Pedstrip: extracting a maximal subset of available, unrelated individuals from a pedigree

Jeremy M. R. Martin*, Ian J. Mackay and Christopher J. Rawlings

Oxagen Ltd, 91 Milton Park, Abingdon, Oxon OX14 4RY, UK

Received on April 29, 2002; revised on July 15, 2002; accepted on July 20, 2002

ABSTRACT

Summary: Certain types of genetic analysis are simplified by assembling a collection of unrelated individuals, e.g. case-control experiments. If a family study is being performed then it will be necessary to extract subsets of unrelated, available individuals from pedigrees. Our program provides an optimal method for performing this task.

Availability: The software is available, free of charge, on request from Oxagen Ltd.

Contact: SD@oxagen.co.uk

Supplementary information: http://www.oxagen.co.uk

Family studies are designed to detect and estimate genetic effects by exploiting relationships between individuals. However, data and blood samples can sometimes have supplementary uses for which it is desirable that the largest collection of unrelated individuals be selected. Below we give two examples of such use.

(1) Samples of DNA may be required to search for polymorphisms within candidate genes. This search will be most efficient if the selected individuals are unrelated. Other constraints, on the quantity of DNA available or the affected status of an individual, can still be included in the selection criteria.

(2) Unrelated affected and unaffected individuals from one study may be required to generate all or part of a case control collection. In this case, although one could argue that a better experiment would use the entire family collection, this might prove too expensive. Moreover, selection of unrelated individuals fits better with DNA pooling strategies, which are being increasingly employed.

The problem that we wish to solve, therefore, is as follows. Given a pedigree, where some individuals, including founders, may be excluded from the study, construct the subset of the largest possible size, within which no two members are blood-related. Henceforth we shall refer to this problem as MSAU (Maximal Subset of Available Unrelateds).

A brute-force approach to solving the problem would be to examine all subsets of available individuals of the pedigree. However to solve a pedigree containing n available individuals would require the inspection of 2^n subsets.

A refinement to the brute-force method is to examine subsets in order of increasing size. We start by considering all pairs of available individuals. If an unrelated pair is found we store that as the current ‘winner’ and we move on to triples, then quadruples, quintuples etc. As soon as a size k is found for which no subset of available unrelated individuals exists, there is no need to search any further. Unfortunately this approach is still infeasible for large pedigrees.

In fact MSAU turns out to be NP-complete (Brookshear, 1989) so there would appear to be little hope of finding an efficient algorithm to solve the general case. Nonetheless we have discovered some reduction techniques, which we believe will make it possible to solve the vast majority of cases found in practice efficiently.

FORMAL STATEMENT OF THE MSAU PROBLEM

Consider a pedigree P which is a directed, acyclic graph (V(P), E(P)). The vertices V(P) represent individuals and the edges E(P) represent parent–child relationships. Because nobody has more than two parents the maximum in-degree of any vertex is 2.

The descendents of an individual i may be found by constructing the set of all individuals reachable from vertex i using a walk of non-zero length. The ancestors of i are those individuals for which i is a descendant. Individuals with no ancestors are called founders.

There is a Boolean function, available, which is defined on the individuals V(P) to define whether they are available or unavailable.

DEFINITION. two individuals, i and j, are unrelated if ancestors(i) and ancestors(j) are disjoint sets.

*To whom correspondence should be addressed.
**THE PROBLEM**

find a maximal subset of unrelated, available individuals in \( P \).

**THE PEDSTRIP ALGORITHM**

We have discovered four important properties, which enable us to simplify solving MSAU for large pedigrees.

**RULE 1.** All descendents of any available individual can be deleted from \( P \) without changing the size of the maximal set of unrelated, available individuals.

**RULE 2.** Any unavailable individual, with no offspring, can be deleted from \( P \) without changing the size of the maximal set of unrelated, available individuals.

**RULE 3.** Consider a set of available siblings \( \{s_1, s_2, \ldots, s_n\} \) within \( P \), each with no offspring. All but one of any such set may be deleted without changing the size of the maximal set of unrelated, available individuals.

**RULE 4.** Suppose that \( P \) is disconnected and decomposes into connected components \( \{C_1, C_2, \ldots, C_m\} \). Then solving MSAU for \( P \) reduces to solving MSAU for each component \( C_i \) and then joining together the resultant subsets.

It is important to note that none of the pruning operations described in rules 1–3 can alter a relationship between two individuals that remain in the pedigree afterwards.

Based on these rules, the Pedstrip algorithm proceeds as follows:

**STEP 1.** delete any descendent of an available individual;

**STEP 2.** delete any unavailable individual with no remaining offspring (this step should be repeated until no such individual remains);

**STEP 3.** prune each remaining set of available sibling down to one individual;

**STEP 4.** calculate the set of connected components of the remainder of the pedigree;

**STEP 5.** for each connected component, solve MSAU by an exhaustive search of subsets, ordered by size.

For illustration, here is the output from Pedstrip when it is run on the pedigree shown in Figure 1 (in which unavailable individuals are indicated using a diagonal line):

- deleting offspring of an available parent: \((12, 20, 21, 13, 22, 15, 16, 17, 18, 19)\)
- deleting unavailable individuals with no remaining offspring: \((10, 5)\)
- pruning available siblings: \((8)\)
- partitioning available remainder: \((11) (6) (7) (14) (23) (9)\)
- maximal set of unrelated individuals: \((6, 7, 9, 11, 14, 23)\)

**NP-COMPLETENESS**

We mentioned above that the MSAU problem falls into the classification of NP-complete, as do many interesting problems in Bioinformatics and Computer Science (Brookshear, 1989) A proof of this fact is given in the on-line supplement to this paper, together with a formal proof of correctness of the four rules that underpin the Pedstrip algorithm. Essentially this involves showing that any instance of the maximal-clique problem from graph theory, which is known to be NP-complete, can be converted to MSAU by an efficient algorithm (i.e. with polynomial complexity).

No efficient solution has ever been found to solve any NP-complete problem. However it does seem that MSAU is one of the fortunate problems for which the pathological cases are extremely rare in practice. We are yet to find a real example that Pedstrip is unable to solve very rapidly.

We have applied it successfully to several pedigrees of size 1000 or more, which would have been totally infeasible to solve using brute force alone.

The Pedstrip program was developed entirely using Perl and has a graphical interface which uses the Perl/Tk module. It will run with any operating system that supports Perl and Tk, including Windows and all varieties of Unix. It uses a standard linkage file as input.

**REFERENCES**