DADSHARE

A program for examining shared paternity among progeny from microsatellite data.

By Bill Amos
Department of Zoology,
Downing Street,
Cambridge
CB2 3EJ

DadShare is a program written as an Excel macro. It operates by taking maternal genotypes associated with offspring and deducing paternal alleles. Relatedness is then calculated between all pairwise comparisons and a clustering algorithm used to generate a dendrogram which links the most closely related individuals. To interpret the tree, two further analyses are performed. First, Monte Carlo simulation explores the sorts of patterns generated having 1, 2, 3, ... n fathers fathering the n offspring in the file. Second, within each cluster, groups of offspring who are compatible with a single father are determined and marked in colour.

The program is written and annotated in a way that, barring problems from what is surely a rather amateurish programming style, someone used to programming in VBA could edit it and incorporate new features. Alternatively, I am more than happy to entertain and even put into operation good suggestions. Naturally, although I have tested this program reasonably thoroughly, and it appears to work fine, I cannot accept responsibility for any errors it may generate when analysing other peoples' data.

Requirements
DadShare was written in Excel 98 for Macintosh, but should work fine on either Macintosh or PC using Excel 2000. The program assumes the presence of 3 worksheets named Sheet1, Sheet2 and Sheet3, Sheets 1&2 containing data and Sheet3 being available for output. As with all tree-building type programs, time of execution is highly dependent on sample size. I have allowed up to 500 individuals, but this would take a very long time to complete (probably over night on a G3). A dataset around 100 individuals takes around 10 minutes to run and with 300 individuals the time increases to around 50 minutes, most of which is taken by the repetitive simulations. A sample dataset is provided (most offspring have different dads).

Data input

You are prompted for one of two kinds of allele frequency input:
1. Allele frequencies
Allele frequencies are inputted either from raw data (the genotypes of unrelated individuals) or as precalculated frequencies. For 2 loci, appropriate formats are:

a) Raw data

A	A	A	A	A	A
Name1	M	253	255	124	126
Name2	O	253	257	120	124
Name3	O	255	255	124	126
etc.					

Note, every cell in the first row must contain something, from cell A1 to the edge of the data. It does not matter what these cells contain, as long as it is something (the number of loci is determined by counting along row 1 until the first empty cell is found). Column 1 contains names, preferably short, but up to you. Column 2 can contain anything, though for the illustration I have shown it containing the "M" / "O" classification used to indicate mothers and offspring. Missing data are best given as zeros, though blanks work most of the time (I have occasionally encountered files where apparently blank cells seem to have been set immutably to 'text': this will generate an error message). I recommend at least 20 reasonably unrelated individuals, though the precision of the frequency estimates is not critical.

b) Precalculated frequencies.

242	0.01	221	0.85	1	0.4
244	0.12	227	0.15	4	0.1
250	0.21			3	0.3
238	0.12			8	0.2
etc.	etc.				

The first figure, in this case 242, must appear in cell A1. Allele names must be numbers between 1 and 1000, and must not be letters.

To keep array sizes small, all alleles are rescaled such that the shortest is 1, the next shortest 2 etc. This is true for both the methods of data entry. It is useful if the frequencies sum to 1, but if they don't the program automatically ensures that this is so. A potential problem arises through paternal alleles which appear in the experimental data but which do not appear in the data on which allele frequencies are based. The program checks for such alleles and assigns them a nominal frequency of 0.01, assuming the reason they were missed from the frequency data is because they are rare.

2. Data for analysis

Raw data is placed on Sheet2 and should contain genotypes for mothers and their offspring, and should be entered in exactly the same format as illustrated in 1a above, with the proviso that column 2 must now contain "O" or "M" for every individual. Individuals **must** be arranged mother, offspring, offsring . . In other words, the program takes individual 1 (must be a mother) and then considers all following individuals with "O" status to be the offspring of that mother until the next "M" is encountered. Since the end of the files is determined by the first blank cell in column 1, all individuals must have names. The program is written specifically to analyse cases where one parent is known. However, it will also work (though much less informatively) for groups of individuals where neither parent is known. Data for such an analysis should be entered as a 'Mother', code "M" and with every allele scored as '0', followed by the all individuals to be analysed, each of which should have an 'O' code.

What the program does

DadShare starts by loading the frequency data and the family data and rescaling. It then creates a subfile containing deduced paternal alleles for all offspring. An all-against-all matrix of relatedness values is then generated according to the methods of Queller and Goodnight, but using only paternal information. This matrix is then clustered using a simple UPGMA algorithm, sequentially linking the most related pairs of taxa, until a dendrogram is formed. Relatedness values associated with each node are given in the output file on Sheet3. The resulting tree is then searched for clusters of offspring which are compatible with a single father. This search is not exhaustive, in the sense that the search follows strictly the branching order by which the tree was constructed. Compatible groups are indicated by coloured blocks (and all blocks contain numbers to distinguish blocks with very similar colours, or block when output in black and white).

To assist in interpretation of the results, the program then uses Monte Carlo simulation to generate randomised datasets having 1 father for all offspring, 2, 3, 4, and 5 fathers sharing all offspring equally and, at the other end of the scale, all fathers gaining exactly 1, 2, 3, 4, and 5 offspring each. Each of these 10 scenarios is repeated 10 times. In each replicate, the standard clustering is performed and the average branch-tip (i.e. between the previously unlinked taxa) relatedness value noted. The means and standard deviations of these simulations are given in the top left of the output sheet, and may be compared with the observed value (shown above the appropriate column. Maximum relatedness values found in each simulation are also recorded, such that unusually high values can be identified.

The program has been tested on data from seals (9 loci, 150 individuals) and broods of ladybirds (3 highly variable loci, up to 50 individuals) and seems to perform well in terms of helping to provide an objective

analysis of the most likely number of fathers involved. Naturally, a single male gaining one offspring in a system where several other males have already contributed with always be difficult if not impossible to resolve. However, I have found the current version useful and am quite pleased with what it achieves.

If the program proves popular and useful I hope to add features such as scaled branch lengths and bootstrapping to assess the significance of particular nodes. All suggestions are welcome. Good luck!