ABSTRACT

Summary: In family-based genetic studies, it is often useful to identify a subset of unrelated individuals. When such studies are conducted in population isolates, however, most if not all individuals are often detectably related to each other. To identify a set of maximally unrelated (or equivalently, minimally related) individuals, we have implemented simulated annealing, a general-purpose algorithm for solving difficult combinatorial optimization problems. We illustrate our method on data from a genetic study in the Old Order Amish of Lancaster County, Pennsylvania, a population isolate derived from a modest number of founders. Given one or more pedigrees, our program automatically and rapidly extracts a fixed number of maximally unrelated individuals.

Availability: http://www.hg.med.umich.edu/labs/douglaslab/software.html (version 1.0.0)

Contact: jddoug@umich.edu

1 INTRODUCTION

Genetic studies frequently involve the sampling of families. However, for some practical applications, e.g. polymorphism discovery, it is often valuable to identify a maximal subset of available unrelated (MSAU) individuals from such samples. This is a combinatorial optimization problem that belongs to the class of NP-complete (non-deterministic polynomial time complete) problems (Martin et al., 2003). In fact, Martin et al. showed that any instance of the maximal-clique problem from graph theory, which is known to be NP-complete, can be converted to the MSAU problem by an efficient algorithm, i.e. with polynomial time complexity. Algorithms for finding deterministic solutions to NP-complete problems are conjectured to increase in complexity at an exponential rate in the problem size, \( n \) (Brookshead, 1989). In this case, identifying an optimal set of unrelated individuals requires examining \( 2^n \) subsets (where \( n \) is the number of individuals), which is impractical for even modest \( n \). To address this specific problem, Martin et al. (2003) proposed and implemented some reduction techniques (in a program named Pedstrip). Because the Pedstrip algorithm implicitly assumes the existence of many unrelated, available individuals, it is not applicable (or useful) when none (or only a small number) of the available individuals are unrelated, as is typical of samples from population isolates.

Owing to finite population size, most (if not all) individuals from population isolates are often detectably related to one another, i.e. any two individuals share one or more recent, common ancestors. In fact, when genetic studies are conducted in population isolates, smaller family units can typically be connected into a single, large extended family via genealogical records (e.g. see Fig. 1). Moreover, because the founders are usually dead and immigration rates are customarily low, most of the individuals available for study are non-founders. Thus, when samples from family-based genetic studies are derived from population isolates, the problem outlined and addressed by Martin et al., namely, identifying a maximal subset of unrelated individuals, is transformed to the problem of identifying a set of maximally unrelated (or equivalently, minimally related) individuals. To tackle this associated problem, we implemented simulated annealing (Cerny, 1985; Kirkpatrick et al., 1983), a general-purpose algorithm for solving difficult combinatorial optimization problems.

2 ALGORITHM AND IMPLEMENTATION

Simulated annealing (SA) is a probabilistic algorithm for finding the global minimum of a cost function that may possess several local minima. It works by emulating the physical process whereby a solid is slowly cooled so that its structure is eventually ‘frozen’ at a minimum energy configuration. Energy \( (E) \) increases can occur during the cooling process, but their frequency is governed by the laws of thermodynamics, which state that at temperature \( T \), the probability of an increase in energy of magnitude \( \Delta E \) is given by \( \exp(-\Delta E/T) \). SA can be applied to our problem (and optimization problems in general) by mapping the basic elements of this physical cooling process onto the elements of a combinatorial optimization problem.

In our implementation, we select a subset of individuals (system states or feasible solutions) from the sample such that the maximum or mean pair-wise kinship coefficient (energy or cost) in the subset is minimized. Conditional on the pedigree, the kinship coefficient between two individuals is the probability that a randomly chosen allele from one individual and a randomly chosen allele from the other individual at an autosomal locus are inherited identical by descent from a recent, common ancestor. Given one or more pedigrees in the sample,

*To whom correspondence should be addressed.
temperature, SA with the maximum kinship coefficient as the cost function resulted in a subset of more distantly related individuals relative to a random subset. For example, based on 10,000 replicate runs of SA and random sampling of 20 individuals, the median [range] of the maximum kinship coefficient was 0.0386 [0.0376–0.0392] under SA versus 0.2767 [0.0641–0.2905] with random sampling. In addition (and as expected), the maximum cost function led to a larger decrease in the maximum kinship coefficient relative to the mean cost function, e.g. median of 0.0386 versus 0.0546. Conversely, the mean cost function led to a larger (but diminutive) decrease in the mean kinship coefficient relative to the maximum cost function, e.g. median of 0.0250 versus 0.0272.

To reduce the number of feasible solutions, we tested SA on a trimmed pedigree by removing descendants and siblings of sampled individuals. For the current application, trimming reduced the number of sampled individuals to 110 and had virtually no impact on the final result and total run time. For example, based on the same parameters as described above and 10,000 replicate runs of SA, the median [range] of the maximum kinship coefficient was 0.0410 [0.0410–0.0410] under SA with the maximum kinship coefficient as the cost function (compared to 0.115 [0.1018–0.1491] with random sampling). Note that for some sampling configurations, including the current one, running the algorithm on a trimmed (rather than an untrimmed) pedigree may result in a final subset of individuals whose pair-wise kinship coefficients are somewhat higher. This is not due to the SA algorithm but is an artifact of the trimming procedure, which removes sampled descendants (rather than sampled ancestors) of sampled individuals.

We also considered several different neighborhood sizes, including random replacement of 1, 2 and 5 individuals. Using neighborhood sizes greater than 1 substantially increased the total run time with no improvement in the final result. For example, to achieve results with a neighborhood size of 5 that were even comparable to those with a neighborhood size of 1 (as described above) required 250,000 iterations (rather than 1000) at each temperature. As a result, the total run time increased to ~42 min relative to ~6 s (based on a server running RedHat Enterprise Linux with a 3 GHz Intel Duo Core Xenon processor and 16 GB of RAM).

3 APPLICATION

We applied our SA approach to a sample of 343 related women from extended Amish families in a genetic study of breast density, with the goal of identifying a subset of 20 maximally unrelated women for subsequent sequencing. All 343 women were connected into a 12-generation 1711-member pedigree using AGDB 4.0 (Agarwala et al., 2001) and PedHunter (Agarwala et al., 1998). In total, there were 425 sibling pairs (from 115 different nuclear families), 15 parent–offspring pairs, 78 aunt–niece pairs, 14 double first cousin pairs, 818 first cousin pairs and 7575 more distantly related pairs after merging in three generations from AGDB. Based on the 12-generation pedigree, which connects individuals with all known recent, common ancestors, the median [range] pair-wise kinship coefficient among the 343 sampled individuals was 0.04 [0.01, 0.29].

Given a neighborhood size of 1, an initial temperature of 1, a temperature reduction rate of 0.9, and 1000 iterations at each temperature, SA produced good solutions for our application (see below).

To start the algorithm, we randomly sample a preset number of individuals and set an initial temperature $T$. Then we randomly replace an individual in the initial subset with an individual outside the subset (neighboring solution). If this new subset results in a lower or higher energy, it is accepted (change of state) with probability 1 or $\exp(-\Delta E/T)$, respectively. This process is iteratively repeated according to a specified cooling schedule, including number of iterations at each temperature and a geometric temperature reduction function (control parameters). The algorithm stops when the temperature reaches zero (frozen state or heuristic solution). Although it is not possible to prove that SA produces the optimal result (given the impracticality of an exhaustive search), empirical work suggests that it produces good solutions for our application (see below).

We also implemented our SA algorithm in a freely available Java program named PedMine (Pedigree Mining). Our program allows the user to select a fixed number of maximally unrelated individuals (without accounting for the sex of each individual) or fixed numbers of maximally unrelated male and female individuals. In addition to being platform independent, PedMine is fast, easy to use and does not depend on the use of third-party programs. For example, using a standard pedigree file as input and a laptop computer running Windows XP (with a 1.83 GHz Intel Duo Core Centrino processor and 512 MB of RAM), one SA run (including kinship coefficient calculations) was completed in approximately 14 s for a 1711-member pedigree with 343 sampled individuals. We have also successfully applied PedMine to a larger pedigree (specifically, a 3797-member pedigree with 861 sampled individuals), and
although the same parameters and decisions as illustrated above worked well for this pedigree (data not shown), they should be carefully tuned for each application to ensure convergence.

ACKNOWLEDGEMENTS
We thank Kenneth Lange for his helpful suggestions on the material presented here. We also thank Alejandro Schaffer and Richa Agarwala for making the pedigree information available to us. This research was supported by NIH grants R01 CA122844 (J.A.D.) and T32 HG00040 (C.I.S.) and by the Program in Biomedical Sciences at the University of Michigan.

Conflict of Interest: none declared.

REFERENCES
Agarwala, R. et al. (1998) Software for constructing and verifying pedigrees within large genealogies and an application to the Old Order Amish of Lancaster County, Genome Res., 8, 211–221.