Practical exercise on population–based association studies

David Clayton and Joanna Howson

The data for this practical exercise concern a population–based case–control study of the association between juvenile rheumatoid arthritis (JRA) and the NRAMP1 gene (and a closely linked marker, D2S1471). These data are not the final corrected values generated by the study, but provide a useful illustration of the range of methods available. The analyses to be explored in this practical exercise make considerable use of a `Stata` package, `genassoc`, written by DGC and available from

```
http://www-gene.cimr.cam.ac.uk/clayton/software/stata
```

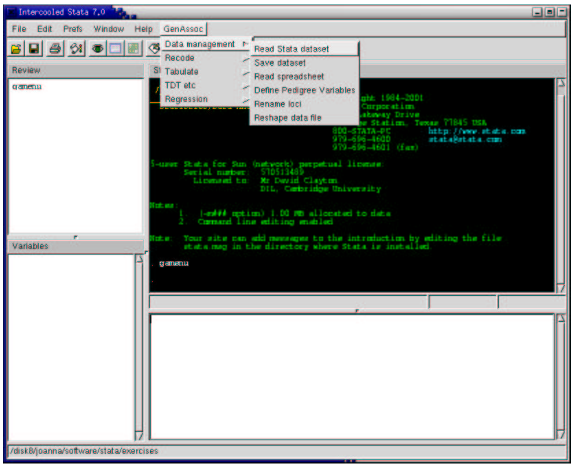
The programs in this package can be run under either a menu based system, and/or a command line. However, standard `Stata` commands must be run from the command line.To activate the menu system,

```
. gamenu
```

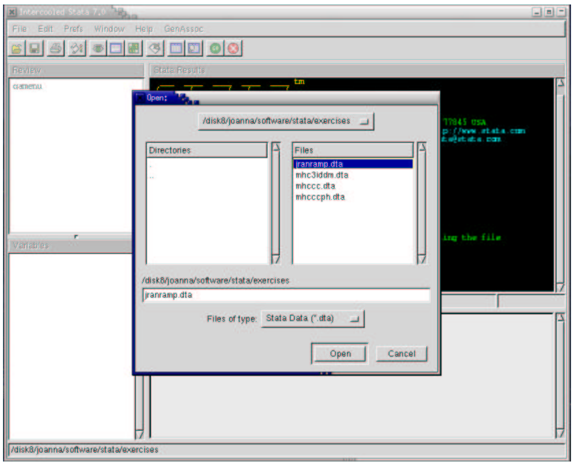
This causes a new drop–down, labelled **Genassoc**, to be added to the standard Stata menu system. We are now ready to start the exercises.

1 Analysis at chromosome level

1. Go to the **GenAssoc** menu at the top of the window, in order to read in the data file `jranramp.dta`. Select the **Read Stata dataset** option from the **Data management** sub menu, as in the illustration below.



```
ght 1984-2001
Corporation
akeway Drive
e Station, Texas 77845 USA
800-STATA-PC          http://www.stata.com
979-696-4600          stata@stata.com
979-696-4601 (fax)

5-user Stata for Sun (network) perpetual license:
       Serial number:  570513489
         Licensed to:  Mr David Clayton
                       DIL, Cambridge University

Notes:
      1.  (-m### option) 1.00 MB allocated to data
      2.  Command line editing enabled

Note:  Your site can add messages to the introduction by editing the file
       stata.msg in the directory where Stata is installed.

. gamenu
```



/disk8/joanna/software/stata/exercises

Files: jranramp.dta, mhc3iddm.dta, mhccc.dta, mhcccph.dta

/disk8/joanna/software/stata/exercises

jranramp.dta

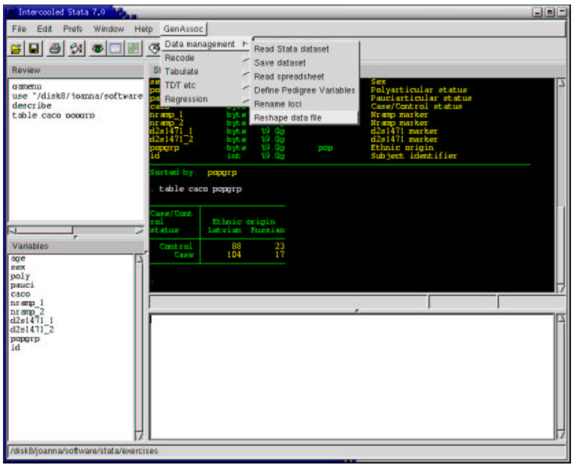
Files of type: Stata Data (*.dta)

The equivalent command line is,

```
. use jranramp
```

You can use the command `describe` to find out what variables are available in this dataset.

2. For this first group of analyses, we shall change the data file from one in which each line describes a person, to one in which each line describes a chromosome. From the **GenAssoc** menu, select the **Data Management** sub menu and the **Reshape data file** option, as illustrated below.

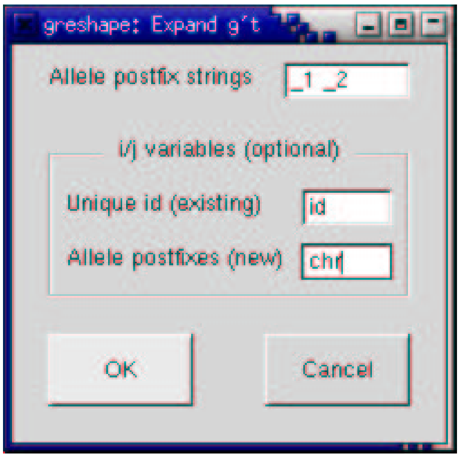


```
gamenu
use "/disk8/joanna/software
describe
table caco popgrp
```

```
nramp_1      byte   %9.0g                Nramp marker
nramp_2      byte   %9.0g                Nramp marker
d2s1471_1    byte   %9.0g                d2s1471 marker
d2s1471_2    byte   %9.0g                d2s1471 marker
popgrp       byte   %9.0g        pop     Ethnic origin
id           int    %9.0g                Subject identifier

Sorted by:  popgrp

. table caco popgrp

Case/Cont
rol        Ethnic origin
status    Latvian  Russian

  Control      88       23
     Case     104       17
```

```
age
sex
poly
pauci
caco
nramp_1
nramp_2
d2s1471_1
d2s1471_2
popgrp
id
```

/disk8/joanna/software/stata/exercises



Allele postfix strings: _1 _2

i/j variables (optional)

Unique id (existing): id

Allele postfixes (new): chr

Enter `id` as the ‘Unique id’, `chr` as ‘Allele postfixes’ and click OK.

The equivalent command line is,

```
.  greshape, id(id) gen(chr)
```

The file has doubled in length, but there is now only one copy of the `nramp` and `d2s1471` variables. Each line of data now refers to a single allele and the allele “postfix” strings (here `_1` and `_2`) have been stored in the variable `chr`. However, note that the *phase* of `nramp` and `d2s1471` is unknown and the postfix labels `_1` and `_2` have been assigned arbitrarily. Thus we cannot treat the two loci in the new file as a *haplotype*.

3. The command

```
. table caco nramp
```

shows the distribution of NRAMP alleles in cases and controls. Does there appear to be association?

4. The D2S1471 marker is a tandem repeat marker, the allele numbers representing increasing sequence lengths. The distribution of alleles is shown by

```
. table caco d2s1471
```

to be clearly bimodal with peaks at alleles 5 and 11 — particularly in controls. Arguably, there may be a case for creating a variable which picks out these two main groups of alleles:

```
. egen d2simp = cut(d2s1471), at(1,8,20)
. table d2simp d2s1471
```

`egen` is short for “extended generate”. This command is used for creating new variables using non-standard functions; `cut` is such a function.

5. Use the `table` command to see if allele frequencies vary between ethnic groups. It is, perhaps, advisable to do this in controls alone, by adding `if caco==0` to the statement.

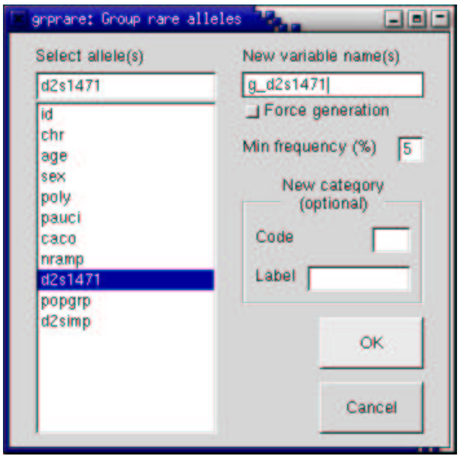
6. The *odds ratio* comparing the 2 and 3 alleles of the NRAMP1 gene is given by:

   ```
   . mhodds caco nramp, co(2,3)
   ```

7. Making comparisons for D2S1471 is harder, because there are so many alleles. We could carry out an overall chi–squared test:

   ```
   . tab caco d2s1471, chi2
   ```

   (note that `tab` is short for `tabulate` — a different command from `table`). But some of the cell frequencies are very small and the chi–squared test is not strictly valid. It could also be argued that this test would “waste” degrees of freedom on testing alleles which are so rare as to be of little interest and/or have little chance of having detectable effects. Commonly the rarer alleles are grouped together, and a command is provided to facilitate this. This can be accessed by from the **Group rare alleles** window, under the **Recode** sub menu of **GenAssoc**, as illustrated below.



grprare: Group rare alleles

Select allele(s): d2s1471

New variable name(s): g_d2s1471

❑ Force generation

Min frequency (%): 5

New category (optional)

Code:

Label:

You may, however, wish to use a shorter the variable name for the recoded locus. By default, this command will group rare alleles until no allele has a relative frequency below 5%, but different thresholds may be selected. The optional parts of the command allow you to control the numerical code and its label given to the new grouped allele category but sensible defaults are chosen.

The command line which does the same thing is

```
. grprare d2s1471, gen(g_d2s1471)
```

You should now re-calculate the chi–squared test.

8. We can test each allele in turn by setting up an "indicator" variable for each allele which indicates its presence or absence. The quick way to do this is

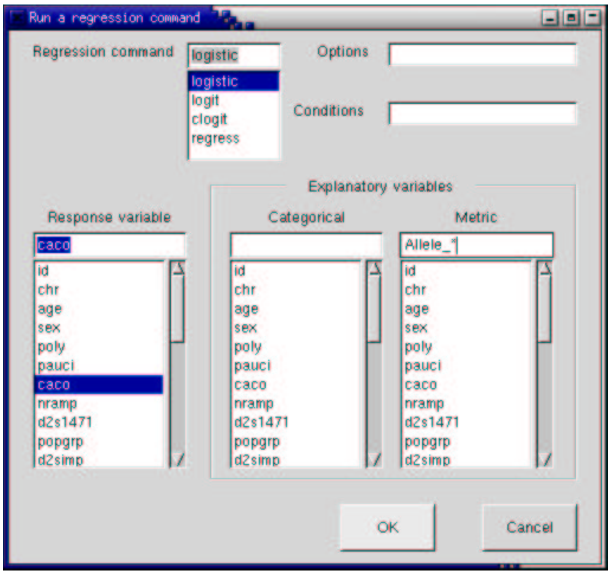
   ```
   . tab d2s1471, gen(Allele_)
   ```

   (or you may prefer to use the new grouped version of the marker). You will see that the command has created a set of new indicator variables `Allele_1, Allele_2, ...` which are 1/0 indicators denoting presence or absence of each allele.Then, to test for different frequencies of allele 1, for example, in cases and controls

   ```
   . tab caco Allele_1, chi2
   ```

   Note, however, there is potentially one test for each allele and correction for multiple testing is complicated by the fact that the tests are not independent of one another.

9. The multiple degree of freedom test can also be carried out using logistic regression on the indicator variables. To do this from the **GenAssoc** menu interface, choose **Fit** from the **Regression** sub menu and complete the window as in the illustration below,



Run a regression command

Regression command: logistic

Options:

Conditions:

Response variable: caco

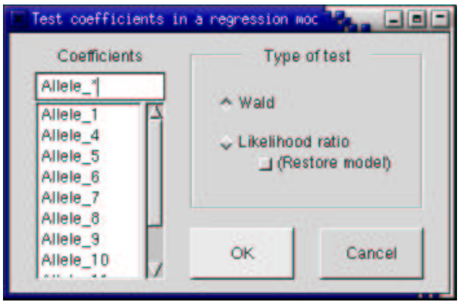
Explanatory variables

Categorical:

Metric: Allele_1

then click OK. The regression model will then be fitted. To carry out the multiple degree of freedom test, select **Test (drop)** from the **Regression** sub menu, choose `Allele_*` as the 'Coefficients', and click the 'Wald' button on, as shown below.

To do the same thing by command lines:



```
. logistic caco Allele_*
. testparm Allele_*
```

This might seem like a rather involved way of calculating the test, but, as we shall see in the next section, the regression method can still be used when we are unwilling to assume Hardy-Weinberg equilibrium.

10. You will have seen that logistic regression also estimates odds ratios. But not every allele can have an odds ratio calculated, since one allele must be chosen as “reference”. The indicator variable for this reference allele is simply dropped from the analysis. The command *logistic* has its own internal logic for doing this, but you may have other ideas. For example, if you wished to take allele 5 (the most frequent allele) as reference,

```
. drop Allele_5
```

    Now complete the regression window as before, or use the command line,

```
. logistic caco Allele_*
```

2 Analysis at person level

Analysis at the chromosome level depends on the assumption of Hardy–Weinberg equilibrium in the population. Under this assumption, the two chromosomes for each individual can be regarded as *independently* sampled from a population of chromosomes. This will also hold for an appropriately chosen control sample. If there is no association between disease and the genetic locus, or if the effect follows the model of multiplicative effects of the two alleles, the two chromosomes in cases can also be assumed to be sampled at random. However, in the presence of association, one or more “high risk” alleles will be more common in cases than in controls. If we analyse at the person level, there is no need to assume Hardy–Weinberg equilibrium — we simply let disease risk depend on the genotypes as observed and do not worry about their frequencies — and the multiplicative model is something that we may or may not choose to assume. Note that some commonly occurring genotyping errors have the effect of disturbing Hardy-Weinberg equilibrium in the observed genotypes.

1. We will start by going back to the original data. So clear the dataset we have been using with,

```
 . clear
```

Select, **Read Stata dataset** from the **Data management** sub menu under **GenAssoc** and select the jranramp.dta file. Alternatively, use the command line,

```
 . use jranramp
```

We may find it useful to recreate the simplified diallelic coding of D2S1471, although we now need to create two new variables:

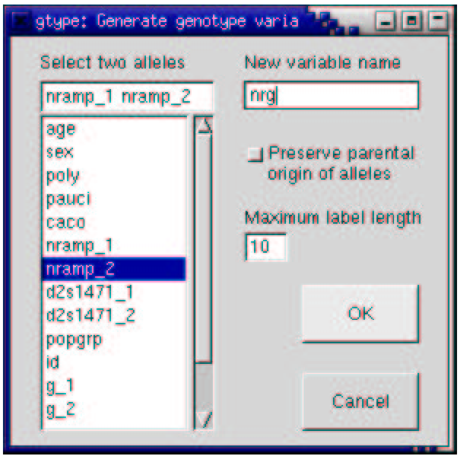
```
. egen d2simp_1 = cut(d2s1471_1), at(1,8,20)
. egen d2simp_2 = cut(d2s1471_2), at(1,8,20)
```

It might also be a good idea to group the rare alleles of this marker. So complete the **Group rare alleles** window, from the **Recode** sub menu under **GenAssoc**, by selecting alleles d2s1471_1 and d2s1471_2, and give the new variables names, such as g_1 and g_2. Alternatively use the command line,

```
. grprare d2s1471_1 d2s1471_2, gen(g_1 g_2)
```

(or, more briefly, `grprare d2s1471_* gen(g_1 g_2)`).

2. The genotype at each locus is now described by two variables. For some analyses it is useful to be able to combine these into a single variable representing the genotype. From the **GenAssoc** menu this can be done by selcting **Create genotype variables** from the **Recode** sub menu:



gtype: Generate genotype varia

Select two alleles: nramp_1 nramp_2

New variable name: nrg

□ Preserve parental origin of alleles

Maximum label length: 10

At the command line prompt, we use the `gtype` function within the Stata `egen` command, as follows:

```
. egen nrg = gtype(nramp_1 nramp_2)
```

Now to tabulate cases and controls by genotype:

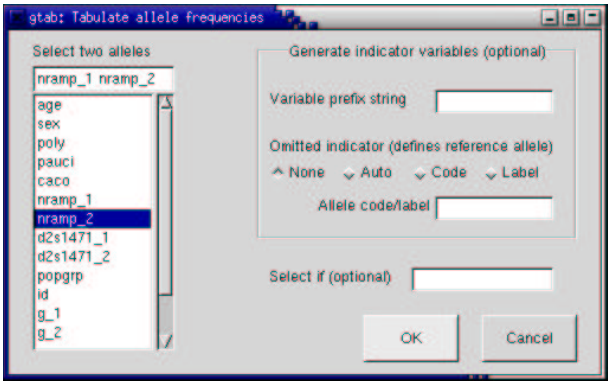
```
. table nrg caco
```

However, when there are many alleles (as with D2S1471), the number of possible genotypes is large and tests have little power.

3. A very useful command is `gtab`, which carries out much the same function as

```
tab ...  , gen(...)
```

in the previous section. This command counts the number of times each allele occurs, as homozygot and as heterozygot, calculates overall allele frequencies, and carries out tests for Hardy–Weinberg equilibrium. Each allele is tested for HWE, but an overall "global" test is also provided. This command can be run from the menu interface as follows: choose **Allele frequencies** from the **Tabulate** sub menu, and select `nramp_1` and `nramp_2` from the variable list,



gtab: Tabulate allele frequencies

Select two alleles: nramp_1 nramp_2

Alternatively, the command line is

```
. gtab nramp*
```

The output from this command suggests that the genotypes are not in Hardy–Weinberg equilibrium (positive *z*–values indicate an excess of homozygots). Perhaps we should check this in controls only. In the **Allele frequencies** window, from the **Tabulate** sub menu, select two alleles, `nramp_1` and `nramp_2` as before but this time fill in the 'Select if (optional)' box with `caco==0`, or

```
. gtab nramp* if caco==0
```

There still seems to be a problem, although this is probably due to genotyping errors rather than population substructure. The command

```
. gtab nramp*, gen(NR_)
```

generates the "indicator" variables `NR_1, NR_2, NR_3` which count the number of times each allele occurs in a genotype (so that each is potentially coded as 0, 1 or 2). The menu alternative is to again choose the **Allele frequencies** menu. Select `nramp_1` and `nramp_2`, now put `NR_` in the 'Variable prefix string' box. Then, to compare the distribution of allele 2 homozygots and heterozygots in cases and controls, and compute a chi–squared test on 2 df, use

```
. tab caco NR_2, chi2
```

4. We can calculate odds ratios in order to compare risks:

   ```
   . mhodds caco NR_2, co(0,1)
   . mhodds caco NR_2, co(1,2)
   ```

5. The above commands show that, although the data are rather sparse for looking at the risk in allele 2 homozygots, they are consistent with the multiplicative model in which the two odds ratios are equal. This model provide a 1 df test which is equivalent to the 1 df test at the chromosome level, but does not assume Hardy–Weinberg equilibrium. To do this with `mhodds`:

   ```
   . mhodds caco NR_2
   ```

   (In general statistical work this test is known as the *Cochran–Armitage* test for trend in proportions.)

6. You can achieve the same results as above by using logistic regression. For the 1 df analysis, complete the **Fit** menu from the **Regression** sub menu. Click `logistic` as the 'Regression command', `caco` as the 'Response variable', and `NR_2` as the 'Metric' explanatory variable. Or use the command,

   ```
   . logistic caco NR_2
   ```

7. For the 2 df analysis we declare `NR_2` as categorical, with `caco` as the response variable and use `logistic` as the regression command. This is done as follows:

   ```
   . xi:logistic caco i.NR_2
   ```

   The `xi:` and i. machinery in the above command is there to signal the fact that `NR_2` is to be treated as a *categorical* variable and causes Stata to automatically create appropriate indicator variables (with names beginning with `_I`).

8. The multiple degree of freedom test based upon all the alleles of D2S1471 can also be carried out by generating indicator variables and using logistic regression. Choose to tabulate **Allele frequencies**. Select `d2s1471_1`, `d2s1471_2` and call the variable prefix, `Allele_`. Click OK to run the command. Select **Fit** from the regression menu to do the regression analysis. Click `logistic` as the command, `caco` as the response variable and `Allele_*` as the *Metric* explanatory variable. Finally complete the test by selecting the **Test (drop)** menu, and fill in the window as before.

   The same sequence of operations is carried out by the command lines:

   ```
   . gtab d2s1471_1 d2s1471_2, gen(Allele_)
   . logistic caco Allele_*
   . testparm Allele_*
   ```

   If you wished to choose, say, allele 5 as reference for calculating odds ratios, you could drop the `Allele_5` indicator before carrying out the regression. Alternatively, there is an option in `gtab` which suppresses generation of an indicator variable for the reference category.)