0000   1st Qu.: 41.00   1st Qu.:3.000   1st Qu.:1  
 Median : 97.00   Median :0.0000   Median : 91.00   Median :4.000   Median :1  
 Mean   : 94.22   Mean   :0.2747   Mean   : 87.35   Mean   :3.505   Mean   :1  
 3rd Qu.:139.00   3rd Qu.:1.0000   3rd Qu.:130.00   3rd Qu.:4.000   3rd Qu.:1  
 Max.   :192.00   Max.   :1.0000   Max.   :180.00   Max.   :4.000   Max.   :1  
   id.mother  bat2           bat2.mother     bat2.father
 Min.   :2   1/1:175 (30%)   1/1:204 (35%)   1/1:220 (38%)  
 1st Qu.:2   1/2:138 (24%)   2/1:218 (37%)   2/1:174 (30%)  
 Median :2   2/1:132 (23%)   2/2:164 (28%)   2/2:192 (33%)  
 Mean   :2   2/2:141 (24%)                                  
 3rd Qu.:2                                                  
 Max.   :2                                                  
```

Note that the summary statistics for `bat2` now distinguish between the 1/2 and 2/1 genotypes. This is because parent of origin is tracked; the program puts the allele inherited from the mother first.[8] We can now carry out a conditional logistic regression analysis, using a new set of contrasts:

```
> attach(ccmhc)
> gcontrasts(bat2) <- "dominance.origin"
> clogit(cc ~ bat2 + strata(set))
```

[8] In the `genetics` package, a genotype variable in which the order of alleles is significant is called a *haplotype* variable. This conflicts with standard usage of the term (as a colection of alleles at linked loci and inherited together on the same chromosome.)

```
Call:
clogit(cc ~ bat2 + strata(set))


              coef exp(coef) se(coef)     z      p
bat2:a:2    0.520     1.682    0.165  3.14 0.0017
bat2:d:2:1 -0.359     0.698    0.257 -1.40 0.1600
bat2:p:2:1 -0.922     0.398    0.519 -1.78 0.0750

Likelihood ratio test=14.2  on 3 df, p=0.00271  n= 586
```

With these contrasts, the three degree of freedom test for different risks between the four types of subject is now broken down into three components

1. the average *additive* effect (a:...),
2. the *dominance* effect (d:...), and
3. the *parent-of-origin* effect (p:...).

The relative risks in the exp(coef) column represent, respectively, the average multiplicative effect of each "2" allele, the average relative risk for heterozygotes as compared with the (geometric) mean risk for the two types of homozygote and, finally, the relative risk for the 2/1 genotype versus the 1/2 genotype.

Interaction between maternal genotype and child's genotype can masquerade as parental origin effects. The following commands investigate interaction between maternal and foetal genotype. Note that we do not require to track parental origin of alleles for this.

```
> clear()
> data(mhc3iddm)
> ccmhc <- pseudocc(bat2, data = mhc3iddm)
> attach(ccmhc)
> gcontrasts(bat2) <- "additive"
> gcontrasts(bat2.mother) <- "additive"
> clogit(cc ~ bat2 + bat2.mother + bat2:bat2.mother + strata(set))

Call:
clogit(cc ~ bat2 + bat2.mother + bat2:bat2.mother + strata(set))


                           coef exp(coef) se(coef)     z    p
bat2:a:2                  0.909     2.482    0.419  2.17 0.03
bat2.mother:a:2              NA        NA    0.000    NA   NA
bat2:a:2:bat2.mother:a:2 -0.435     0.647    0.393 -1.11 0.27

Likelihood ratio test=11.1  on 2 df, p=0.00396  n= 736
```

The interaction is not significant. Note that the main effect of `bat2.mother` could not be fitted. This is because the case and the matched pseudo-controls always have the same mother, so there is no information! Nevertheless this term needs to be in the model, since the regression program doesn't know this.

Counting case-parent trios

An alternative approach to the problem of parent–of–origin effects has been developed by Weinberg and colleagues (*American Journal of Human Genetics*, **65**:229-235, 1998). This involves counting trios of parents and affected offspring. There are 15 types of trio, but only 10 types of trio in which maternal and paternal genotypes are different. An **R**tool to count these types of trio is `trio.types()`. For the `bat2` marker:

```
> clear()
> data(mhc3iddm)
> attach(mhc3iddm)
> trio.types(bat2, first = TRUE, parent.of.origin = TRUE)
```

```
   affected.offspring mother father frequency
1                 1/1    2/1    1/1         8
2                 1/1    1/1    2/1         2
3                 2/1    2/1    1/1         2
4                 2/1    1/1    2/1         7
5                 2/1    2/2    1/1         8
6                 2/1    1/1    2/2         8
7                 2/1    2/2    2/1         2
8                 2/1    2/1    2/2         4
9                 2/2    2/2    2/1         4
10                2/2    2/1    2/2         6
```

Note the use of the `first` option so that only the first affected offspring of any family is used.[9] You will see that the trio types are arranged in five pairs such that the two members of each pair are the same except for reversal of paternal and maternal genotype. Were it not for selection of trios by affected offspring we would expect the two frequencies within a pair to be equal. However, because of the selection, we expect the ratio of frequencies to reflect the ratio of offspring risks. For each pair, how would you expect the ratio of frequencies to be affected by different risks being associated with paternal and maternal copies of the '2' allele? Informally, do the trio counts suggest such an effect?

More formally, we can carry out a test for this using the command `origin()`. This command fits the logistic regression model which predicts the ratio of frequencies of the two trios in each pair, and carries out an analysis of deviance:

[9] The `first` option has been used because Weinberg's method may be misleading if there are multiple affected offspring. Even with this option in force there are difficulties. If families have been ascertained to have at least two affected offspring and there is a parent-of-origin effect, the expected distribution of trios will not be the same as predicted by the simple theory. This should not affect the validity of significance tests, but it will lead to biased estimates of effects.

```
> origin(bat2)

Analysis of Deviance Table

Model: binomial, link: logit

Response: p

Terms added sequentially (first to last)


                Df Deviance Resid. Df Resid. Dev
NULL                               5     7.8791
maternal.origin  1   1.5944        4     6.2848
```

(Note that, by default, `origin()` only uses the first affected offspring in any family).In this case the logistic regression model is a very simple one — it is required only to detect deviation from a 50:50 split in all pairs of trios. However, Weinberg also pointed out that a direct (presumably inter-uterine) effect of maternal genotype on subsequent disease risk in the offspring may also distort the ratio of trio frequencies. This may be allowed for in the logistic regression; the model allows two parameters to model the maternal genotype effect:

```
> origin(bat2, maternal.effect = TRUE)

Analysis of Deviance Table

Model: binomial, link: logit

Response: p

Terms added sequentially (first to last)


                Df Deviance Resid. Df Resid. Dev
NULL                               5     7.8791
mother           2   0.8367        3     7.0424
maternal.origin  1   4.1096        2     2.9328
```

(These commands continue to work with multi-allelic loci, although the degrees of freedom for the test statistics change.)

Extending the case/pseudo–control analysis

Weinberg's approach is more efficient than the case/pseudo–control approach described earlier. Essentially, the additional information is derived at the expense of an additional assumption — that of *exchangeability* of parental genotypes. Formally this means that for a given *mating type*, $a/b + c/d$ say, in the population it is just as likely that the father is a/b and the mother is c/d as if the mother is a/b and the father c/d. However, as described earlier, the likelihood used by this analysis is not correct when, as in the case of our MHC data, families are ascertained on the basis of > 1 affected offspring; parameter estimates will then be biased.

With the same caveat regarding ascertainment, the parental exchangeability assumption may also be incorporated into the creation of a case/pseudo–control study; additional pseudo–controls are created by simply switching the maternal and paternal alleles of the existing case and pseudo–controls. To experiment with this, return to the dataframe you created for the seven-family TDT exercise. If you no longer have this, a version is available as `tdt.solution`. You can create the new case/pseudo-control study as follows:

```
> clear()
> data(tdt.solution)
> pscc <- pseudocc(marker, data = tdt.solution, exchangeable = TRUE)
```

(Note that setting `exchangeable` to `TRUE` also sets `parent.of.origin`.) Look at the resultant dataframe:

```
> show(pscc)
```

```
   set cc pedigree id id.father id.mother marker marker.mother marker.father
1    1  1        1  3         1         2    3/1           3/4           1/2
2    1  0        1  3         1         2    4/1           3/4           1/2
3    1  0        1  3         1         2    3/2           3/4           1/2
4    1  0        1  3         1         2    4/2           3/4           1/2
5    1  0        1  3         2         1    1/3           1/2           3/4
6    1  0        1  3         2         1    1/4           1/2           3/4
7    1  0        1  3         2         1    2/3           1/2           3/4
8    1  0        1  3         2         1    2/4           1/2           3/4
9    2  1        2  3         1         2    3/1           3/3           1/2
10   2  0        2  3         1         2    3/1           3/3           1/2
11   2  0        2  3         1         2    3/2           3/3           1/2
12   2  0        2  3         1         2    3/2           3/3           1/2
13   2  0        2  3         2         1    1/3           1/2           3/3
14   2  0        2  3         2         1    1/3           1/2           3/3
15   2  0        2  3         2         1    2/3           1/2           3/3
16   2  0        2  3         2         1    2/3           1/2           3/3
17   3  1        3  3         1         2    1/2           1/1           1/2
18   3  0        3  3         1         2    1/2           1/1           1/2
19   3  0        3  3         1         2    1/1           1/1           1/2
20   3  0        3  3         1         2    1/1           1/1           1/2
21   3  0        3  3         2         1    2/1           1/2           1/1
22   3  0        3  3         2         1    2/1           1/2           1/1
23   3  0        3  3         2         1    1/1           1/2           1/1
24   3  0        3  3         2         1    1/1           1/2           1/1
```

```
25 4 1     4 3      1      2  1/1      1/2      1/2
26 4 0     4 3      1      2  2/2      1/2      1/2
27 4 0     4 3      2      1  1/1      1/2      1/2
28 4 0     4 3      2      1  2/2      1/2      1/2
```

If you have time, you might like to repeat the analysis for parent-of-origin effects in the `mhc3iddm` data under the parental exchangeability assumption.

Exercise 8: Multiple testing

Multi-phase studies with "stopping for futility"

This exercise explores two situations in which multiple testing is involved. In the first of these, we test many markers (or regions) for association in an initial screen, taking through any which achieve a certain level of statistical significance to be tested in further subjects. Several phases may be considered. Markers which do not look promising in early stages are dropped from further study. In the sequential testing literature, this strategy is termed "stopping for futility".

We shall consider the design of a study in which 5000 cases and 5000 controls are available. We shall first explore the power that would be achieved if all of these cases and controls were used in a single stage design. We shall assume a relative risk of 1.33 for a causal variant with population frequency 20% tagged with $R^2 = 0.8'$ by 10 tag SNPs in the gene. We shall require a significance level of $\alpha = 10^{-6}$:

```
> htPower.cc(df = 10, alpha = 1e-06, P = 0.2, theta = 1.33, R2 = 0.8,
+     N.case = 5000)

$ncp
[1] 56.12091

$power
[1] 0.8990161
```

The power is very close to 90%. Note that the non-centrality parameter is 56.12091 — in a multi-phase study we can choose to allocate how we "spend" this between the difference phases.

We first consider a design in which we use 1000 cases and 1000 controls (20% of the sample) in the first stage, and consider genes for phase 2 only if they achieve $p < 0.1$ in phase 1. Phase 2 will use 2000 cases and 2000 controls (40% of the available resource). Finally, genes will be tested in the remaining 2000 cases and 2000 controls if they achieve $p < 0.01$ in phase 2. At the end of the study we could imagine carrying out tests at nominal signifcance levels 10^{-4}, 10^{-5}, and 10^{-6}. The following command calculates the probabilities of possible outcomes for any single SNP in this design under the null hypothesis (since simulation is used, this can take some time to run)

```
> N.stage(df = 10, ncp = 0, frac = c(0.2, 0.4), alpha.int = c(0.1,
+     0.01), alpha.end = c(1e-04, 1e-05, 1e-06))

$p.final
[1] 3.601374e-05 4.924903e-06 6.190267e-07

$p.interim
[1] 0.100000000 0.004126574

$simulations
[1] 803556  80350   3300
```

The output element p.final represent the probabilities of achieving "significance" at the final three nominal levels. It can be seen that these are appreciably smaller than the nominal levels — if we are to adjust p-values for stopping for futility, the adjustment is *downwards*. This can be controversial, and there are differing opinions on whether one should adapt final significance levels in this manner. The output element p.interim gives the probability that the set of tags considerd will survive in the study beyond each stage. Thus 10% of tags for true negative genes will not be considered beyond phase 1.

To calculate the power against the alternative scenario discussed above, we simply re-run the command with ncp=56.12091 in place of ncp=0. You will find that, with this design, there is quite a substantial loss of power. The secret of a good design is not to lose, early on, findings that will eventually achieve significance. To improve the current design, you might choose to increase the sample size at phase 1, or to take a greater proportion of SNPs through to phase 2. You might like to explore some of these options.

Although this software allows for the tests carried out after each phase to be chi-squared tests with different degrees of freedom, the conclusions are not very sensitive to the df parameter.

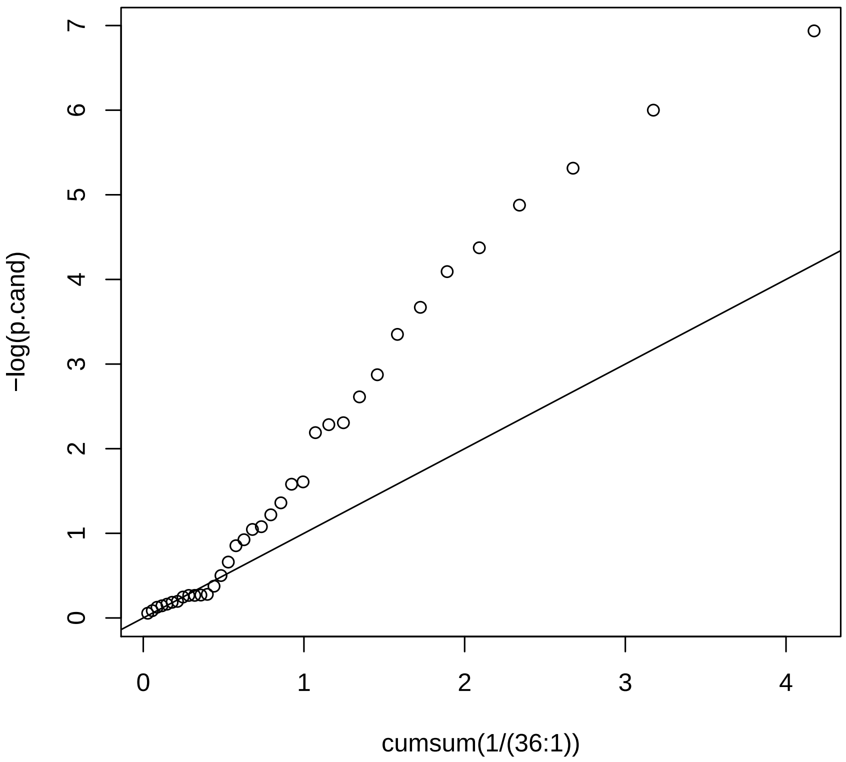
False discovery rates

The next exercise considers the estimation of false discovery rates when one has carried out a succession of tests. It uses the qvalue package written by John Storey of the University of Washington. We will first read in three datasets:

```
> data(pvalues)
> summary(p.cand)

     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
0.0009711 0.0691400 0.3457000 0.4104000 0.7659000 0.9457000

> summary(p.nsSNP)

     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
1.077e-26 2.333e-01 4.946e-01 4.906e-01 7.403e-01 9.998e-01

> summary(p.noHLA)

     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
5.559e-05 2.423e-01 5.007e-01 4.976e-01 7.436e-01 9.998e-01
```

Each of these is a set of p-values. The first set were obtained by testing a sequence of 36 good candidate genes, so it is not unreasonable to expect a rather high proportion of true positives. We will start by comparing the distribution of these p-values with its expectation under the hypothesis of no true positives. Then the p–values would be uniformly distributed, and $-\log p$ would be distributed as exponential variates with mean 1. We'll use the latter result and plot $-\log p$ against the expected values of an ordered sample of size 36 from the exponential(1) distribution[10]. We'll also add a line of slope 1 to the plot to show what we would expect if there were no true posistives:

```
> plot(cumsum(1/(36:1)), -log(p.cand))
> abline(0, 1)
```

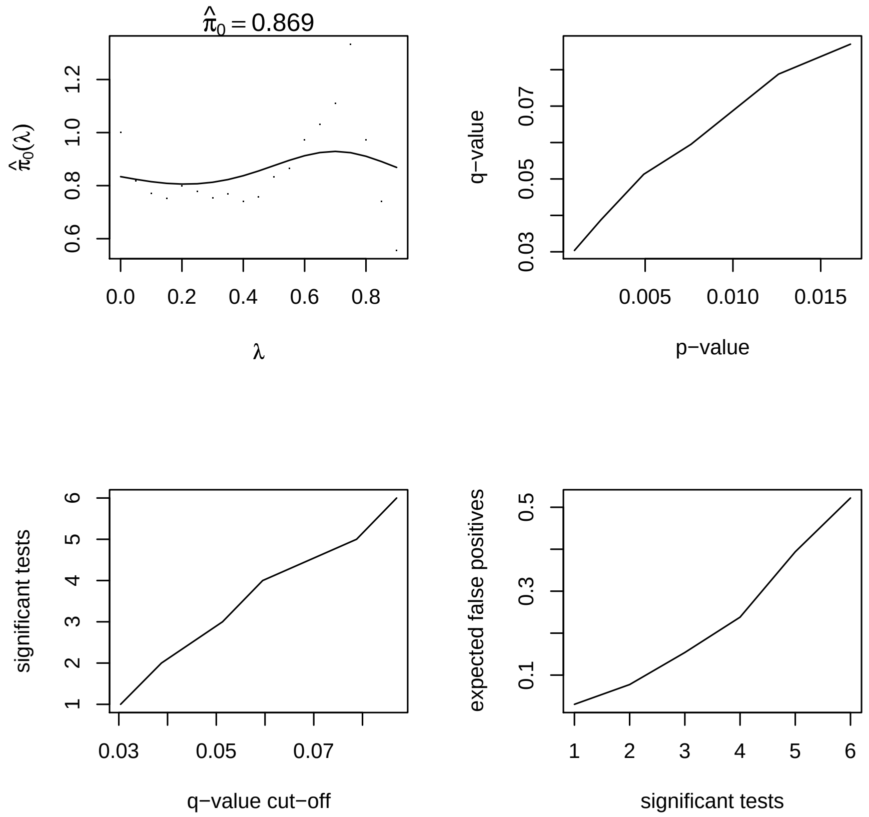


Note that *small* values of p correspond with *large* values for $-\log p$. This plot suggests that there are substantial numbers of true positives, although few achieve p-values that would be considered particularly noteworthy in genetic epidemiology.

The q-value method attemps to estimate the true negative rate (denoted by $\widehat{\pi}_0$) using the observed proportion of p-values that exceed a value λ. As λ is increased, the upward bias on $\widehat{\pi}_0$ is reduced, but the variance of the estimate is increased. The software attempts an optimal trade-off between bias and variance in choosing an appropriate value of λ with which to estimate $\widehat{\pi}_0$. A sample size of 36 is really too small for these procedures, but it is nevertheless interesting to see what happens with these data. The following commands calculate the q-values and plot some useful graphs:

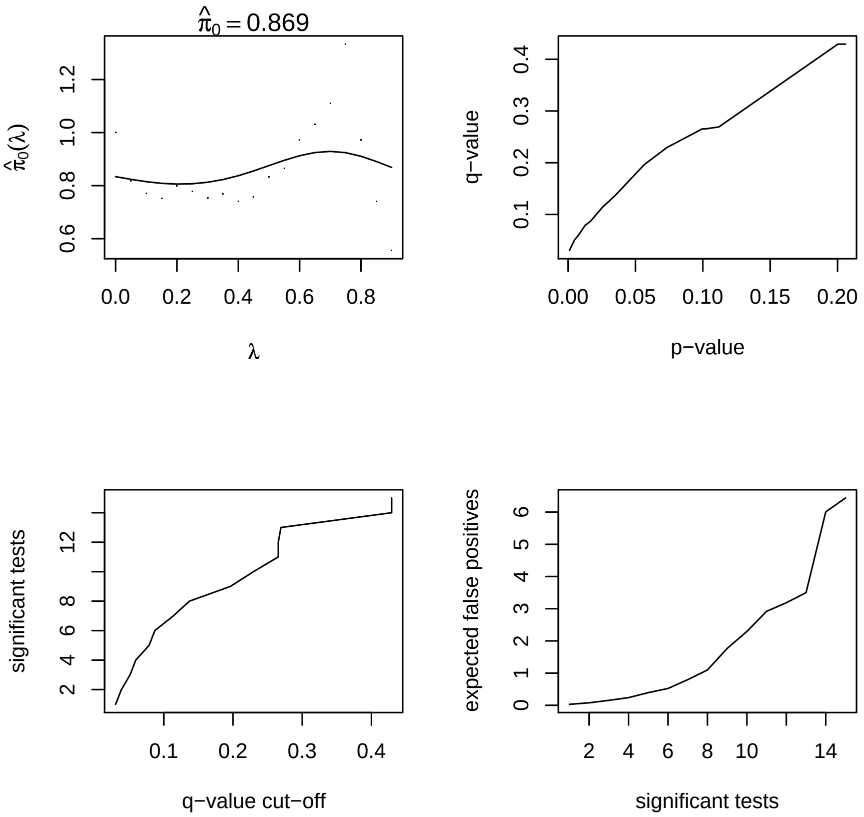
```
> q.cand <- qvalue(p.cand)
> plot(q.cand)
```

[10] For a sample of size N these are given by $\frac{1}{N}$, $\left(\frac{1}{N}+\frac{1}{N-1}\right)$, $\left(\frac{1}{N}+\frac{1}{N-1}+\frac{1}{N-2}\right)$, etc..



The graph in the top left corner shows the estimate of $\widehat{\pi}_0$ as a function of λ, together with the "optimal" estimate. The top right graph plots q-values against p-values. The bottom left graph plots the number of tests which would be selected at different thresholds for the q-value and the bottom right graph estimates the number of these that would be expected to be false positives. The default ranges of values plotted were not too useful here so we'll re-plot them:

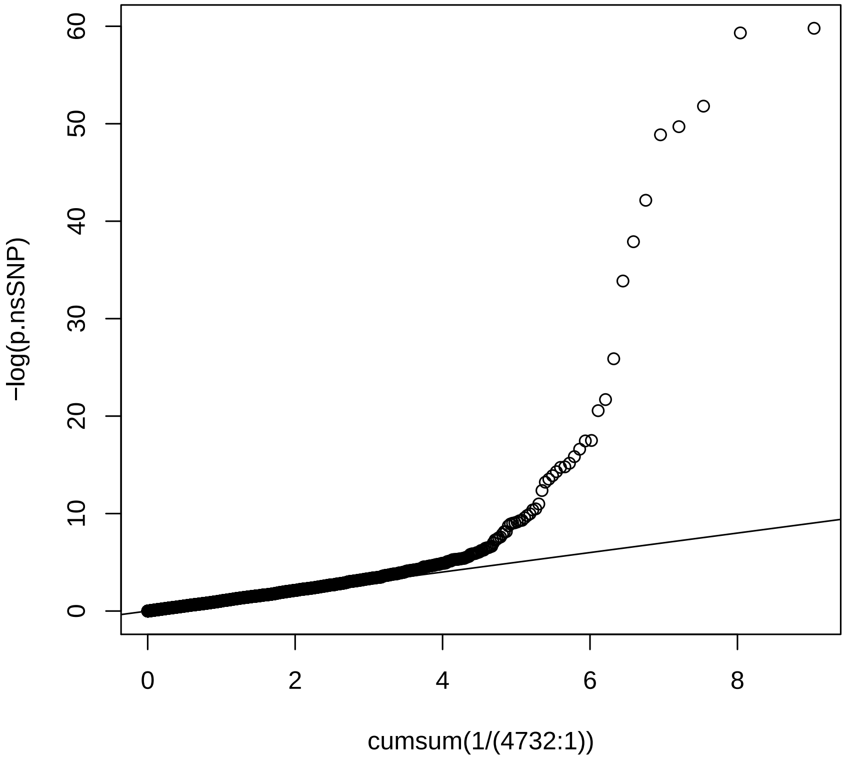
```
> plot(q.cand, rng = c(0, 0.5))
```



This suggests that, out of the 12 most significant results we might expect only 3 of them to be false positives. However, this estimate is dependent on the estimate of the true negative rate, $\widehat{\pi}_0$, and this is unreliable when based on so few points.

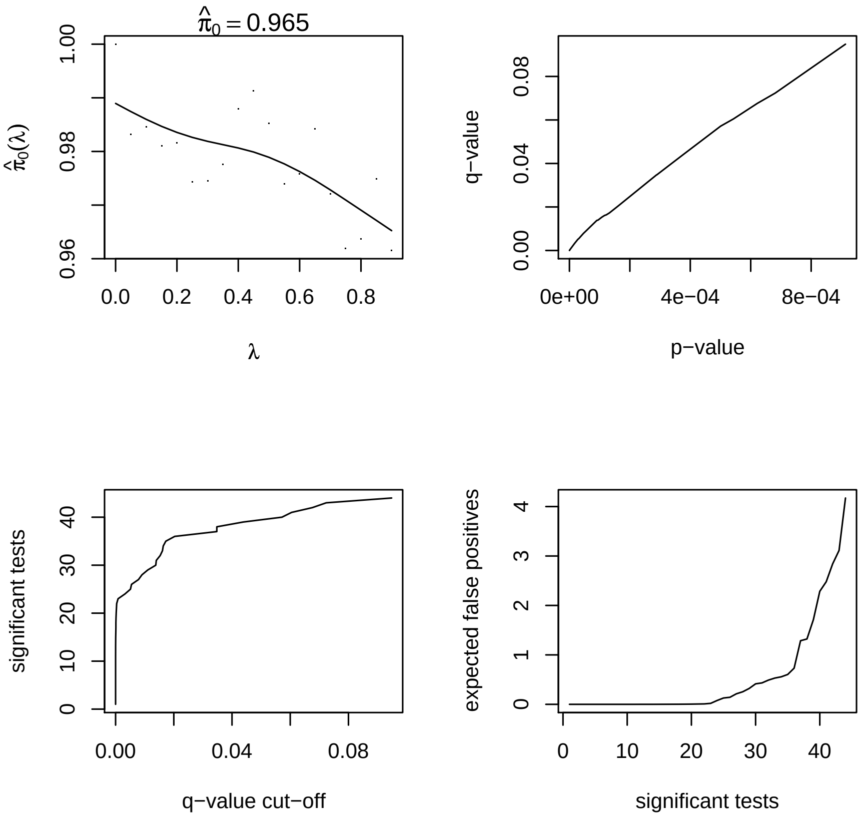
The next example concerns a screen of 4732 non-synonymous SNPs in ~ 800 cases of type 1 diabetes and approximately the same number of controls. These SNPs were not selected to be in candidate genes, so the true positive rate must be expected to be very much smaller in this case. We'll first look at the distribution of $-\log p$ values

```
> plot(cumsum(1/(4732:1)), -log(p.nsSNP))
> abline(0, 1)
```



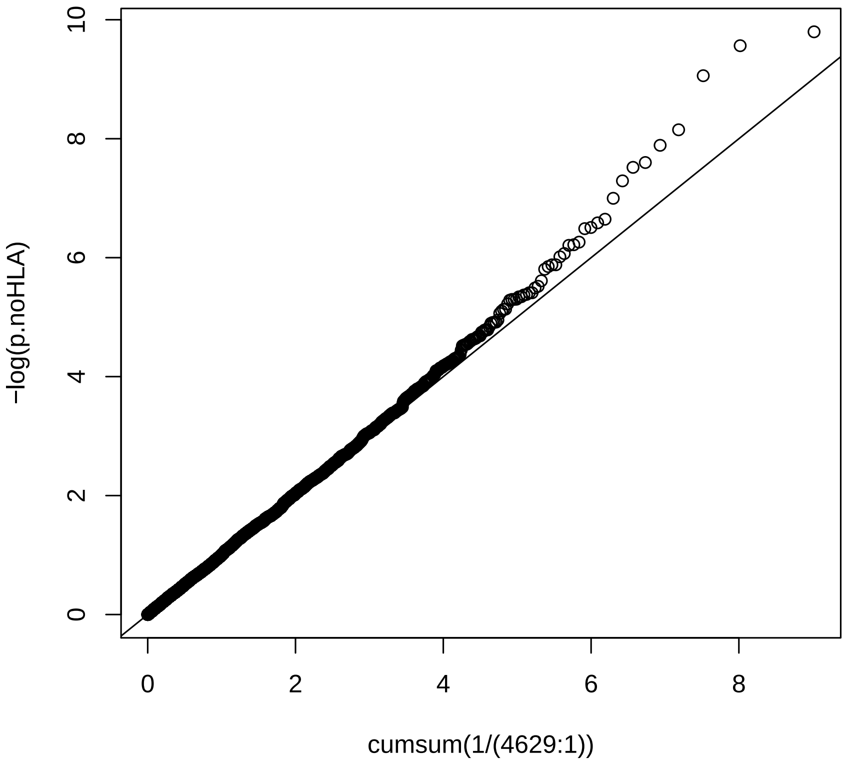
This suggests substantial numbers of true positives, and this is supported by the *q*-value analysis:

```
> q.nsSNP <- qvalue(p.nsSNP)
> plot(q.nsSNP)
```

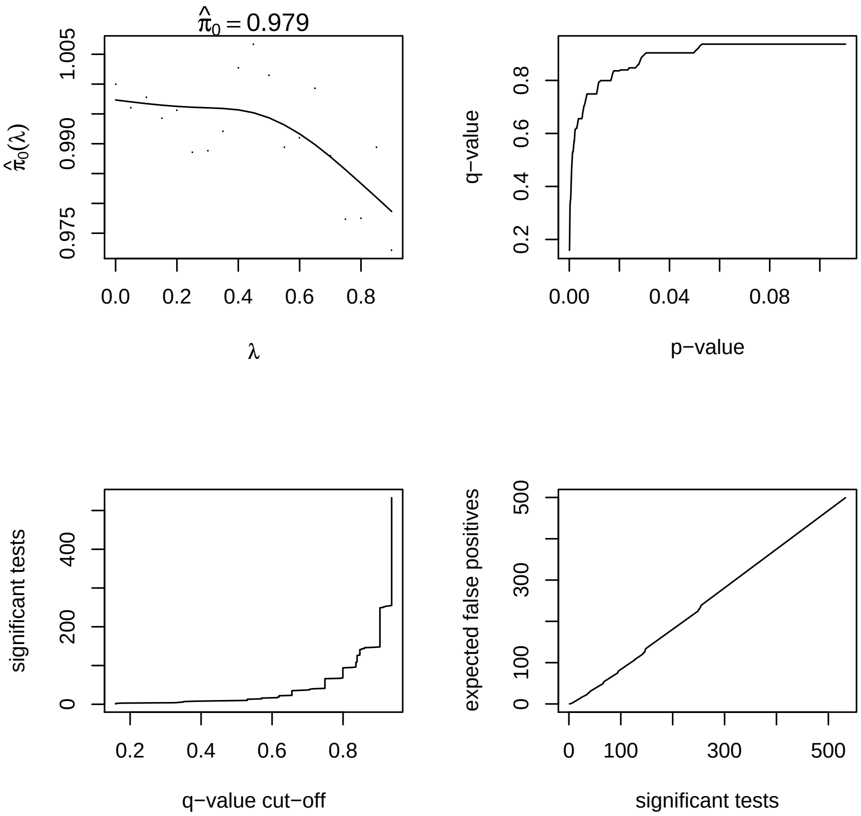


However 103 of these SNPs were in the HLA region on chromosome 6, and there are known to be very strong HLA associations in this region coupled with strong linkage disequilibrium. If we omit these and analyse the remaining 4629 SNPs we get a very different picture:

```
> plot(cumsum(1/(4629:1)), -log(p.noHLA))
> abline(0, 1)
```



```
> q.noHLA <- qvalue(p.noHLA)
> plot(q.noHLA)
```



On the face of it this is rather depressing — over the range of thresholds most "findings" will be false positives. However, this was the first stage of a multi-phase study and we must expect most positives to be false positives. The graphs are not, perhaps, as much use as a tabulation here. The following command allows us to see how many of these tests would pass thresholds which allowed false positive rates of 90% and 95%:

```
> summary(q.noHLA, cut = c(0.9, 0.95))

Call:
qvalue(p = p.noHLA)

pi0:        0.978645

Cumulative number of significant calls:

        <0.9 <0.95
p-value 4179  4397
q-value  147  1399
```