Supplementary Material

Ancestral Haplotype Reconstruction in Endogamous Populations using Identity-By-Descent

Kelly Finke¹,², Michael Kourakos¹, Gabriela Brown¹, Huyen Trang Dang³, Shi Jie Samuel Tan⁴, Yuval B. Simons⁵, Shweta Ramdas⁶, Alejandro A. Schäffer⁷, Rachel L. Kember⁸, Maja Bućan⁶, Sara Mathieson⁴,

¹ Department of Computer Science, Swarthmore College, Swarthmore, Pennsylvania, USA
² Department of Biology, Swarthmore College, Swarthmore, Pennsylvania, USA
³ Department of Computer Science, Bryn Mawr College, Bryn Mawr, Pennsylvania, USA
⁴ Department of Computer Science, Haverford College, Haverford, Pennsylvania, USA
⁵ Department of Genetics, Stanford University, Stanford, California, USA
⁶ Department of Genetics, University of Pennsylvania, Philadelphia, Pennsylvania, USA
⁷ Cancer Data Science Laboratory, National Cancer Institute, National Institutes of Health, Bethesda, Maryland, USA
⁸ Department of Psychiatry, University of Pennsylvania, Philadelphia, Pennsylvania, USA
† Corresponding author: Sara Mathieson, smathieson@haverford.edu
Algorithm 1: Overview

**Input:** \( G \) = genotyped individuals, \( NG \) = non-genotyped individuals, \( P \) = pedigree tree relating all individuals in \( G \) and \( NG \)

**Output:** \( R \) = reconstructed individuals, \( G_p \) = groups for each individual \( p \in R \)

find IBDs shared between \( G \) using GERMLINE

for \( I_k \in \text{IBDs} \) do
    \( C_k \) = cohort of individuals from \( G \) sharing \( I_k \)
    \( S_k \) = sources of \( C_k \) (Algorithm 2)
    \( d_k(s) \) = number of descendence paths for each \( s \in S_k \) (Algorithm 2)
end

\( R = G \)

\( IS \) = list of IBDs to source

while \( R \) not changing and \( IS \) not empty // this loop tracks iterations do
    for \( I_k \in IS \) do
        while assignment unsuccessful and \( S_k \) is not empty do
            selected source \( s^* = \arg \min_s d_k(s) \)
            if \( d_k(s^*) \) > path threshold then
                ignore \( I_k \)
            end
            else
                individuals \( D_k(s^*) \) = all individuals lying on each path from \( s^* \) to \( C_k \)
                assign \( I_k \) to all individuals in \( D_k(s^*) \)
                if \( I_k \) conflicts with reconstructed individual in \( D_k(s^*) \) then
                    remove \( I_k \) from all \( D_k(s^*) \)
                    remove \( s^* \) from \( S_k \)
                    assignment is unsuccessful
                end
            end
        end
        reset \( IS \) to empty list
    end

for individual \( p \in NG \) do
    \( G_p \) = reconstructed haplotype groups (Algorithm 3)
    if exactly 2 strong groups in \( G_p \) then
        add \( p \) to \( R \)
    end
    if 2 strong groups and one or more weak groups in \( G_p \) then
        remove weak groups from \( G_p \)
        add all IBDs from weak groups to \( IS \)
        add \( p \) to \( R \)
    end
end

return \( R, G_p \) for each \( p \in R \)
Algorithm 2: Source and Descendance Path Finding

**Input:** $C$ = a cohort of individuals sharing a single IBD, $\mathcal{P}$ = pedigree tree containing relationships between individuals

**Output:** $S$ = a list of possible non-redundant sources for cohort $C$

```plaintext
queue $Q = \text{list}(C)$

for cohort member $p \in C$ do
    multiset $M_p = \{p\}$
end

while $Q$ is not empty do
    individual $p = Q.pop$
    if $p$ is married-in then
        skip the following (married-in have no known ancestors)
    end
    if $p^{(f)}$ has not been processed then
        father’s multiset $M_f = M_p$
        father’s children set $CH_f = p$
        add father to $Q$
    end
    else
        extend father’s multiset $M_f$ by $M_p$
        add $p$ to father’s children set $CH_f$
        add $M_p$ and $p$ to $M$ and $CH$ of any processed ancestors of father
    end
    repeat process for $p^{(m)}$
end

sources $S = \text{all individuals } p \text{ s.t. } M_p \text{ contains all } c \in C$

for source $s \in S$ do
    $M_{ch_{\text{max}}} = \text{largest } M_{ch} \text{ for } ch \in CH_s$
    if length of $M_s = M_{ch_{\text{max}}}$ then
        remove redundant source $s$ from $S$
    end
end

for source $s \in S$ do
    if $s.spouse$ in $S$ and $M_s = M_s.spouse$ then
        remove $s$ and $s.spouse$ from $S$
        add couple $s$&$s.spouse$ to $S$, s.t. $M = M_s$ and $CH = CH_s$
    end
end

for source $s \in S$ do
    number of descendance paths $d(s) = \prod_{c \in C} m_s(c)$, where $m_s(c)$ = multiplicity of $c$ in $M_s$
end

return $S$ and $d(s)$ for all $s \in S$
```
Algorithm 3: Grouping

Input: \( R = \) genotyped or reconstructed individuals, \( A = \) non-reconstructed individuals, ungrouped IBDs \( I_p \) have been placed in each individual \( p \)

Output: \( G_p = \) groups for each individual \( p \)

for individual \( p \in R \) do
    for IBD \( I \in I_p \) do
        add \( I \) to one or both groups in \( G_p \) depending on zygosity
    end
end

for individual \( p \in A \) do
    find any homozygous groups \( G_p^{(o)} \)
    use overlapping IBDs in \( I_p \) to build heterozygous groups \( G_p^{(e)} \)
    duplicate groups in \( G_p^{(o)} \) and create \( G_p = G_p^{(o)} \cup G_p^{(e)} \)
    remove all IBDs from \( S_p \) that were used to build groups in \( G_p \)
    for pairs of groups \( G_i, G_j \in G_p \) and remaining IBD \( I \in I_p \) do
        if \( I \) overlaps \( G_i \) and \( G_j \) sufficiently then
            merge \( G_j \) into \( G_i \) and delete \( G_j \)
        end
    end
    for pairs of groups \( G_i, G_j \in G_p \) do
        if \( G_i \) and \( G_j \) overlap or “line up” then
            merge \( G_j \) into \( G_i \) and delete \( G_j \)
        end
    end
end
Complexity Analysis

The worst-case time complexity analysis involves several incomparable quantities. We use:

\[ |G| = \text{number of genotyped individuals} \]
\[ |NG| = \text{number of ungenotyped individuals with a genotyped descendant} \]
\[ |R| = \text{number of reconstructed individuals} \]
\[ |S| = \text{number of sources} \]
\[ |P| = \text{number of individuals in the pedigree} \]
\[ |A| = \text{number of generations} \]

and note that \( |G| \leq |P| , \quad |NG| \leq |P| , \quad |R| \leq |P| , \quad \) and \( |S| \leq |P| \). Additionally, we use:

\[ |I| = \text{number of unique IBD segments} \]
\[ I_{\text{max}} = \text{maximum number of SNPs within an IBD segment} \]
\[ |H| = \text{total number of SNPs on the chromosome} \]
\[ |J| = \text{sum of number of SNPs in all IBDs, which is bounded by} I_{\text{max}} \cdot |I| \]
\[ |C| = \text{sum of all cohort sizes} \]
\[ r = \text{number of iterations of the algorithm} \]

We analyze four parts:

1. processing of \texttt{GERMLINE} output as input to thread
2. Algorithm 3
3. Algorithm 2
4. Algorithm 1

Each row of \texttt{GERMLINE} output reports two individuals sharing one IBD. Therefore the size of the \texttt{GERMLINE} output file is \( O(|I||G|^2) \), which we simplify to \( O(|I||P|^2) \). These rows have to be processed to assign to each individual their corresponding IBDs and compile lists of individuals sharing each IBD. The two gathering tasks can be implemented to take time \( O(|I||P|^2) \) and in a memory efficient manner if we assume that the \texttt{GERMLINE} output is sorted so that all rows for the same IBD are consecutive. However, the current implementation takes time \( O(|I||P|^3) \) because it uses a general and inefficient implementation of union-find to determine the set of individuals carrying each IBD. In this context, the sets are finite and we know that the final result will be a single set for each IBD, so union-find can be implemented in linear \( O(|P|) \) time using bit arrays.

The time-consuming steps in Algorithm 3 compare IBDs for overlap and conflicts. In the worst case one might have to compare every SNP in one IBD to every SNP in another IBD. Algorithm 3 is run for every ungenotyped individual, so the worst case running time for all uses of Algorithm 3 in one iteration is \( O(|NG||J|^2) \), which we simplify to \( O(|J|^2|P|) \).

In Algorithm 2, building the multisets takes \( O(|P|^2) \) time. Testing for sources takes \( O(|P||C|) \) time. Removing redundant sources takes \( O(|P|) \) time. Replacing single sources by couples takes \( O(|P|) \) time. Computing products of the number of paths takes \( O(|S||C|) \) time, which is \( O(|P||C|) \). The dominant time is to build all the paths from one source to one target, which \( O(|P|^2 \cdot 2^{|A|}) \), where \( A \) is the number of generations. Overall, this is
There are three time-consuming parts of Algorithm 1 with different complexities. The assignment steps require \(O(I_{\text{max}}|I||P|)\) time. The costs of identifying conflicts and overlaps to make the groups for each individual is \(O(|H||P|)\) with a careful implementation of these two tests. As indicated above the cost of all the calls to Algorithm 3 is \(O(|J|^2|P|)\). Putting the three terms together, we get \(O(I_{\text{max}}|I| + |H| + |J|^2|P|)\). For this data set and any interesting endogamous data set, we would expect the \(O(|J|^2|P|)\) term to dominate.

Since the cost of Algorithm 3 is subsumed in Algorithm 1, we do not include it in the final tally. Algorithms 1 and 2 are used for \(r\) iterations, while the processing of GERMLINE output is done only one time. Putting together the costs of preprocessing and Algorithms 1 and 2, we arrive at a complexity of

\[
O \left( |I||P|^3 + r(P^2 \cdot 2^{|A|} + |C||P| + I_{\text{max}}|I| + |H| + |J|^2|P|) \right)
\]

In a typical dataset, we would expect \(|P|\) to be far smaller than \(|H|\), \(|I|\), or \(|J|\) and that is true for this dataset. Thus, we expect that the dominant would be \(O(r|J|^2|P|)\) or the incomparable exponential term \(O(rP^2 \cdot 2^{|A|})\). Since these two terms are incomparable, we performed runtime profiling with the python module cProfile. These experiments validated that the procedures taking \(O(r|J|^2|P|)\) and \(O(r(P^2 \cdot (2^{|A|}))\) dominate. For the pedigree structures tested, on the shortest chromosomes (21, 22), these two parts of thread take similar amounts of time, within a multiplicative factor of 2. On most of the (longer) chromosomes the \(O(r|J|^2|P|)\) term dominates because \(|J|\) increases with the length of the chromosome, but the costs of finding paths from sources \(O(rP^2 \cdot 2^{|A|})\) does not depend substantially on the chromosome length; on the longer chromosomes, one might need to consider more potential sources, but this consideration does not affect the asymptotic upper bound on running time.

### Probabilistic source identification

Given an IBD segment \(I\) and associated cohort \(C\) (genotyped individuals that carry \(I\)), we wish to approximate the probability that each potential source \((s_1, s_2, \ldots, s_k)\) is the true origin of \(I\). At a high level, we compute this by compiling the probabilities of transmitting the IBD from the source to each member of the cohort.

Let the genetic distance of \(I\) be \(d\) cM. Then the probability of a recombination event within the IBD segment during one meiosis is:

\[
r = \frac{1 - e^{-2d/100}}{2}
\]

First we will compute the probability that a child receives 0, 1, or 2 copies of the IBD from its parents. Let \([f_0, f_1, f_2]\) be the probabilities that the first parent has 0, 1, or 2 copies of the IBD, and similarly for the second parent \([m_0, m_1, m_2]\). Then we can compute the child probabilities as
follows:

\[
c_0 = \left( f_0 + \frac{1}{2} f_1 (1 + r) \right) \left( m_0 + \frac{1}{2} m_1 (1 + r) \right)
\]

\[
c_1 = \left( f_0 + \frac{1}{2} f_1 (1 + r) \right) \left( \frac{1}{2} m_1 (1 - r) + m_2 \right) + \left( \frac{1}{2} f_1 (1 - r) + f_2 \right) \left( m_0 + \frac{1}{2} m_1 (1 + r) \right)
\]

\[
c_2 = \left( \frac{1}{2} f_1 (1 - r) + f_2 \right) \left( \frac{1}{2} m_1 (1 - r) + m_2 \right)
\]

Then, for each source \( s_i \) we assume there is one copy of the IBD to start (i.e. the probabilities are \([0,1,0]\)). In the case of a couple source, we arbitrarily choose one parent to have the single copy of the IBD segment. We maintain a queue of individuals, which starts out with the children of the source. For each individual in the queue, we compute the probabilities of IBD transmission given their parent probabilities, and then add their children to the queue. Whenever we reach a cohort individual \( a \), we retain \( a_1 \) if the individual had one copy and \( a_2 \) if they had two copies (we know the individual had at least one copy since they are in the cohort for this IBD). Finally, we compute the average of these retained probabilities, which we denote \( P(s_i) \). To choose a source, we take the max:

\[
s^* = \arg \max_i P(s_i)
\]

Then the algorithm proceeds as follows – if this was a bad source (conflicts with genotyped or reconstructed individuals), we still remove it and choose the source with the next highest probability. Also note that these probabilities do not sum to 1, but could be normalized to do so. Finally, note that these probabilities are approximations of the complete probability for a particular configuration, as they do not include the probability the IBD segment was not transmitted to genotyped individuals outside the cohort.