PROGRAM NOTE

GERUD 2.0: a computer program for the reconstruction of parental genotypes from half-sib progeny arrays with known or unknown parents

ADAM G. JONES
Department of Biology, Texas A & M University, 3258 TAMU, College Station, TX 77843, USA

Abstract

Large groups of related progeny, as can be collected from discrete egg masses or fruits, present excellent opportunities for parentage analysis by allowing the reconstruction of parental genotypes. Current techniques of parental genotypic reconstruction require the knowledge of at least one parental genotype. Here, I present a new computer program that reconstructs parental genotypes when no parents are known, provided that the progeny array contains only full and half siblings. The prospects for successfully reconstructing parental genotypes for such progeny arrays with unknown parents are nearly as good as those for the case when one parent is known with certainty.

Keywords: maternity, mating systems, microsatellites, parentage, paternity

Received 28 October 2004; revision received 07 February 2005; accepted 01 April 2005

The molecular study of parentage provides fundamental data important to the investigation of numerous important biological subjects, such as sexual selection, behavioural ecology and conservation genetics (Avise et al. 2002). Many species of lower vertebrates, insects and plants produce large clutches or litters that lend themselves to numerous types of parentage analysis, one of which is the reconstruction of parental genotypes (Jones & Ardren 2003). The only existing computer program for reconstruction of parental genotypes from progeny arrays is GERUD1.0 (Jones 2001). One of the limitations of GERUD1.0, however, is that it requires that one of the parents of the progeny array be located, sampled and genotyped. Such a scenario may not be possible for field-collected groups of progeny. In this paper, I describe a major update of GERUD that (i) removes this constraint and (ii) improves the algorithm that GERUD uses to choose the best combination of parents.

The new version of GERUD, version 2.0, uses data from codominant genetic markers such as microsatellites or allozymes and is available for download at http://www.bio.tamu.edu/USERS/ajones/JonesLab.htm. Much like GERUD1.0, GERUD2.0 uses an exhaustive algorithm to reconstruct the minimum set of parents that can explain the progeny array, so it is useful both for determining rates of multiple mating and for deducing the genotypes of the parents of a progeny array (Jones et al. 2002). Because this program is designed to analyse progeny arrays containing only full and half siblings, all members of the array must have either the same mother or the same father. For the sake of clarity, throughout this paper, I assume a hypothetical progeny array with a single mother and multiple fathers. However, the program also can be used to analyse clutches with a single father and multiple mothers.

There are multiple techniques available to determine the number of sires contributing gametes to a progeny array with a single mother. The simplest method is the single-locus minimum method, in which the number of paternal alleles segregating at the locus with the largest number of alleles in the progeny array is divided by two, rounding up. Since, ignoring mutation, each potential parent can contribute at most two alleles, the actual number of parents can never be lower than this single-locus estimate. Another technique is the multilocus minimum method, in which the total number of multilocus paternal (or maternal) gametotypes (i.e. haplotypes across all loci) is divided by \(2^L\) and rounded up (Fiumera et al. 2001). Obviously, this multilocus minimum method potentially requires a much larger sample of progeny than the single-locus method, because even with only three loci, \(2^L = 8\). In contrast, the single-locus

Correspondence: A. G. Jones, Fax: (979) 845 2891; E-mail: agjones@tamu.edu

© 2005 Blackwell Publishing Ltd
method has the potential to detect multiple mating with a sample as small as three offspring if they all exhibit different paternal alleles. GERUD uses a distinct approach to determine the minimum number of parents for a progeny array that makes full use of the multilocus data.

The GERUD2.0 algorithm starts by determining all maternal genotypes consistent with the progeny array. For the first locus, all alleles present in the progeny array are combined in all pairwise arrangements (including each allele with itself). Each pairwise combination of alleles is tested against the progeny array. If each individual from the progeny array possesses one or the other of the alleles in a given pair, then the genotype is retained as a potential maternal genotype. This procedure is repeated for all loci, yielding a set of potential maternal genotypes for each locus. These single-locus maternal genotypes are then combined across loci in every possible arrangement to produce a set of potential multilocus maternal genotypes.

For each potential maternal genotype, GERUD2.0 uses a stepwise procedure to determine the minimum number of sires necessary to explain the progeny array. First, all paternal alleles are determined by subtraction of the known maternal allele. For offspring with the same genotype as the mother, both alleles are retained as potential paternal alleles. Next, these paternal alleles are combined to produce all possible multilocus genotypes. These potential paternal genotypes are tested in combination to determine which set of potential paternal genotypes could have produced the entire progeny array. If one paternal genotype can explain the array, then the minimum number of sires is one and the program retains all single-sire solutions. If one sire cannot explain all of the progeny genotypes in the array, then all pairwise combinations of potential paternal genotypes are tested. If two sires cannot explain the array, then all combinations of three paternal genotypes are tested, and so forth. Thus, for each potential maternal genotype, GERUD2.0 determines the minimum number of paternal genotypes that are necessary to produce the genotypes of the progeny in the array. Only the solutions with the smallest number of parents are retained. Some progeny arrays have unique solutions, which are usually correct (Jones 2001), but others may have multiple solutions, and some of these solutions may have different maternal genotypes.

For those progeny arrays with multiple minimum-parent solutions, GERUD2.0 can choose the most likely solution (each of which consists of a single maternal genotype and one or more paternal genotypes), based on patterns of Mendelian segregation and expected genotypic frequencies in the population. The genotypic frequency probability for a given solution is simply the product of expected genotypic frequencies across all loci and parents in the solution. The Mendelian segregation probability is the binomial probability that a given genotype would produce the observed number of copies of each of its alleles in the progeny array, assuming random segregation. These Mendelian probabilities are then multiplied across all parents and loci to produce the Mendelian probability for the entire solution. The total relative probability for a solution is the product of the genotypic frequency probability and the Mendelian frequency probability. The program ranks the solutions by their relative probabilities and returns them to the user.

To facilitate assessment of the power of this algorithm to reconstruct parental genotypes, the GERUD2.0 package also includes GERUDSIM2.0, a program that simulates progeny arrays based on parameters supplied by the user and tests how often the GERUD2.0 algorithm recovers the correct number and genotypes of the parents. This program requires the population allele frequencies for each locus under consideration. It then randomly generates progeny arrays and analyses them using the GERUD2.0 algorithm. GERUDSIM2.0 reports how frequently it recovers the correct number of parents as well as the correct paternal and maternal genotypes under the hypothesized pattern of parentage.

The GERUD2.0 algorithm recovers the correct maternal genotype in 95% or more of simulated progeny arrays, except when markers have low levels of polymorphism or the progeny array consists of full sibs only. The reason for the latter result is that for progeny arrays with a single mother and father, it is impossible to reconstruct multilocus genotypes, because it is not possible to determine which genotype is maternal vs. paternal at each locus. When there are multiple sires, however, the maternal genotype is clearly the one that is consistent with all of the progeny in the array, whereas the other genotypes are paternal in origin. For a more detailed description of the logic underlying the deduction of multilocus genotypes, see Jones & Avise (1997).

Interestingly, the lack of a known maternal genotype has almost no effect on the probability of correct reconstruction of paternal genotypes from progeny arrays (Fig. 1). The results when the maternal genotype is known (Fig. 1a, c) compared to those when the mother is unknown (Fig. 1b, d) show only a slight decrease in success rates. Figure 1 also shows that for reasonably polymorphic loci, with an average of eight or more alleles per locus, the GERUD2.0 algorithm does an excellent job of reconstructing both the number of parents and their genotypes. GERUD2.0 also outperforms the single-locus minimum method for inferring multiple paternity under most circumstances.

Figure 2 illustrates several important aspects of GERUD2.0. Figure 2 (a, b) shows how parental reconstruction and determination of the number of sires depend on the sample size, given allele frequencies observed in actual populations of organisms. Low levels of polymorphism at the loci under consideration in two of the species limit the ability...
Fig. 1 Probabilities of success when **gerud**2.0 is used to reconstruct the number of sires (top panels) or the genotypes of the sires (bottom panels) from a progeny array with one mother whose genotype is either known (left panels) or unknown (right panels). These results are based on 1000 simulation runs of **gerud**sim2.0, assuming two loci with equal levels of polymorphism. Each locus is assumed to have 2, 4, 8, 16 or 32 equally frequent alleles, and each male is assumed to sire 900 offspring in a progeny array from which 60 offspring are sampled for analysis. The single-locus technique in the top panels (solid lines) represents the single-locus minimum technique in which the number of paternal alleles at the locus with the largest number of paternal alleles segregating in the progeny array is divided by two and rounded up to produce a measure of the minimum number of sires. The multilocus method (dashed lines in top panels) is based on the exhaustive reconstruction method used by **gerud**2.0. Note that the results when the mother’s genotype is known (left panels) do not differ substantially from the results when the maternal genotype is unknown (right panels).

Fig. 2 The performance of **gerud**2.0, given the allele frequency distributions observed in actual natural populations. All four panels use data from the Banggai cardinalfish (Hoffman et al. 2004), rough-skinned newts (Jones et al. 2001) and broad-nosed pipefish (Jones et al. 1999). The top panels show the effect of sample size on the ability of **gerud**2.0 to reconstruct parental genotypes (a) or to infer the correct number of sires (b) when the mother is unknown. These simulations assume a progeny array with 900 offspring from each of three sires, from which, a sample of two to 100 progeny is drawn at random to be analysed. Each point is based on 1000 simulation runs. The lower left panel (c) compares results from **gerud**1.0 to those from **gerud**2.0 when the mother’s genotype is known. These first three panels (a–c) use loci Pka21 (n = 6 alleles, $H_E = 0.74$) and Pka24 (n = 6, $H_E = 0.78$) from Banggai cardinalfish (population O, Hoffman et al. 2005), Tgr01 (n = 12, $H_E = 0.88$) and Tgr06 (n = 9, $H_E = 0.80$) from rough-skinned newts and typh04 (n = 39, $H_E = 0.95$) and typh16 (n = 20, $H_E = 0.92$) from broad-nosed pipefish. Note that these loci were chosen for the purposes of illustration, so they are not necessarily the best loci to use for actual parentage analysis in these species. The final panel (d) shows the effects of adding loci on the performance of **gerud**2.0. Additional loci are added by starting either with the two most polymorphic loci followed by the next polymorphic locus and so on (high to low, solid lines) or by starting with the least polymorphic locus followed by progressively more polymorphic ones (low to high, broken lines). In addition to the loci mentioned previously, this panel (d) used loci Pka09 (n = 8, $H_E = 0.74$) and Pka13 (n = 13, $H_E = 0.82$) from cardinalfish, Tgr02 (n = 18, $H_E = 0.91$) and Tgr10 (n = 15, $H_E = 0.92$) from newts and typh12 (n = 15, $H_E = 0.60$) and typh18 (n = 38, $H_E = 0.97$) from pipefish.
of gerud2.0 to correctly infer multiple paternity and to reconstruct parental genotypes (Fig. 2a, b). Figure 2(c) shows that the new algorithm implemented in gerud2.0 outperforms the original algorithm for ranking solutions used by gerud1.0 across a wide range of conditions. Finally, Fig. 2(d) indicates that additional loci can produce improved results. Interestingly, using additional loci with higher levels of polymorphism generally improves results (low to high, dashed lines in Fig. 2d). However, adding additional loci with low levels of polymorphism can actually decrease the efficacy of genotypic reconstruction. The best strategy will typically be to use the three loci with the highest levels of polymorphism in the initial parentage analysis, and then to verify that genetic data from other loci are consistent with the initial results.

In summary, gerud2.0 and gerudsim2.0 provide a new tool for the reconstruction of parental genotypes from progeny arrays. These programs improve upon the existing techniques for the reconstruction of parental genotypes from a progeny array with a known parent by improving the way the most likely solution is chosen. They also provide a new method for reconstructing parental genotypes from progeny arrays with unknown parents, provided that the progeny array is composed of full and half siblings.

Acknowledgements

I am grateful to Michael Goodisman for helpful discussions about these techniques. This work was supported by funds from Texas A & M University and a grant from the National Science Foundation.

References


