Molecular Ecology (2003) **12**, 2243–2250 doi: 10.1046/j.1365-294X.2003.01886.x

A graph-theoretic approach to the partition of individuals into full-sib families

JENNIFER BEYER and BERNIE MAY
Genomic Variation Laboratory, Department of Animal Science, University of California, Davis, One Shields Avenue, Davis, CA 95616, USA

Abstract

We present an algorithm to partition a single generation of individuals into full-sib families using single-locus co-dominant marker data. Pairwise likelihood ratios are used to create a graph that represents the full-sib relationships within the data set. Connected-component and minimum-cut algorithms from the graph theory are then employed to find the full-sib families within the graph. The results of a large-scale simulation study show that the algorithm is able to produce accurate partitions when applied to data sets with eight or more loci. Although the algorithm performs best when the distribution of allele frequencies and family sizes in a data set is uniform, the inclusion of more loci or alleles per locus allows accurate partitions to be created from data sets in which these distributions are highly skewed.

Keywords: algorithm, full-sib families, graph theory, pedigree reconstruction, population, relatedness

Received 7 March 2003; revision received 9 April 2003; accepted 9 April 2003

Introduction

Knowledge of the degree of relatedness between individuals can reveal social organization within a population, allow for the study of the evolution of social behaviours, and aid in the prevention of inbreeding in captive breeding programmes. Several methods for the estimation of pairwise relatedness using single-locus genetic marker data have been developed (Queller & Goodnight 1989; Ritland 1996; Lynch & Ritland 1999), in addition to likelihood methods that estimate the probability of a particular relationship between a pair of individuals (Herbinger *et al*. 1997; Goodnight & Queller 1999). There are situations, however, where information concerning pairwise relationships is insufficient, and knowledge of the overall family structure of a population would be more useful. For example, the number of families and their relative sizes would be important for the identification and subsequent management of threatened populations. This information could also be useful in the evaluation of the success of breeding programmes, which strive to maintain genetic diversity in future generations. Finally, the pedigree information could be used in the estimation of population genetic parameters such as heritability and gene flow.

Correspondence: Bernie May. Fax: (530) 752–0175; E-mail: bpmay@ucdavis.edu

Several studies have addressed the problem of identifying full-sib families within a single generation using single-locus marker data. Blouin *et al*. (1996) used a distance measure based on allele sharing as a basis for UPGMA clustering. Visual identification of full-sib and half-sib families was then possible from the resulting dendrogram. A Bayesian approach was proposed by Painter (1997). His algorithm, which requires the choice of an appropriate prior distribution of allele frequencies, groups individuals into full-sib families by maximizing the likelihood of a partition given the data. The exclusion principle formed the basis for an algorithm developed by Almudevar & Field (1999), which does not require the use of population allele frequencies. Markov chain Monte Carlo algorithms have also been used to detect full-sib families. Such algorithms developed by Thomas & Hill (2000) and Smith *et al*. (2001) eliminated the need to examine all plausible partitions. Thomas & Hill (2002) recently extended their algorithm to detect full-sib families nested within half-sib families.

Here, a novel graph-theoretic approach is used to partition a single generation of individuals into full-sib families. The algorithm uses pairwise likelihood ratios calculated from equations developed by Goodnight & Queller (1999) to create a graph that represents the full-sib relationships within the data set. Existing graph algorithms, including connected-component and minimum-cut algorithms, are then used to find and correct the full-sib families in the

partition as needed. A simulation study is conducted to test the algorithm for its sensitivity to the number of loci, number of alleles, family distribution, and allele frequency distribution. While we introduce this algorithm for the simple case of full-sib family identification, the approach lends itself to further extension through simple modifications, which are discussed, and also through the exploitation of the vast collection of graph algorithms that may have applications to the problem of pedigree reconstruction.

Methods

Pairwise likelihoods

Input to the algorithm is a list of individuals and their genotypes at each locus. Individuals are assumed to belong to a single generation, and genotypes are assumed to be from neutral, unlinked, co-dominant markers. Using these data, pairwise likelihood values are calculated as described by Goodnight & Queller (1999). For each pair of individuals in the data set, the likelihood of observing the pair's genotypes if they are truly full-sibs, L_{FS}, is calculated, as is the likelihood for the hypothesis that the pair is unrelated, L_{UR}. A likelihood ratio, L_{FS}/L_{UR}, is then calculated for each pair. Likelihood ratios are also calculated for 1000 simulated full-sib pairs and 1000 simulated unrelated pairs. The ratios for the simulated pairs are used to establish *P*-values for significance, again following the method of Goodnight & Queller (1999).

Full-sib graphs

A graph is used to represent the full-sib relationships within the data set. The graph contains a node for each individual in the data set, with an edge connecting each pair of nodes that has a likelihood ratio greater than the ratio at the $\alpha = 0.05$ significance level. The graph may contain subgraphs that are not connected to one another. Such subgraphs are called connected components. Figure 1 shows a graph with two connected components. Ideally, nodes representing individuals within a full-sib family would have edges connecting them to each other, but would have no edges connecting them to nodes that belong to other families. Thus, each connected component would represent a full-sib family. Furthermore, because each individual in a family is a full-sib to every other individual in the family, each node in an ideal graph of a full-sib family would have an edge connecting it to every other node in the family. A graph that is connected in such a way is called a complete graph. The connected component in Fig. 1 that has four nodes is complete. A component can be scored as to how well it approximates a true family by calculating the ratio between the number of edges in the component and the number of edges in a complete graph with the same number of nodes. Higher scores provide stronger evidence that the component represents a true family.

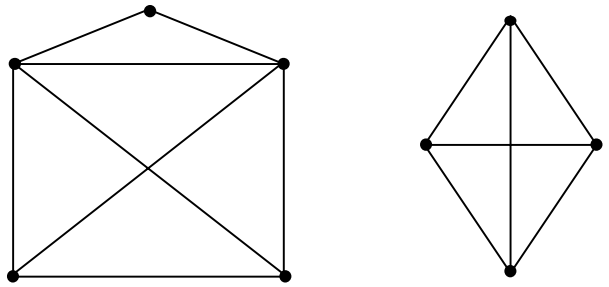

Fig. 1 One graph with two connected components. The component with four nodes is a complete subgraph.

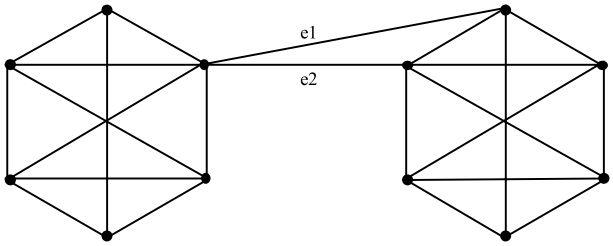

Fig. 2 A graph with two linking edges, labelled e1 and e2.

A pair of individuals may have a high likelihood ratio through chance even though the individuals are not full-sibs. When this occurs, the graph contains edges that connect different families. The presence of these edges, here called linking edges, results in components that represent more than one family. An example of such a graph with two linking edges is shown in Fig. 2. Components of this type are usually far from complete and can be identified by their low scores. The partitioning algorithm uses minimum cuts to find and remove edges connecting pairs of individuals that are not full-sibs. A minimum cut in a graph is the smallest set of edges that must be removed to disconnect the graph such that there is no path connecting a given pair of nodes. A minimum cut can be calculated for every pair of nodes in a graph, and this information can be stored in a data structure called a GH cut tree (Gomory & Hu 1961). A set of linking edges is likely to be a minimum cut because the number of linking edges is expected to be small compared to the number of edges connecting individuals belonging to the same family. Furthermore, the removal of a set of linking edges will result in two subgraphs with much higher scores than the original component. Each of these high scoring subgraphs more closely approximates a true family than does the original component. Therefore, a search of the GH cut tree for a cut whose removal produces high scoring subgraphs should locate the linking edges within a component.

The algorithm

The individuals in a data set are partitioned into full-sib families by following a simple algorithm that begins by creating a graph based on pairwise likelihood ratios. Potential families are identified using an algorithm that finds the connected components in the graph. These components are then analysed for the presence of edges that link more than one family together. Any linking edges that were found are removed from the graph and the resulting components are output as the full-sib family partition. A detailed outline of the algorithm is given below, and the C ++ source code is available at http://genome-lab.ucdavis.edu/People/JenBeyer/familyFinder.html.

Algorithm PARTITION.
Input: genotypes of each individual in the data set.
Output: a list of the full-sib families in the data set.

1 Calculate L_{FS}/L_{UR} likelihood ratios for all pairs of individuals in the data set.
2 Build a graph representing the full-sib relationships found in the previous step.
3 Find the connected components in the graph and store them in a queue.
4 While the queue is not empty do
5 begin
6 Remove a component from the queue and calculate its score.
7 Build a GH cut tree for the component.
8 For each cut with less than 1/3 the total number of edges in the component do
9 begin
10 Score the components that would result if the cut's edges were removed.
11 If the scores are the best found so far, then store them.
12 end
13 If the best scores found are higher than the score for the original component, then separate the families and put them in the queue for further analysis.
14 Otherwise save the original component as a result family.
15 end
16 Output the resulting family partition.

Simulations

Three sets of simulations were conducted to test the algorithm's ability to create accurate partitions. Simulation set A was designed to be similar to one of the simulation sets used by Smith *et al*. (2001) to test their algorithm. Data sets contained 50 individuals from five full-sib families. Four family distributions were used with family sizes for each of the distributions set at (10, 10, 10, 10, 10), (20, 10, 10, 5, 5), (30, 5, 5, 5, 5) and (40, 5, 2, 2, 1), respectively. To determine the required number of loci and alleles per locus, each distribution was tested with four, eight, 12 and 16 loci; each with four, eight, 12 and 16 alleles per locus. Additionally, separate trials were conducted with equally frequent alleles and nonequifrequent alleles. The allele frequency settings for tests using nonequifrequent alleles are given in Table 1. One hundred simulated data sets were run with each combination of parameters.

Table 1 Allele frequency settings for simulations with nonequifrequent alleles

No. of alleles	Frequencies
4	0.4, 0.3, 0.2, 0.1
8	2 alleles at 0.3, 2 alleles at 0.1, 4 alleles at 0.05
12	2 alleles at 0.2, 2 alleles at 0.1, 8 alleles at 0.05
16	2 alleles at 0.2, 2 alleles at 0.1, 4 alleles at 0.05, 8 alleles at 0.025

Simulation set B was designed to test the performance of the algorithm on large data sets. All parameters were identical to those used in simulation set A except each data set contained 500 individuals, with each family containing 10 times as many siblings as in set A. As a result of time considerations, 50 simulated data sets were run with each combination of parameters.

A third set of simulations, set C, was conducted to evaluate the effect of the number of families per data set on the algorithm's ability to create a correct partition. The number of families was varied between one and 10, with 100 individuals divided as equally as possible among the families. Tests were conducted using eight loci with four, eight, 12 and 16 equifrequent alleles per locus. As in simulation set A, 100 simulated data sets were run with each combination of parameters.

Result partitions were scored by counting the number of individuals who had to be moved to a different family to create a correct partition. The number of families in each partition was also noted. After all simulations in a set were completed, the percentage of correct partitions and the average partition score were calculated for each parameter setting, as were the average number of families reported and the range for the number of families.

Results

Simulation set A — factors affecting accuracy of results

The results of simulation set A are presented in Table 2. Accuracy of the algorithm was measured as the percentage of correctly classified data sets among the 100 trials at each

Table 2 Percentage of correctly classified data sets among the 100 trials in simulation set A. In parentheses is the average number of individuals who had to be moved to another family to create a correct classification. Results are shown for four family distributions, with equifrequent and nonequifrequent alleles. Accuracy of 80% or higher is marked with an asterisk

Equifrequent alleles					Non-equifrequent alleles				
	alleles per locus					alleles per locus			
	4	8	12	16		4	8	12	16
loci	—	—	—	—	loci	—	—	—	—
4	0 (25.1)	1 (8.09)	10 (5.46)	19 (4.94)	4	0 (25.8)	0 (15.5)	7 (6.28)	6 (7.21)
8	13 (6.96)	90 (0.28)*	97 (0.07)*	99 (0.01)*	8	3 (8.73)	34 (2.65)	94 (0.06)*	96 (0.04)*
12	59 (1.37)	100 (0.00)*	100 (0.00)*	100 (0.00)*	12	39 (2.52)	96 (0.05)*	100 (0.00)*	100 (0.00)*
16	90 (0.42)*	100 (0.00)*	100 (0.00)*	100 (0.00)*	16	83 (0.39)*	100 (0.00)*	100 (0.00)*	100 (0.00)*
	family sizes: 10, 10, 10, 10, 10					family sizes: 10, 10, 10, 10, 10			
	alleles per locus					alleles per locus			
	4	8	12	16		4	8	12	16
loci	—	—	—	—	loci	—	—	—	—
4	0 (27.0)	0 (10.2)	4 (6.94)	17 (4.80)	4	0 (27.7)	0 (17.9)	1 (8.70)	2 (8.26)
8	7 (8.01)	67 (0.76)	90 (0.16)*	96 (0.04)*	8	1 (12.3)	36 (2.78)	82 (0.42)*	84 (0.32)*
12	56 (1.78)	98 (0.05)*	100 (0.00)*	99 (0.05)*	12	35 (2.61)	83 (0.22)*	100 (0.00)*	100 (0.00)*
16	81 (0.46)*	100 (0.00)*	100 (0.00)*	100 (0.00)*	16	69 (0.78)	98 (0.03)*	100 (0.00)*	100 (0.00)*
	family sizes: 20, 10, 10, 5, 5					family sizes: 20, 10, 10, 5, 5			
	alleles per locus					alleles per locus			
	4	8	12	16		4	8	12	16
loci	—	—	—	—	loci	—	—	—	—
4	0 (31.1)	5 (14.3)	9 (7.65)	22 (4.31)	4	0 (32.5)	0 (25.0)	3 (13.4)	6 (11.6)
8	2 (10.8)	58 (0.77)	85 (0.21)*	82 (0.30)*	8	3 (15.3)	32 (3.90)	70 (0.53)	67 (0.53)
12	45 (2.15)	95 (0.09)*	100 (0.00)*	100 (0.00)*	12	23 (4.67)	76 (0.48)	96 (0.08)*	96 (0.06)*
16	77 (0.51)	99 (0.01)*	100 (0.00)*	100 (0.00)*	16	53 (1.26)	95 (0.13)*	100 (0.00)*	100 (0.00)*
	family sizes: 30, 5, 5, 5, 5					family sizes: 30, 5, 5, 5, 5			
	alleles per locus					alleles per locus			
	4	8	12	16		4	8	12	16
loci	—	—	—	—	loci	—	—	—	—
4	0 (36.5)	2 (24.4)	10 (15.7)	12 (12.2)	4	0 (37.9)	0 (32.0)	0 (21.9)	5 (21.7)
8	5 (21.3)	33 (2.26)	55 (0.61)	54 (0.52)	8	1 (24.4)	18 (8.37)	45 (0.84)	43 (1.29)
12	26 (5.49)	64 (0.52)	76 (0.24)	79 (0.21)	12	13 (7.31)	40 (1.73)	67 (0.39)	66 (0.38)
16	45 (1.05)	77 (0.27)	90 (0.10)*	89 (0.13)*	16	25 (2.67)	59 (0.49)	83 (0.19)*	92 (0.08)*
	family sizes: 40, 5, 2, 2, 1					family sizes: 40, 5, 2, 2, 1			

parameter setting. The distribution of family sizes within a data set had a strong effect on accuracy, with the algorithm performing best when the sizes of families in the data set were equal. Accuracy levels of 90% or higher were observed when families of equal size were tested using eight loci with eight or more alleles, and 100% accuracy was observed with data sets containing 12 or more loci with eight or more alleles. Accuracy decreased as the family size distribution became increasingly skewed by the presence of a large dominating family. This family distribution effect is illustrated in Fig. 3. Accuracy values are plotted for simulations with each of the four family distributions as observed with eight loci and varying numbers of equally frequent alleles. The fourth distribution, with family sizes of 40, 5, 2, 2 and 1, produced particularly poor results when tested with only eight loci, reaching a peak accuracy of only 55%. Despite this trend of decreasing accuracy with increasingly skewed family distributions, accuracy levels greater than 80% were observed using eight loci and the third family distribution, which contained families of size 30, 5, 5, 5 and 5.

Allele frequency distributions also affected the algorithm's accuracy, with better performance observed when using equally frequent alleles. The performance difference between equifrequent and nonequifrequent alleles increased as family distributions became increasingly skewed, with the fourth distribution showing the largest decrease in performance when tested with nonequifrequent alleles. While decreases in accuracy of greater than 20% were observed, this occurred most frequently for parameter settings in which the algorithm did not perform well even with equally frequent alleles, such as with the highly skewed fourth family distribution and with data sets containing relatively few loci or alleles. With the first three family distributions, accuracy decreased by less than 5% when tested using 12 loci with 12 or more alleles and 16 loci with eight or more alleles. In the case of families of equal size, the decrease in accuracy was also less than 5% when using eight loci with 12 or more alleles.

In addition to family and frequency distribution, accuracy was affected by the number of loci and alleles in the data sets. In general, higher numbers of loci and alleles produced better results. However, a low number of loci could be offset by a high number of alleles, and similarly, a low number of alleles could be offset by a high number of loci. Despite this trade-off, the use of only four loci was clearly insufficient, with less than 23% accuracy for all tests. With as few as eight loci, however, accuracy of 90% or higher was observed using eight or more equifrequent alleles and an equal family distribution, as well as when using 12 or more equifrequent alleles and the second family distribution, which contained families of size 20, 10, 10, 5 and 5. With 12 or more loci and eight or more equifrequent alleles, accuracy of 95% or higher was observed with all but the extreme case of the fourth family distribution.

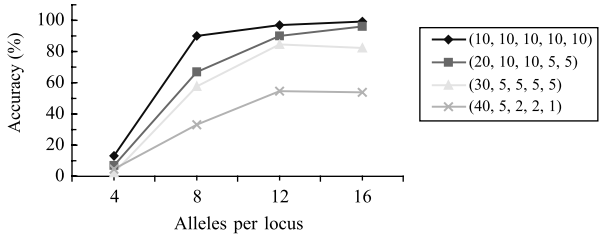

Fig. 3 Effect of family size distribution on accuracy. Results are shown for tests using eight loci, with four, eight, 12 and 16 equifrequent alleles per locus. The key gives the sizes of the families in the four distributions that were tested.

In addition to accuracy, Table 2 presents a second measure of performance for each test in simulation set A by noting in parentheses the average number of individuals who must be moved to a different family to create a correct partition. This value was very low in many situations, even in cases where the overall accuracy was quite poor. For example, with eight loci, eight equifrequent alleles, and a family distribution of 20, 10, 10, 5 and 5 individuals per family, accuracy was 67%, while the average number of individuals who were assigned to the wrong family was only 0.76 (1.52%). Similar scores, where the overall accuracy was less than 80% yet the average number of incorrectly classified individuals was less than 1.0 (2%), were observed with 17 different parameter settings. Thus, in cases where a perfect result partition is relatively unlikely, nearly perfect partitions are often produced.

The average number of families per result partition for each combination of parameters is shown in Table 3, along with the range of values observed. The range values reveal a tendency of the algorithm to overestimate the number of families in a data set when too little information is provided, as was the case with data sets containing only four loci. The average number of families per result partition deviated by less than 1.0 from the correct value of five for tests with each family distribution when using eight loci with eight or more equifrequent alleles or 12 or more loci with four or more equifrequent alleles, with only one exception. These consistent values across each of the family distributions indicate that the family distribution effect is negligible when one is interested in the number of families in a data set.

Simulation set B — performance with large data sets

Simulation set B, which tested data sets containing 500 individuals, produced results comparable to those observed in simulation set A, with a few exceptions. Increases in accuracy of up to 70% were observed for tests using four loci, although 16 alleles were required to produce accuracy levels higher than 80%. Additionally, the larger data sets

Table 3 Average number of families observed for the 100 trials in simulation set A. The range for the number of families in each set of trials is given in parentheses. Results are shown for four family distributions, with equifrequent and nonequifrequent alleles

Equifrequent alleles					Non-equifrequent alleles				
	alleles per locus					alleles per locus			
	4	8	12	16		4	8	12	16
loci	—	—	—	—	loci	—	—	—	—
4	21.8 (10,35)	8.64 (5,16)	7.35 (5,12)	7.04 (5,11)	4	22.9 (8,43)	13.8 (5,29)	7.58 (5,13)	8.04 (5,15)
8	7.92 (5,15)	5.11 (5,7)	5.01 (5,6)	5.01 (5,6)	8	9.6 (5,21)	6.00 (5,12)	5.05 (5,6)	5.03 (5,6)
12	5.48 (4,10)	5.00 (5,5)	5.00 (5,5)	5.00 (5,5)	12	6.06 (5,13)	5.03 (5,6)	5.00 (5,5)	5.00 (5,5)
16	5.03 (4,6)	5.00 (5,5)	5.00 (5,5)	5.00 (5,5)	16	5.12 (4,7)	5.00 (5,5)	5.00 (5,5)	5.00 (5,5)
	family sizes: 10, 10, 10, 10, 10					family sizes: 10, 10, 10, 10, 10			
	alleles per locus					alleles per locus			
	4	8	12	16		4	8	12	16
loci	—	—	—	—	loci	—	—	—	—
4	22.2 (7,42)	9.04 (5,19)	8.03 (5,12)	6.96 (5,12)	4	22.6 (9,36)	14.4 (6,25)	8.93 (5,19)	8.42 (5,18)
8	8.01 (4,18)	5.33 (4,7)	5.12 (5,7)	5.04 (5,6)	8	10.4 (5,21)	6.30 (5,14)	5.18 (4,7)	5.11 (4,7)
12	5.56 (4,11)	5.01 (5,6)	5.00 (5,5)	4.99 (4,5)	12	5.99 (4,13)	5.14 (5,6)	5.00 (5,5)	5.00 (5,5)
16	5.12 (4,8)	5.00 (5,5)	5.00 (5,5)	5.00 (5,5)	16	5.29 (5,8)	5.02 (5,6)	5.00 (5,5)	5.00 (5,5)
	family sizes: 20, 10, 10, 5, 5					family sizes: 20, 10, 10, 5, 5			
	alleles per locus					alleles per locus			
	4	8	12	16		4	8	12	16
loci	—	—	—	—	loci	—	—	—	—
4	22.9 (10,39)	10.8 (5,26)	8.10 (5,22)	6.85 (5,14)	4	25.1 (10,39)	16.7 (7,34)	10.1 (4,22)	9.30 (5,19)
8	8.77 (4,25)	5.43 (4,8)	5.15 (5,7)	5.16 (4,7)	8	11.0 (4,25)	6.71 (4,15)	5.38 (4,8)	5.45 (5,9)
12	5.77 (4,12)	5.04 (5,6)	5.00 (5,5)	5.00 (5,5)	12	6.59 (4,13)	5.29 (4,10)	5.02 (4,6)	5.05 (5,7)
16	5.23 (4,8)	5.01 (5,6)	5.00 (5,5)	5.00 (5,5)	16	5.45 (4,10)	5.04 (4,7)	5.00 (5,5)	5.00 (5,5)
	family sizes: 30, 5, 5, 5, 5					family sizes: 30, 5, 5, 5, 5			
	alleles per locus					alleles per locus			
	4	8	12	16		4	8	12	16
loci	—	—	—	—	loci	—	—	—	—
4	26.1 (10,41)	12.7 (5,30)	9.44 (4,27)	8.26 (4,22)	4	28.6 (10,44)	20.0 (5,41)	11.2 (4,29)	11.9 (4,28)
8	11.7 (4,28)	5.91 (3,12)	5.53 (5,8)	5.49 (5,7)	8	13.3 (2,36)	7.65 (3,23)	5.65 (4,8)	5.63 (4,9)
12	6.19 (3,16)	5.18 (4,7)	5.22 (4,6)	5.21 (5,6)	12	7.05 (3,16)	5.72 (3,10)	5.33 (4,7)	5.28 (3,7)
16	5.08 (3,8)	5.10 (4,6)	5.08 (4,6)	5.13 (5,7)	16	5.57 (3,14)	5.31 (4,7)	5.15 (5,6)	5.08 (5,6)
	family sizes: 40, 5, 2, 2, 1					family sizes: 40, 5, 2, 2, 1			

allowed for better estimation of the number of families per data set, with a smaller range of observed values. Lastly, accuracy was greatly improved for tests using the fourth family distribution. Figure 4 compares the algorithm's performance when using this distribution with 50 individuals, as in simulation set A, and when using 500 individuals, as in simulation set B. Accuracy is plotted for tests using eight loci with varying numbers of equifrequent alleles per locus. Performance was considerably better with the large data sets, which contained families of size 400, 50, 20, 20 and 10, with accuracy levels higher than 90% when using 12 or more alleles. Since the family distribution was equally skewed in the data sets from simulation set A, the performance difference is probably the result of the very small families, containing only one or two individuals, that were present in those data sets.

Simulation set C — effect of number of families per data set

The results of simulation set C, which tested the effect of the number of families per data set, are shown in Fig. 5. As in Fig. 4, accuracy is plotted for tests using eight loci with varying numbers of equifrequent alleles. The use of only four alleles, which was insufficient for most tests in simulation set A, produced poor results in this set of simulations as well. Figure 5 also shows a decrease in accuracy with data sets containing a single family. While accuracy in tests using eight or more alleles and a single family was between 73% and 88%, this was a considerable decrease from the perfect performance observed for most data sets containing more than one family. Finally, Fig. 5 illustrates that for data sets with 100 individuals, the number of families in the data set did not affect accuracy when using eight alleles and fewer than eight families or when using 12 or more alleles and fewer than nine families. Decreases in accuracy of up to 16% for tests using eight alleles and 3% for tests using 12 or more alleles were observed when the number of families per data set was increased beyond these limits, as would be expected when the same amount of information is used to differentiate among an increasing number of families.

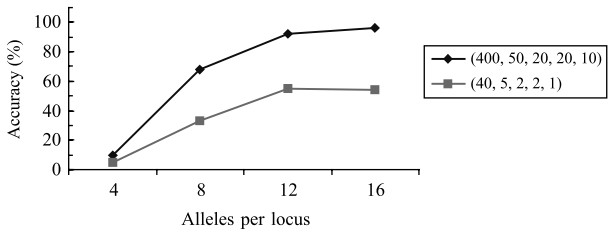

Fig. 4 Effect of very small families on accuracy. Results are shown for tests using eight loci, with four, eight, 12 and 16 equifrequent alleles per locus. The key gives the sizes of the families in the two distributions that were tested.

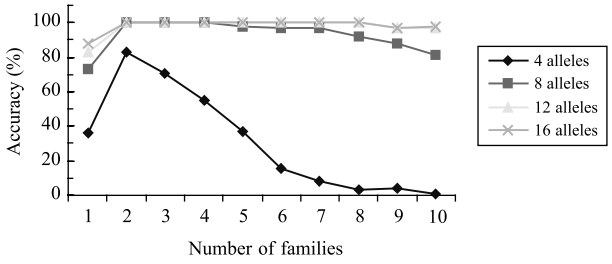

Fig. 5 Accuracy of the algorithm as a function of the number of families in a data set containing 100 individuals. Results are shown for simulations using eight loci, with four, eight, 12 and 16 equifrequent alleles per locus.

Discussion

We have presented an algorithm that partitions a single generation of individuals into full-sib families using pairwise likelihoods and existing graph algorithms. The graph-based model offers several advantages. First, it provides a simple means to summarize the information from pairwise analysis by representing all of the sibling relationships from a data set in a single structure. Families can then be inferred because of the transitive nature of the full-sibling relationship. Additionally, the minimum-cut algorithm can correct for pairs of individuals that have a high likelihood ratio as a result of chance. This correction is possible because an individual that does not belong to a given family is not likely to be well connected to the entire family. Therefore, the chance connections will probably be detected as linking edges and will be removed. Similarly, minor scoring errors can be tolerated and the algorithm need not force family genotypes to be genetically feasible. Finally, the model can be easily expanded to contain information about additional types of relationships within a data set and the vast collection of graph-based algorithms can be exploited to detect other levels of family structure.

Simulations show that the algorithm can produce accurate partitions using eight loci with 12 or more alleles per locus or 12 or more loci with eight or more alleles per locus. Additionally, when the focus of study is the number of families in a data set, the algorithm performs well despite the presence of a skewed distribution of family sizes. Accuracy decreases when alleles are not equally frequent, although this decrease is minimal when the data contain a relatively large number of loci and alleles. These results are comparable to results obtained by other recently proposed algorithms. Simulations performed by Thomas & Hill (2000) resulted in accurate partitions using 10 loci with five or more alleles per locus and simulations conducted by Smith *et al*. (2001) achieved accurate results with eight loci and eight or more alleles per locus.

While some previously proposed methods have required the choice of an appropriate prior distribution for the

population allele frequencies (Painter 1997) or family sizes (Thomas and Hill 2000), our algorithm does not require knowledge of these distributions. Thomas & Hill (2000) and Smith *et al*. (2001) also noted the tendency of large families to be split into smaller groups when the family size distribution is skewed. Both have developed allele frequency updating methods to correct partially for this problem when a skewed family size distribution is suspected. With the algorithm presented here, the performance decrease because of skewed distributions is negligible for data sets containing 12 or more loci, eight or more alleles, and families with more than four individuals.

The algorithm, however, has difficulty creating correct partitions when applied to data sets that contain very small families. This was particularly evident with data sets containing families with only one or two individuals, such as with the fourth family distribution. Accuracy levels of approximately 90% were observed, but required data sets with 16 loci and 12 or more alleles per locus. The second part of the algorithm, which refines the graph defined by the pairwise likelihoods, does not act on very small families as it is usually not favourable to split a family with so few individuals. The algorithm also produces less accurate partitions when applied to data sets containing a single family. With these data sets, the sample allele frequencies are equal to the allele frequencies within the family. Thus, they are not a good estimate for the allele frequencies in the population. This results in lower likelihood ratios, making it more probable that some individuals will be isolated from the rest of the family.

The approach taken here allows for many possible extensions to the algorithm. The sibling graphs could be modified to contain more information about the relationships among the individuals in the data set by using weighted graphs. Assigning edges with higher weights to pairs with higher likelihood ratios would allow for a more sophisticated scoring system that would probably produce better decisions as to when to split a graph into two families. Alternative scoring systems are also possible using the current graphs. For example, subgraph scores could be based on a likelihood value for the entire family group.

The algorithm could also be modified to detect half-sibs. Likelihood ratios could be calculated to identify pairs of individuals that are half-sibs, and edges representing these relationships could be added to the graph after the full-sib partition has been created. These edges would then link families that share a common parent, identifying half-sib families in the data set. Further enhancements may be possible for other relationships as well because of the availability of likelihood equations for many relationships (Goodnight & Queller 1999) and the wealth of existing graph algorithms that may have applications to the problem of pedigree reconstruction.

References

Almudevar A, Field C (1999) Estimation of single-generation sibling relationships based on DNA markers. *Journal of Agricultural, Biological, and Environmental Statistics*, **4**, 136–165.

Blouin MS, Parsons M, Lacaille V, Lotz S (1996) Use of microsatellite loci to classify individuals by relatedness. *Molecular Ecology*, **5**, 393–401.

Gomory RE, Hu TC (1961) Multi-terminal network flows. *SIAM Journal on Applied Mathematics*, **9**, 551–570.

Goodnight KF, Queller DC (1999) Computer software for performing likelihood tests of pedigree relationship using genetic markers. *Molecular Ecology*, **8**, 1231–1234.

Herbinger CM, Doyle RW, Taggert CT *et al*. (1997) Family relationships and effective population size in a natural cohort of Atlantic cod (*Gadus morhua*) larvae. *Canadian Journal of Fisheries and Aquatic Sciences*, **54** (Suppl. 1), 11–18.

Lynch M, Ritland K (1999) Estimation of pairwise relatedness with molecular markers. *Genetics*, **152**, 1753–1766.

Painter I (1997) Sibship reconstruction without parental information. *Journal of Agricultural, Biological, and Environmental Statistics*, **2**, 212–229.

Queller DC, Goodnight KF (1989) Estimating relatedness using genetic markers. *Evolution*, **43**, 258–275.

Ritland K (1996) Estimators for pairwise relatedness and individual inbreeding coefficients. *Genetical Research*, **67**, 175–185.

Smith BR, Herbinger CM, Merry HR (2001) Accurate partition of individuals into full-sib families from genetic data without parental information. *Genetics*, **158**, 1329–1338.

Thomas SC, Hill WG (2000) Estimating quantitative genetic parameters using sibships reconstructed from marker data. *Genetics*, **155**, 1961–1972.

Thomas SC, Hill WG (2002) Sibship reconstruction in hierarchical population structures using Markov chain Monte Carlo techniques. *Genetical Research*, **79**, 227–234.

This work is part of the MSc research of Jennifer Beyer at UC Davis. Bernie May is an Adjunct Professor of Animal Science and Director of the Genomic Variation Laboratory (GVL). Researchers at the GVL use molecular genetic techniques and analyses to address a diverse array of questions related to the conservation of threatened and endangered species and the genetic improvement of aquacultural taxa.
