Research Article

Path-Counting Formulas for Generalized Kinship Coefficients and Condensed Identity Coefficients

En Cheng¹ and Z. Meral Ozsoyoglu²

¹ Computer Science Department, The University of Akron, Akron, OH 44325, USA
² Electrical Engineering and Computer Science Department, Case Western Reserve University, 10900 Euclid Avenue, Cleveland, OH 44106, USA

Correspondence should be addressed to En Cheng; echang@uakron.edu

Received 14 January 2014; Accepted 8 May 2014; Published 21 July 2014

Abstract

An important computation on pedigree data is the calculation of condensed identity coefficients, which provide a complete description of the degree of relatedness of two individuals. The applications of condensed identity coefficients range from genetic counseling to disease tracking. Condensed identity coefficients can be computed using linear combinations of generalized kinship coefficients for two, three, four individuals, and there are recursive formulas for computing those generalized kinship coefficients (Karigl, 1981). Path-counting formulas have been proposed for the (generalized) kinship coefficients for two (three) individuals but there have been no path-counting formulas for the other generalized kinship coefficients. It has also been shown that the computation of the (generalized) kinship coefficients for two (three) individuals using path-counting formulas is efficient for large pedigrees, together with path encoding schemes tailored for pedigree graphs. In this paper, we propose a framework for deriving path-counting formulas for generalized kinship coefficients. Then, we present the path-counting formulas for all generalized kinship coefficients for which there are recursive formulas and which are sufficient for computing condensed identity coefficients. We also perform experiments to compare the efficiency of our method with the recursive method for computing condensed identity coefficients on large pedigrees.

1. Introduction

With the rapidly expanding field of medical genetics and genetic counseling, genealogy information is becoming increasingly abundant. In January 2009, the US Department of Health and Human Services released an updated and improved version of the Surgeon General’s Web-based family health history tool [1]. This Web-based tool makes it easy for users to record their family health history. Large extended human pedigrees are very informative for linkage analysis. Pedigrees including thousands of members in 10-20 generations are available from genetically isolated populations [2, 3]. In human genetics, a pedigree is defined as “a simplified diagram of a family’s genealogy that shows family members’ relationships to each other and how a specific trait, abnormality, or disease has been inherited” [4]. Pedigrees are utilized to trace the inheritance of a specific disease, calculate genetic risk ratios, identify individuals at risk, and facilitate genetic counseling. To calculate genetic risk ratios or identify individuals at risk, we need to assess the degree of relatedness of two individuals. As a matter of fact, all measures of relatedness are based on the concept of identical by descent (IBD). Two alleles are identical by descent if one is an ancestral copy of the other or if they are both copies of the same ancestral allele. The IBD concept is primarily due to Cotterman [5] and Malecot [6] and has been successfully applied to many problems in population genetics.

The simplest measure of relationship between two individuals is their kinship coefficient. The kinship coefficient between two individuals $i$ and $j$ is the probability that an allele selected randomly from $i$ and an allele selected randomly from the same autosomal locus of $j$ are identical by descent. To better discriminate between different types of pairs of relatives, identity coefficients were introduced by Gillois [7] and
Harris [8] and promulgated by Jacquard [9]. Considering the four alleles of two individuals at a fixed autosomal locus, there are 15 possible identity states. Disregarding the distinction between maternally and paternally derived alleles, we obtain 9 condensed identity states. The probabilities associated with each condensed identity state are called condensed identity coefficients, which are useful in a diverse range of fields. This includes the calculation of risk ratios for qualitative disease, the analysis of quantitative traits, and genetic counseling in medicine.

A recursive algorithm for calculating condensed identity coefficients proposed by Karigl [10] has been known for some time. This method requires that one calculates a set of generalized kinship coefficients, from which one obtains condensed identity coefficients via a linear transformation. One limitation is that this recursive approach is not scalable when applied to very large pedigrees. It has been previously shown that the kinship coefficients for two individuals [11–13] and the generalized kinship coefficients for three individuals [14, 15] can be efficiently calculated using path-counting formulas together with path encoding schemes tailored for pedigree graphs.

Motivated by the efficiency of path-counting formulas for computing the kinship coefficient for two individuals and the generalized kinship coefficient for three individuals, we first introduce a framework for developing path-counting formulas to compute generalized kinship coefficients concerning three individuals, four individuals, and two pairs of individuals. Then, we present path-counting formulas for all generalized kinship coefficients which have recursive formulas proposed by Karigl [10] and are sufficient to compute condensed identity coefficients. In summary, our ultimate goal is to use path-counting formulas for generalized kinship coefficients computation so that efficiency and scalability for condensed identity coefficients calculation can be improved.

The main contributions of our work are as follows:

(i) a framework to develop path-counting formulas for generalized kinship coefficients;
(ii) a set of path-counting formulas for all generalized kinship coefficients having recursive formulas [10];
(iii) experimental results demonstrating significant performance gains for calculating condensed identity coefficients based on our proposed path-counting formulas as compared to using recursive formulas [10].

2. Materials and Methods

This section describes kinship coefficients and generalized kinship coefficients, identity coefficients, and condensed identity coefficients in more detail. Conceptual terms for the path-counting formulas for three and four individuals are introduced in Section 2.3. In addition, an overview of path-counting formula derivation is presented.

2.1. Kinship Coefficients and Generalized Kinship Coefficients. The kinship coefficient between two individuals \( a \) and \( b \) is the probability that a randomly chosen allele at the same locus from each is identical by descent (IBD). There are two approaches to computing the kinship coefficient \( \Phi_{ab} \): the recursive approach [10] and the path-counting approach [16]. The recursive formulas [10] for \( \Phi_{ab} \) and \( \Phi_{aa} \) are

\[
\Phi_{ab} = \frac{1}{2} (\Phi_{fa} + \Phi_{mb}) \quad \text{if } a \text{ is not an ancestor of } b, \\
\Phi_{aa} = \frac{1}{2} (1 + \Phi_{fm}) = \frac{1}{2} (1 + F_a),
\]

where \( f \) and \( m \) denote the father and the mother of \( a \), respectively, and \( F_a \) is the inbreeding coefficient of \( a \).

Wright’s path-counting formula [16] for \( \Phi_{ab} \) is

\[
\Phi_{ab} = \sum_{A} \sum_{\langle P_{sa}, P_{sb} \rangle \in PP} \left( \frac{1}{2} \right)^{r+1} (1 + F_A),
\]

where \( A \) is a common ancestor of \( a \) and \( b \), \( PP \) is a set of non-overlapping path-pairs \( \langle P_{sa}, P_{sb} \rangle \) from \( A \) to \( a \) and \( b \), \( r \) is the length of the path \( P_{sa} \), \( s \) is the length of the path \( P_{sb} \), and \( F_A \) is the inbreeding coefficient of \( A \). The path-pair \( \langle P_{sa}, P_{sb} \rangle \) is non-overlapping if and only if the two paths share no common individuals, except \( A \).

Recursive formulas proposed by Karigl [10] for generalized kinship coefficients concerning three individuals, four individuals, and two pairs of individuals are listed as follows in (3), (4), and (5):

\[
\Phi_{abc} = \frac{1}{2} (\Phi_{fbc} + \Phi_{mbc}) \\
\quad \text{if } a \text{ is not an ancestor of } b \text{ or } c,
\]

\[
\Phi_{aab} = \frac{1}{2} (\Phi_{ab} + \Phi_{fmb}) \\
\quad \text{if } a \text{ is not an ancestor of } b,
\]

\[
\Phi_{aaa} = \frac{1}{4} (1 + 3 \Phi_{fm}) = \frac{1}{4} (1 + 3 F_a),
\]

\[
\Phi_{abcd} = \frac{1}{2} (\Phi_{fbc} + \Phi_{mbcd}) \\
\quad \text{if } a \text{ is not an ancestor of } b \text{ or } c \text{ or } d,
\]

\[
\Phi_{asbc} = \frac{1}{2} (\Phi_{abc} + \Phi_{fmb}) \\
\quad \text{if } a \text{ is not an ancestor of } b \text{ or } c,
\]

\[
\Phi_{asab} = \frac{1}{4} (\Phi_{ab} + 3 \Phi_{mb}) \\
\quad \text{if } a \text{ is not an ancestor of } b,
\]

\[
\Phi_{aasa} = \frac{1}{8} (1 + 7 \Phi_{fm}) = \frac{1}{8} (1 + 7 F_a).
\]
Φ_{abc} = \frac{1}{2} \left( Φ_{fbc} + Φ_{mbc} \right)
if a is not an ancestor of b or c, 

Φ_{aba} = \frac{1}{2} \left( Φ_{ab} + Φ_{fmb} \right)
if a is not an ancestor of b or c, 

Φ_{aaa} = \frac{1}{4} \left( 1 + 3Φ_{fm} \right) = \frac{1}{4} \left( 1 + 3F_a \right).

(5)

Φ_{abc} is the probability that randomly chosen alleles at the same locus from each of the three individuals (i.e., a, b, and c) are identical by descent (IBD). Similarly, Φ_{abed} is the probability that randomly chosen alleles at the same locus from each of the four individuals (i.e., a, b, c, and d) are IBD. Φ_{ab,cd} is the probability that a random allele from a is IBD with a random allele from b and that a random allele from c is IBD with a random allele from d at the same locus. Note that Φ_{abc} = 0 if there is no common ancestor of a, b, and c. Φ_{ab,cd} = 0 if there is no common ancestor of a, b, c, and d, and Φ_{ab,cd} = 0 in the absence of a common ancestor either for a and b or for c and d.

2.2. Identity Coefficients and Condensed Identity Coefficients. Given two individuals a and b with maternally and paternally derived alleles at a fixed autosomal locus, there are 15 possible identity states, and the probabilities associated with each identity state are called identity coefficients. Ignoring the distinction between maternally and paternally derived alleles, we categorize the 15 possible states to 9 condensed identity states, as shown in Figure 1. The states range from state 1, in which all four alleles are IBD, to state 9, in which none of the four alleles are IBD. The probabilities associated with each condensed identity state are called condensed identity coefficients, denoted by \{\Delta_i \mid 1 \leq i \leq 9 \}. The condensed identity coefficients can be computed based on generalized kinship coefficients using the linear transformation shown as follows in (6):

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 \\
2 & 2 & 1 & 1 & 2 & 2 & 1 & 1 & 1 \\
4 & 0 & 2 & 0 & 2 & 0 & 2 & 1 & 0 \\
8 & 0 & 4 & 0 & 2 & 0 & 2 & 1 & 0 \\
8 & 0 & 2 & 0 & 4 & 0 & 2 & 1 & 0 \\
16 & 0 & 4 & 0 & 4 & 0 & 2 & 1 & 0 \\
4 & 4 & 2 & 2 & 2 & 2 & 1 & 1 & 1 \\
16 & 0 & 4 & 0 & 4 & 0 & 4 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
\Delta_1 \\
\Delta_2 \\
\Delta_3 \\
\Delta_4 \\
\Delta_5 \\
\Delta_6 \\
\Delta_7 \\
\Delta_8 \\
\Delta_9
\end{bmatrix}
= \begin{bmatrix}
1 \\
2Φ_{aa} \\
2Φ_{ab} \\
4Φ_{ab} \\
8Φ_{abab} \\
8Φ_{abab} \\
16Φ_{ababb} \\
16Φ_{abab}
\end{bmatrix}.

(6)

In our work, we focus on deriving the path-counting formulas for the generalized kinship coefficients, including Φ_{abc}, Φ_{abed}, and Φ_{ab,cd}.

2.3. Terms Defined for Path-Counting Formulas for Three and Four Individuals

(1) Triple-Common Ancestor. Given three individuals a, b, and c, if A is a common ancestor of the three individuals, then we call A a triple-common ancestor of a, b, and c.

(2) Quad-Common Ancestor. Given four individuals a, b, c, and d, if A is a common ancestor of the four individuals, then we call A a quad-common ancestor of a, b, c, and d.

(3) \( P(A, a) \). It denotes the set of all possible paths from A to a, where the paths can only traverse edges in the direction of parent to child such that \( P(A, a) \neq NULL \) if and only if A is an ancestor of a. \( P_{Ac} \) denotes a particular path from A to a, where \( P_{Ac} \in P(A, a) \).

(4) Path-Pair. It consists of two paths, denoted as \( \langle P_{Ac}, P_{Ab} \rangle \), where \( P_{Ac} \in P(A, a) \) and \( P_{Ab} \in P(A, b) \).

(5) Nonoverlapping Path-Pair. Given a path-pair \( \langle P_{Ac}, P_{Ab} \rangle \), it is nonoverlapping if and only if the two paths share no common individuals, except A.

(6) Path-Triple. It consists of three paths, denoted as \( \langle P_{Ac}, P_{Ab}, P_{Ac} \rangle \), where \( P_{Ac} \in P(A, a) \), \( P_{Ab} \in P(A, b) \), and \( P_{Ac} \in P(A, c) \).

(7) Path-Quad. It consists of four paths, denoted as \( \langle P_{Ac}, P_{Ab}, P_{Ac}, P_{Ad} \rangle \), where \( P_{Ac} \in P(A, a) \), \( P_{Ab} \in P(A, b) \), \( P_{Ac} \in P(A, c) \), and \( P_{Ad} \in P(A, d) \).

(8) Bi_C(P_{Ac}, P_{Ab}). It denotes all common individuals shared between \( P_{Ac} \) and \( P_{Ab} \), except A.

(9) Tri_C(P_{Ac}, P_{Ab}, P_{Ac}). It denotes all common individuals shared among \( P_{Ac} \), \( P_{Ab} \), and \( P_{Ac} \), except A.

(10) Quad_C(P_{Ac}, P_{Ab}, P_{Ac}, P_{Ad}). It denotes all common individuals shared among \( P_{Ac} \), \( P_{Ab} \), \( P_{Ac} \), and \( P_{Ad} \), except A.

(11) Crossover and 2-Overlap Individual. If \( s \in Bi_C(P_{Ac}, P_{Ab}) \), we call s a crossover individual with respect to \( P_{Ac} \) and \( P_{Ab} \) if the two paths pass through different parents of s. On the other hand, if \( P_{Ac} \) and \( P_{Ab} \) pass through the same parent of s, then we call s a 2-overlap individual with respect to \( P_{Ac} \) and \( P_{Ab} \).

(12) 3-Overlap Individual. If \( s \in Tri_C(P_{Ac}, P_{Ab}, P_{Ac}) \) and the three paths \( P_{Ac} \), \( P_{Ab} \), and \( P_{Ac} \) pass through the same parent of s, then we call s a 3-overlap individual with respect to \( P_{Ac} \), \( P_{Ab} \), and \( P_{Ac} \).

(13) 2-Overlap Path. If s is a 2-overlap individual with respect to \( P_{Ac} \) and \( P_{Ab} \), then both \( P_{Ac} \) and \( P_{Ab} \) pass through the same parent of s, denoted by p, and the edge from p to s is called an overlap edge. All consecutive overlap edges constitute a path and this path is called a 2-overlap path. If the 2-overlap path
extends all the way to the ancestor $A$, we call it a root 2-overlap path.

(14) 3-Overlap Path. It consists of all 3-overlap individuals in a consecutive order. If the 3-overlap path extends all the way to the root $A$, we call it a root 3-overlap path.

Example 1. Consider the path-pairs from $A$ to $a$ and $b$ in Figure 2, where $A$ is a common ancestor of $a$ and $b$. For path-pair1, $Bi_C(P_{aa}, P_{ab}) = \{s,e,t\}$, and $A \rightarrow s \rightarrow e \rightarrow t \rightarrow a$ is a root 2-overlap path with respect to $P_{aa}$ and $P_{ab}$. For path-pair4, $Bi_C(P_{aa}, P_{ab}) = \{e,t\}$, where $e$ is a crossover individual; $t$ is a 2-overlap individual with respect to $P_{aa}$ and $P_{ab}$, and $e \rightarrow t$ is a root 2-overlap path with respect to $P_{aa}$ and $P_{ab}$.

Example 2. There are four path-quads listed in Figure 3, from $A$ to four individuals $a$, $b$, $c$, and $d$, where $A$ is a quad-common ancestor of the four individuals. For path-quad2, considering the paths $P_{aa}$ and $P_{ab}$, the path $A \rightarrow t \rightarrow f \rightarrow s$ is a root 2-overlap path; $\{t, f, s\}$ are 2-overlap individuals with respect to $P_{aa}$ and $P_{ab}$. For path-quad3, $\{t, f, s\}$ are 3-overlap individuals with respect to $P_{aa}$, $P_{ab}$, and $P_{ac}$, and the path $A \rightarrow t \rightarrow f \rightarrow s$ is a root 3-overlap path.

Then, we summarize all the conceptual terms used in the path-counting formulas for two individuals, three individuals, and four individuals in Table 1 which reveals a glimpse of our framework for generalizing Wright’s formula to three and four individuals from terminology aspect.

2.4. An Overview of Path-Counting Formula Derivation. According to Wright’s path-counting formula [16] (see (2)) for two individuals $a$ and $b$, the path-counting approach requires identifying common ancestors of $a$ and $b$ and calculating the contribution of each common ancestor to $\Phi_{ab}$. More specifically, for each common ancestor, denoted as $A$, we obtain all path-pairs from $A$ to $a$ and $b$ and identify acceptable path-pairs. For $\Phi_{ab}$, an acceptable path-pair $\langle P_{aa}, P_{ab} \rangle$ is a nonoverlapping path-pair where...
Figure 3: Examples of path-quads.

Table 1: The conceptual terms used for two, three, and four individuals.

<table>
<thead>
<tr>
<th>Two individuals</th>
<th>Three individuals</th>
<th>Four individuals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common ancestor</td>
<td>Path-pair</td>
<td>Quad-common ancestor</td>
</tr>
<tr>
<td>Path-pair</td>
<td>Path-triple</td>
<td>Path-pair</td>
</tr>
<tr>
<td>$Bi(C(P_{Aa}, P_{Ab})$</td>
<td>$Tri(C(P_{Aa}, P_{Ab}, P_{Ac})$</td>
<td>$Quad(C(P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad})$</td>
</tr>
<tr>
<td>N/A</td>
<td>3-Overlap individual</td>
<td>3-Overlap individual</td>
</tr>
<tr>
<td>N/A</td>
<td>2-Overlap path</td>
<td>Crossover individual</td>
</tr>
<tr>
<td>N/A</td>
<td>Root 2-overlap path</td>
<td>Root 3-overlap path</td>
</tr>
<tr>
<td>N/A</td>
<td>Crossover individual</td>
<td>Crossover individual</td>
</tr>
</tbody>
</table>

the two paths share no common individuals, except $A$. In Figure 2, path-pair2 is an acceptable path-pair, while path-pair1, path-pair3, and path-pair4 are not acceptable path-pairs. The contribution of each common ancestor $A$ to $\Phi_{ab}$ is computed based on the inbreeding coefficient of $A$, modified by the length of each acceptable path-pair.

To compute $\Phi_{abc}$, the path-counting approach requires identifying all triple-common ancestors of $a$, $b$, and $c$ and summing up all triple-common ancestors’ contributions to $\Phi_{abc}$. For each triple-common ancestor, denoted as $A$, we first identify all path-triples each of which consists of three paths from $A$ to $a$, $b$, and $c$, respectively. Some examples of path-triples are presented in Figure 2.

For $\Phi_{abc}$, only nonoverlapping path-pairs are acceptable. A path-triple $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$ consists of three path-pairs $\langle P_{Aa}, P_{Ab} \rangle$, $\langle P_{Aa}, P_{Ac} \rangle$, and $\langle P_{Ab}, P_{Ac} \rangle$. For $\Phi_{abc}$, a path-triple might be acceptable even though either 2-overlap individuals or crossover individuals exist between a path-pair. The main challenge we need to address is finding necessary and sufficient conditions for acceptable path-triples.

Aiming at solving the problem of identifying acceptable path-triples, we first use a systematic method to generate all possible cases for a path-pair by considering different types of common individuals shared between the two paths. Then, we introduce building blocks which are connected graphs with conditions on every edge in the graph that encapsulates a set of acceptable cases of path-pairs. In each building block, we represent paths as nodes and interactions (i.e., shared common individuals between two paths) as edges. There are at least two paths in a building block. For each building block, we obtain all acceptable cases for concerned path-pairs. Given a path-triple, it can be decomposed to one or multiple building blocks. Considering a shared path-pair between two building blocks, we use the natural join operator from relational algebra to match the acceptable cases for the shared path-pair between two building blocks. In other words, considering the acceptable cases for building blocks as inputs, we use the natural join operator to construct all acceptable cases for a path-triple. Acceptable cases for a path-triple are identified and then used in deriving the path-counting formula for $\Phi_{abc}$.

Then, we summarize all the main procedures used for deriving the path-counting formula for $\Phi_{abc}$ in a flowchart shown in Figure 4. The main procedures are also applicable for deriving the path-counting formulas for $\Phi_{abcd}$ and $\Phi_{abc}$.

3. Results and Discussion

3.1. Path-Counting Formulas for Three Individuals. We first introduce a systematic method to generate all possible cases
for a path-pair. Then we discuss building blocks for path-triples and identify all acceptable cases which are used in deriving the path-counting formula for $\Phi_{abc}$.

3.1.1. Cases for a Path-Pair. Given a path-pair $\langle P_{Aa}, P_{Ab} \rangle$ with $Bi.C(P_{Aa}, P_{Ab}) \neq \text{NULL}$, where $A$ is a common ancestor of $a$ and $b$ and $Bi.C(P_{Aa}, P_{Ab})$ consists of all common individuals shared between $P_{Aa}$ and $P_{Ab}$ except $A$, we introduce three patterns (i.e., crossover, 2-overlap, and root 2-overlap) to generate all possible cases for $\langle P_{Aa}, P_{Ab} \rangle$.

(1) $X(P_{Aa}, P_{Ab})$: $P_{Aa}$ and $P_{Ab}$ share one or multiple crossover individuals.

(2) $T(P_{Aa}, P_{Ab})$: $P_{Aa}$ and $P_{Ab}$ are root 2-overlapping from $A$, and the root 2-overlap path can have one or multiple 2-overlap individuals.

(3) $Y(P_{Aa}, P_{Ab})$: $P_{Aa}$ and $P_{Ab}$ are overlapping but not from $A$, and the 2-overlap path can have one or multiple 2-overlap individuals.

Based on the three patterns, $X(P_{Aa}, P_{Ab})$, $T(P_{Aa}, P_{Ab})$, and $Y(P_{Aa}, P_{Ab})$, we use regular expressions to generate all possible cases for the path-pair $\langle P_{Aa}, P_{Ab} \rangle$. For convenience, we drop $\langle P_{Aa}, P_{Ab} \rangle$ and use $X$, $T$, and $Y$ instead of patterns $X(P_{Aa}, P_{Ab})$, $T(P_{Aa}, P_{Ab})$, and $Y(P_{Aa}, P_{Ab})$, whenever there is no confusion. When $Bi.C(P_{Aa}, P_{Ab}) \neq \text{NULL}$, the eight cases shown in (7) cover all possible cases for $\langle P_{Aa}, P_{Ab} \rangle$. The completeness of eight cases shown in (7) for $\langle P_{Aa}, P_{Ab} \rangle$ can be proved by induction on the total number of $T$, $X$, and $Y$ appearing in $\langle P_{Aa}, P_{Ab} \rangle$. Using the pedigree in Figure 2, Cases 1–3 and Case 6 are illustrated in (8), (9), (10), and (11):

\[
\begin{align*}
\text{Case 1: } & T \\
\text{Case 2: } & X^+ \\
\text{Case 3: } & TX^+ \\
\text{Case 4: } & T(X^+X^+), \\
\text{Case 5: } & T(X^+X^+)^+ X^+ \\
\text{Case 6: } & X^+Y \\
\text{Case 7: } & X^+(YX^+)^+ \\
\text{Case 8: } & X^+(YX^+)^+ Y,
\end{align*}
\]

\[
\begin{align*}
A \rightarrow s \rightarrow e \rightarrow t \rightarrow a \in T, \\
A \rightarrow s \rightarrow e \rightarrow t \rightarrow b \in T.
\end{align*}
\]
where \(s, e, t\) are 2-overlap individuals and the overlap path is a root 2-overlap path:

\[
A \rightarrow s \rightarrow e \rightarrow t \rightarrow a \\
A \rightarrow s \rightarrow f \rightarrow t \rightarrow b \in TX,
\]

(9)

where \(s\) is a 2-overlap individual and the overlap path is a root 2-overlap path; \(t\) is a crossover individual:

\[
A \rightarrow s \rightarrow e \rightarrow t \rightarrow a \\
A \rightarrow d \rightarrow f \rightarrow t \rightarrow b \in X,
\]

(10)

where \(t\) is a crossover individual:

\[
A \rightarrow c \rightarrow e \rightarrow t \rightarrow a \\
A \rightarrow s \rightarrow e \rightarrow t \rightarrow b \in XY,
\]

(11)

where \(e\) is a crossover individual; \(t\) is a 2-overlap individual and the overlap path is a 2-overlap path.

### 3.1.2. Path-Pair Level Graphical Representation of a Path-Triple

Given a path-triple \((P_{Aa}, P_{Ab}, P_{Ac})\), we represent each path as a node. The path-triple can be decomposed to three path-pairs (i.e., \(P_{Aa}, P_{Ab}, P_{Ac}\)). For each path-pair, if the two paths share at least one common individual (i.e., either 2-overlap individual or crossover individual), except \(A\), then there is an edge between the two nodes representing the two paths. Therefore, we obtain four different scenarios \(S_0\)–\(S_3\), shown in Figure 5.

In Figure 5, the scenario \(S_0\) has no edges, so it means that \((P_{Aa}, P_{Ab}, P_{Ac})\) consists of three independent paths. In Figure 2, path-triple is an example of \(S_0\). Next, we introduce a lemma which can assist with identifying the options for the edges in the scenarios \(S_1\)–\(S_3\).

**Lemma 3.** Given a path-triple \((P_{Aa}, P_{Ab}, P_{Ac})\), consider the three path-pairs \((P_{Aa}, P_{Ab}), (P_{Aa}, P_{Ac}), \) and \((P_{Ab}, P_{Ac})\), if there is a 2-overlap edge which is represented by \(Y\) in regular expression representation of any of the three path-pairs, and then the path-triple \((P_{Aa}, P_{Ab}, P_{Ac})\) has no contribution to \(\Phi_{abc}\).

**Proof.** In [17], Nadot and Vaysseix proposed, from a genetic and biological point of view, that \(\Phi_{abc}\) can be evaluated by enumerating all eligible inheritance paths at allele-level starting from a triple common ancestor \(A\) to the three individuals \(a, b, \) and \(c\).

For the pedigree in Figure 6, let us consider the path-triple \((P_{Aa}, P_{Ab}, P_{Ac})\) listed as follows, \(P_{Aa} : A \rightarrow a; P_{Ab} : A \rightarrow p_3 \rightarrow p_6 \rightarrow p_7 \rightarrow b; P_{Ac} : A \rightarrow p_4 \rightarrow p_6 \rightarrow p_7 \rightarrow c\).

For \((P_{Ab}, P_{Ac}), p_6\) is a crossover individual, \(p_7\) is an overlap individual, and \(p_6 \rightarrow p_7\) is a 2-overlap edge represented by \(Y\) in regular expression representation (see the definition for \(Y\) in Section 3.1.1).

For the individual \(p_6\), let us denote the two alleles at one fixed autosomal locus as \(g_1\) and \(g_2\). At allele-level, only one allele can be passed down from \(p_6\) to \(p_7\). Since \(p_3\) and \(p_4\) are parents of \(p_6\), \(g_1\) is passed down from one parent, and \(g_2\) is passed down from the other parent. It is infeasible to pass down both \(g_1\) and \(g_2\) from \(p_6\) to \(p_7\). In other words, there are no corresponding inheritance paths for the path-triple \((P_{Ab}, P_{Ac})\) with a 2-overlap edge between \((P_{Ab}, P_{Ac})\) (i.e., Case 6: \(XY\)). Therefore, such kind of path-triples has no contribution to \(\Phi_{abc}\).

**Figure 5:** A path-pair level graphical representation of \((P_{Aa}, P_{Ab}, P_{Ac})\).

**Figure 6:** Examples of pedigree and inheritance paths.
For $B_2$, there can be at most one edge belonging to root overlap (either $T$ or $TX$).

For $B_2$, there can be at most one edge belonging to root overlap (either $T$ or $TX$).

Figure 7: Building blocks $\{B_1, B_2\}$ and basic rules.

$$P_{Aa} \text{ and } P_{Ac} \text{ must share at least one common individual, except } A, \text{ which contradicts the fact that } P_{Aa} \text{ and } P_{Ac} \text{ have no edge.}$$

Next, we focus on generating all acceptable cases for the scenarios $S_1$-$S_3$ in Figure 5, where only $S_3$ contains more than one building block. In order to leverage the dependency among building blocks, we decompose $S_3$ into $S_j = \{u_1 = B_2, u_2 = B_2, u_3 = B_2\}$, shown in Figure 8. For each $u_i$, we have a set of acceptable path-triples, denoted as $R_i$.

Next, we introduce a canonical graph which results from applying the splitting operator for all crossover individuals. The canonical graph has zero crossover individual.

Lemma 4. Given a pedigree graph $G_{k+1}$ having $(k+1)$ crossover individuals regarding $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$, shown in Figure 9, let $s$ denote the lowest crossover individual, where no descendant of $s$ can be a crossover individual among the three paths $P_{Aa}, P_{Ab},$ and $P_{Ac}$. After using the splitting operator for the lowest crossover individual $s$ in $G_{k+1}$, the number of crossover individuals in $G_{k+1}$ is decreased by 1.

Proof. The splitting operator only affects the edges from $s$ to $a'$ and $b'$. If there is a new crossover node appearing, the only possible node is either $a'$ or $b'$. Assume $b'$ becomes a crossover individual; it means that $b'$ is able to reach $a$ and $b$ from two separate paths. It contradicts the fact that $s$ is the lowest crossover individual between $P_{Aa}$ and $P_{Ab}$.

Definition 5 (Canonical Graph). Given a pedigree graph $G$ having one or more crossover individuals regarding $\Phi_{abc}$, if there exists a graph $G'$ which has no crossover individuals with regards to $\Phi_{abc}$ such that

(i) any acceptable path-triple in $G$ has an acceptable path-triple in $G'$ which has the same contribution to $\Phi_{abc}$ as the one in $G$ for $\Phi_{abc}$;

(ii) any acceptable path-triple in $G'$ has an acceptable path-triple in $G$ which and has the same contribution to $\Phi_{abc}$ as the one in $G'$ for $\Phi_{abc}$.

We call $G'$ a canonical graph of $G$ regarding $\Phi_{abc}$.

Lemma 6. For a pedigree graph $G$ having one or more crossover individuals regarding $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$, there exists a canonical graph $G'$ for $G$. 

For $B_2$, there can be at most one edge belonging to root overlap (either $T$ or $TX$).
Computational and Mathematical Methods in Medicine 9

Ancestor-descendant relationship
Parent-child relationship

For computational and mathematical methods in medicine, we obtain $T_2 = R_1 \times R_2$ where $\times$ denotes the Cartesian product operator from relational algebra.

3.1.5. Path-Counting Formula for $\Phi_{abc}$. Now, we present the path-counting formula for $\Phi_{abc}$:

$$
\Phi_{abc} = \sum_A \left[ \sum_{Type \ 1} \left( \frac{1}{2} \right)^{L_{\text{triple}}} \Phi_{A,AA} \right] + \sum_{Type \ 2} \left( \frac{1}{2} \right)^{L_{\text{triple}+1}} \Phi_{AA},
$$

(12)

where $\Phi_{AA} = (1/2)(1 + F_a), \Phi_{AAA} = (1/4)(1 + 3F_a), F_a$: the inbreeding coefficient of $A, A$: a triple-common ancestor of $a, b, and c$. Type 1: $\langle P_{Ab}, P_{Ac}, P_{Ac} \rangle$ has zero root 2-overlap. Type 2: $\langle P_{Ab}, P_{Ac}, P_{Ac} \rangle$ has one root 2-overlap path $P_{Ac}$ ending at the individual $s$:

$$
L_{\text{triple}} = \begin{cases} 
L_{P_{AA}} + L_{P_{Ab}} + L_{P_{Ac}} & \text{for Type 1} \\
L_{P_{AA}} + L_{P_{Ab}} + L_{P_{Ac}} - L_{P_{Ac}} & \text{for Type 2},
\end{cases}
$$

(13)

and $L_{P_{Ac}}$: the length of the path $P_{Ac}$ (also applicable for $P_{Ab}, P_{Ac},$ and $P_{Ac}$).

For completeness, the path-counting formula for $\Phi_{abc}$ is given in Appendix A; and the correctness proof of the path-counting formula is given in Appendix B.

3.2. Path-Counting Formulas for Four Individuals

3.2.1. Path-Pair Level Graphical Representation of $\langle P_{Ab}, P_{Ab}, P_{Ac}, P_{Ac} \rangle$. Given a path-quad $\langle P_{Ab}, P_{Ab}, P_{Ac}, P_{Ac} \rangle$ and $\text{Quad}_C(P_{Ab}, P_{Ab}, P_{Ac}, P_{Ac}) = 0$, the path-quad can have 11 scenarios $S_1$–$S_{10}$ shown in Figure 10 where all four paths are considered symmetrically.

In Figure 11, we introduce three building blocks $[B_1, B_2, B_3]$. For $B_1$ and $B_2$, the rules presented in Figure 7 are also applicable for Figure 11. For $B_3$, we only consider root overlap, because the crossover individuals can be eliminated by using the splitting operator introduced in Section 3.1.4. Note that for $B_3$, if $\text{Tri}_C(P_{Ab}, P_{Ab}, P_{Ac}) = 0$, then it is equivalent to the scenario $S_3$ in Figure 8 Therefore, we only need to consider $B_3$ when $\text{Tri}_C(P_{Ab}, P_{Ab}, P_{Ac}) \neq 0$.

3.2.2. Building Block-Based Cases Construction for $\langle P_{Ab}, P_{Ab}, P_{Ac}, P_{Ac} \rangle$. For a scenario $S_i (0 \leq i \leq 10)$ in Figure 11, we first decompose $S_i$ to one or multiple building blocks. For a scenario $S_i \in \{S_1, S_3, \ldots, S_{10}\}$, it has only one building block, and all acceptable cases can be obtained directly. For $S_3 = \{u_1 = B_1, u_2 = B_1\}$, there is no need to consider the conflict between the edges in $u_1$ and $u_2$ because $u_1$ and $u_2$ are disconnected. Let $R_i$ denote all acceptable cases of the path-pairs in $u_i$, and let $T_i$ denote all acceptable cases for $S_i$. Therefore, we obtain $T_2 = R_1 \times R_2$ where $\times$ denotes the Cartesian product operator from relational algebra.
For $S_j$, we define the largest subgraph of $S_j$ as follows:

1. $S_j$ is a proper subgraph of $S_j$;
2. If $S_j$ contains $B_j$, then $S_j$ must also contain $B_3$;
3. No such $S_k$ exists that $S_j$ is a proper subgraph of $S_k$ while $S_k$ is also a proper subgraph of $S_j$.

For each scenario $S_j$ ($4 \leq i \leq 10$ and $i \neq 6$), we list the largest subgraph of $S_j$, denoted as $S_j$, in Table 2.

For a scenario $S_j$ ($4 \leq i \leq 10$ and $i \neq 6$), let $\text{Diff}(S_j \setminus S_j)$ denote the set of building blocks in $S_j$ but not in $S_j$, where $S_j$ is the largest subgraph of $S_j$. Let $|E_j|$ and $|E_j|$ denote the number of edges in $S_j$ and $S_j$, respectively. According to Table 2, we can conclude that $|E_j| = |E_j| = 1$. In order to leverage the dependency among building blocks, we consider only $B_j$ in $\text{Diff}(S_j \setminus S_j)$. For example, $\text{Diff}(S_5 \setminus S_5) = \{B_3\}$. Let $T_j$ denote all acceptable cases for $S_j$. And let $R_j$ denote the set of acceptable cases for $\text{Diff}(S_j \setminus S_j)$. Then, we can use $S_j$ and $\text{Diff}(S_j \setminus S_j)$ to construct all acceptable cases for $S_j$. Then, we apply this idea for constructing all acceptable cases for each $S_j$ in Table 2.

Given a path-block $(P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad})$, an acceptable case has the following properties:

1. If there is one root 3-overlap path, there can be at most one root 2-overlap path;
2. Otherwise, there can be at most two root 2-overlap paths.

### Table 2: Largest subgraph of a scenario $S_j$ ($4 \leq i \leq 10$ and $i \neq 6$).

<table>
<thead>
<tr>
<th>$S_j$</th>
<th>$S_4$</th>
<th>$S_5$</th>
<th>$S_7$</th>
<th>$S_8$</th>
<th>$S_9$</th>
<th>$S_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_j$</td>
<td>$S_3$</td>
<td>$S_5$</td>
<td>$S_6$</td>
<td>$S_7$</td>
<td>$S_7$</td>
<td>$S_8$</td>
</tr>
</tbody>
</table>

### 3.2.3. Path-Counting Formula for $\Phi_{abcd}$

Now, we present the path-counting formula for $\Phi_{abcd}$ as follows:

\[
\Phi_{abcd} = \sum_{A} \left( \sum_{\text{Type } 1} \left( \frac{1}{2} \right)^{L_{quad}} \Phi_{AAAA} + \sum_{\text{Type } 2} \left( \frac{1}{2} \right)^{L_{quad}+1} \Phi_{AAA} + \sum_{\text{Type } 3} \left( \frac{1}{2} \right)^{L_{quad}+2} \Phi_{AA} \right),
\]

where $\Phi_{AA} = (1/2)(1+F_A), \Phi_{AAA} = (1/4)(1+3F_A), \Phi_{AAAA} = (1/8)(1+7F_A)$, $F_A$: the inbreeding coefficient of $A$; a quad-common ancestor of $a, b, c$, and $d$, Type 1: zero root 2-overlap and zero root 3-overlap path, Type 2: one root 2-overlap path $P_{At}$ ending at $s$.

**Case 1:** two root 2-overlap paths $P_{At}$, $P_{At}$ ending at $s_1$ and $s_2$, respectively.
**Case 2:** one root 3-overlap path $P_{At}$ ending at $t$.
**Case 3:** one root 2-overlap path $P_{At}$, one root 3-overlap path $P_{At}$ ending at $s$ and $t$, respectively.

\[
L_{quad} = \left\{
\begin{array}{ll}
L_{Pa} + L_{Pa} + L_{Pa} + L_{Pa} + L_{Pa} & \text{for Type 1} \\
L_{Pa} + L_{Pa} + L_{Pa} + L_{Pa} + L_{Pa} & \text{for Type 2} \\
-L_{Pa} - L_{Pa} - L_{Pa} - L_{Pa} & \text{for Case 1 of Type 3} \\
L_{Pa} + L_{Pa} + L_{Pa} + L_{Pa} & \text{for Case 2 of Type 3} \\
L_{Pa} + L_{Pa} + L_{Pa} + L_{Pa} & \text{for Case 3 of Type 3},
\end{array}
\right.
\]

and $L_{Pa}$: the length of the path $P_{At}$ (also applicable for $P_{Ab}, P_{Ac}, P_{Ad}$, etc.).

For completeness, the path-counting formulas for $\Phi_{abdc}$ and $\Phi_{abcd}$ are presented in Appendix A. The correctness of the path-counting formula for four individuals is proven in Appendix C.
3.3. Path-Counting Formulas for Two Pairs of Individuals

3.3.1. Terminology and Definitions

(1) 2-Pair-Path-Pair. It consists of two pairs of path-pairs denoted as \(\langle (P_{Aa}, P_{Ab}) , (P_{Ac}, P_{Ad}) \rangle\), where \(P_{Aa} \in P(S, a), P_{Ab} \in P(S, b), P_{Ac} \in P(T, c), P_{Ad} \in P(T, d)\), \(S\) is a common ancestor of \(a\) and \(b\), and \(T\) is a common ancestor of \(c\) and \(d\). If \(S = T\), then \(A\) is a quad-common ancestor of \(a, b, c,\) and \(d\).

(2) Homo-Overlap and Heter-Overlap Individual. Given two pairs of individuals \((a, b)\) and \((c, d)\), if \(s \in Bi.C(P_{Aa}, P_{Ab})\) (or \(s \in Bi.C(P_{Ac}, P_{Ad})\)), we call \(s\) a homo-overlap individual when \(P_{Aa}\) and \(P_{Ab}\) (or \(P_{Ac}\) and \(P_{Ad}\)) pass through the same parent of \(s\). If \(r \in Bi.C(P_{Aa}, P_{Ab})\), where \(i \in \{a, b\}\) and \(j \in \{c, d\}\), we call \(r\) a heter-overlap individual when \(P_{Aa}\) and \(P_{Ab}\) pass through the same parent of \(r\).

(3) Root Homo-Overlap and Heter-Overlap Path. Given a 2-pair-path-pair \(\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle\), if \(s\) is a homo-overlap individual and the homo-overlap path extends all the way to the quad-common ancestor \(A\), then we call it a root homo-overlap path. If \(r\) is a heter-overlap individual and the heter-overlap path extends all the way to the quad-common ancestor \(A\), then we call it a root heter-overlap path.

Example 8. \(A\) is a quad-common ancestor for \(a, b, c,\) and \(d\) in Figure 12. For (a), \(s\) is a homo-overlap individual between \(P_{Aa}\) and \(P_{Ab}\).

For (b), \(t\) is a heter-overlap individual between \(P_{Ac}\) and \(P_{Ad}\). And, \(A \rightarrow s\) and \(A \rightarrow t\) are root homo-overlap paths. For (b), \(x\) is a heter-overlap individual between \(P_{Ac}\) and \(P_{Ad}\). \(y\) is a heter-overlap individual between \(P_{Ab}\) and \(P_{Ac}\). And \(A \rightarrow x\) and \(A \rightarrow y\) are root heter-overlap paths.

3.3.2. Path-Counting Formula for \(\Phi_{abcd}\). Now, we present a path-pair graphical level representation for \(\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle\) shown in Figure 13. The options for an edge can be \(\{T, X, TX\}\). (Refer to Section 3.1.1 for definitions of \(T, X,\) and \(TX)\). Based on the different types of \(\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle\) presented in (14), all cases for \(\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle\) are summarized in Table 3, where \(h\) is the last individual of a root homo-overlap path \(P_{Ah}\) (i.e., the path \(P_{Ah}\) ending at \(h\)) and \(r_1\) and \(r_2\) are the last individuals of root heter-overlap paths \(P_{Ari}\) and \(P_{Ari+2}\), respectively.

Given a pedigree graph having one or multiple progenitors \(\{p_i | i > 0\}\), we define that the generation of a progenitor

<table>
<thead>
<tr>
<th>(\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle)</th>
<th>(\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero root 2-overlap and zero root 3-overlap</td>
<td>Zero root homo-overlap and zero root heter-overlap</td>
</tr>
<tr>
<td>One root 2-overlap path</td>
<td>One root homo-overlap and zero root heter-overlap</td>
</tr>
<tr>
<td>Two root 2-overlap paths</td>
<td>Zero root homo-overlap and two root heter-overlaps</td>
</tr>
<tr>
<td>One root 3-overlap path</td>
<td>One root homo-overlap and two root heter-overlaps, and (h = r_1 = r_2)</td>
</tr>
<tr>
<td>One root 2-overlap and one root 3-overlap</td>
<td>One root homo-overlap and two root heter-overlaps, and (r_1 = r_2 \neq h)</td>
</tr>
<tr>
<td>Two root 3-overlap paths</td>
<td>One root homo-overlap and two root heter-overlaps, and (h = r_1 \neq r_2)</td>
</tr>
</tbody>
</table>

where \(A\): a quad-common ancestor of \(a, b, c,\) and \(d\); \(S\): a common ancestor of \(a\) and \(b\), and \(T\): a common ancestor of \(c\) and \(d\). For \(\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle\) (\(S = T = A\)), there are four types (i.e., Type 1 to Type 4).
Type 1: zero root homo-overlap and zero root heter-overlap.

Type 2: zero root homo-overlap and one root heter-overlap \( P_{Ar} \) ending at \( r \),

\[
\text{zero root homo-overlap and two root heter-overlap } P_{Ar1} \text{ and } P_{Ar2} \text{ ending at } r_1 \text{ and } r_2, \text{ respectively,}
\]

one root homo-overlap \( P_{Ah} \) ending at \( h \) and two root heter-overlap \( P_{Ar1} \) and \( P_{Ar2} \) ending at \( r_1 \) and \( r_2 \), and \( r_1 \neq r_2 \).

\[
\begin{cases}
\text{zero root homo-overlap and two root heter-overlap } \langle P_{Ar1}, P_{Ar2} \rangle \text{ ending at } r_1 \text{ and } r_2, \text{ respectively,} \\
\text{one root homo-overlap } P_{Ah} \text{ ending at } h \\
\text{and two root heter-overlap } P_{Ar1} \text{ and } P_{Ar2} \text{ ending at } r_1 \text{ and } r_2, \text{ and } r_1 \neq r_2.
\end{cases}
\]

Type 3:

Type 4: one root homo-overlap \( P_{Ah} \) ending at \( h \) and two root heter-overlap ending at \( r_1 \) and \( r_2 \), and \( h = r_1 = r_2 \). For \( \langle P_{Sa}, P_{Sb}, (P_{Tc}, P_{Td}) \rangle \) \( (S \neq T) \), there is one type (i.e., Type 5).

Type 5: \( \langle P_{Sa}, P_{Sb} \rangle \) has zero overlap individual, \( \langle P_{Tc}, P_{Td} \rangle \) has zero overlap individual.

At most one path-pair (either \( \langle P_{Sa}, P_{Sb} \rangle \) or \( \langle P_{Tc}, P_{Td} \rangle \)) can have crossover individuals.

Between a path from \( \langle P_{Sa}, P_{Sb} \rangle \) and a path from \( \langle P_{Tc}, P_{Td} \rangle \), there are no overlap individuals, but there can be crossover individuals, \( x \), where \( x \neq S \) and \( x \neq T \):

\[
B = \begin{cases}
S \text{ when } \text{gen}(S) < \text{gen}(T) \\
S \text{ when } \text{gen}(S) = \text{gen}(T) \\
T \text{ otherwise,}
\end{cases}
\]

\[
L_{2 \text{-pair}} = \begin{cases}
L_{Pa} + L_{Pa} & \text{for Type 1} \\
L_{Pa} + L_{Pa} + L_{Pa} & \text{for Type 2} \\
L_{Pa} + L_{Pa} + L_{Pa} + L_{Pa} + L_{Pa} + L_{Pa} + L_{Pa} & \text{for Type 3} \\
L_{Pa} + L_{Pa} + L_{Pa} + L_{Pa} + L_{Pa} + L_{Pa} + L_{Pa} & \text{for Type 4},
\end{cases}
\]

\[
L_{(Pa, Pb)} = L_{Pa} + L_{Pb} \text{ for Type 5,}
\]

\[
L_{(Pc, Pd)} = L_{Pc} + L_{Pd} \text{ for Type 5.}
\]

Note that if \( \langle a, b \rangle \) and \( \langle c, d \rangle \) have zero quad-common ancestors, we have the following formula for \( \Phi_{ab,cd} \):

\[
\Phi_{ab,cd} = \sum_{(S,T) \in \text{Type 6}} \left( \frac{1}{2} \right)^{L_{(Pa, Pd)} + L_{(Pc, Pd)}} \Phi_{SS} \Phi_{TT}. \tag{19}
\]

Type 6: \( \langle P_{Sa}, P_{Sb} \rangle \) is a nonoverlapping path-pair and \( \langle P_{Tc}, P_{Td} \rangle \) is a nonoverlapping path-pair. Between a path from \( \langle P_{Sa}, P_{Sb} \rangle \) and a path from \( \langle P_{Tc}, P_{Td} \rangle \), there are no overlap individuals, but there can be crossover individuals.

\[
L_{(Pa, Pd)} \text{ and } L_{(Pc, Pd)} \text{ are defined as in Type 5.}
\]

The correctness of the path-counting formula for \( \Phi_{ab,cd} \) is proven in Appendix C. For completeness, please refer to [18] for the path-counting formulas for \( \Phi_{aa,bc}, \Phi_{ab,ac}, \Phi_{ab,ab}, \Phi_{ab,cd} \).

3.4. Experimental Results. In this section, we show the efficiency of our path-counting method using NodeCodes for condensed identity coefficients by making comparisons with the performance of a recursive method used in [10]. We implemented two methods: (1) using recursive formulas to compute each required kinship coefficient and generalized kinship coefficient; (2) using path-counting method coupled with NodeCodes to compute each required kinship coefficient and generalized kinship coefficient independently. We refer to the first method as Recursive, the second method as NodeCodes. For completeness, please refer to [18] for the details of the NodeCodes-based method.

Nodecodes of a node is a set of labels each representing a path to the node from its ancestors. Given a pedigree graph, let \( r \) be the progenitor (i.e., the node with 0 in-degree). (For simplicity, we assume there is one progenitor, \( r \), as the ancestor of all individuals in the pedigree. Otherwise, a virtual node \( r \) can be added to the pedigree graph and all progenitors can be made children of \( r \).) For each node \( u \) in the graph, the set of NodeCodes of \( u \), denoted as \( NC(u) \), are assigned using a breadth-first-search traversal starting from \( r \) as follows.

1. If \( u \) is \( r \) then \( NC(r) \) contains only one element: the empty string.

2. Otherwise, let \( u \) be a node with \( NC(u) \), and \( v_0, v_1, \ldots, v_k \) be \( u \)'s children in sibling order; then for each \( x \) in \( NC(u) \), a code \( ax^r \) is added to \( NC(v_i) \), where \( 0 \leq i \leq k \), and \(^r \) indicates the gender of the individual represented by node \( v_i \).
Computations of kinship coefficients for two individuals and generalized kinship coefficients for three individuals presented in [11, 12, 14, 15] are using NodeCodes. The NodeCodes-based computation schemes can also be applied for the generalized kinship coefficients for four individuals and two pairs of individuals. For completeness, please refer to [18] for the details using NodeCodes to compute the generalized kinship coefficients for four individuals and two pairs of individuals based on our proposed path-counting formulas in Sections 3.2 and 3.3.

In order to test the scalability of our approach for calculating condensed identity coefficients on large pedigrees, we used a population simulator implemented in [11] to generate arbitrarily large pedigrees. The population simulator is based on the algorithm for generating populations with overlapping generations in Chapter 4 of [19] along with the parameters given in Appendix B of [20] to model the relatively isolated Finnish Kainuu subpopulation and its growth during the years 1500–2000. An overview of the generation algorithm was presented in [11, 12, 14]. The parameters include starting/ending year, initial population size, initial age distribution, marriage probability, maximum age at pregnancy, expected number of children by time period, immigration rate, and probability of death by time period and age group.

We examine the performance of condensed identity coefficients using twelve synthetic pedigrees which range from 75 individuals to 195,197 individuals. The smallest pedigree spans 3 generations, and the largest pedigree spans 19 generations. We analyzed the effects of pedigree size and the depth of individuals in the pedigree (the longest path between the individual and a progenitor) on the computation efficiency improvement.

In the first experiment, 300 random pairs were selected from each of our 12 synthetic pedigrees. Figure 14 shows computation efficiency improvement for each pedigree. As can be seen, the improvement of NodeCodes over Recursive grew increasingly larger as the pedigree size increased, from a comparable amount of 26.83% on the smallest pedigree to 94.75% on the largest pedigree. It also shows that path-counting method coupled with NodeCodes can scale very well on large pedigrees in terms of computing condensed identity coefficients.

In our next experiment, we examined the effect of the depth of the individual in the pedigree on the query time. For each depth, we generated 300 random pairs from the largest synthetic pedigree.

Figure 15 shows the effect of depth on the computation efficiency improvement. We can see the improvement of NodeCodes over Recursive, ranging from 86.48% to 91.30%.

4. Conclusion

We have introduced a framework for generalizing Wright’s path-counting formula for more than two individuals. Aiming at efficiently computing condensed identity coefficients, we proposed path-counting formulas (PCF) for all generalized kinship coefficients for which are sufficient for expressing condensed identity coefficients by a linear combination. We also perform experiments to compare the efficiency of our method with the recursive method for computing condensed identity coefficients on large pedigrees. Our future work includes (i) further improvements on condensed identity coefficients computation by collectively calculating the set of generalized kinship coefficients to avoid redundant computations, and (ii) experimental results for using PCF in conjunction with encoding schemes (e.g., compact path-encoding schemes [13]) for computing condensed identity coefficients on very large pedigrees.

Appendices

A. Path-Counting Formulas of Special Cases

A.1. Path-Counting Formula for $\Phi_{aabb}$. For $\langle P_{a1}, P_{a2} \rangle$, we introduce a special case, where $P_{a1}$ and $P_{a2}$ are mergeable.
Definition A.1 (Mergeable Path-Pair). A path-pair \( \langle P_{Aa1}, P_{Aa2} \rangle \) is mergeable if and only if the two paths \( P_{Aa1} \) and \( P_{Aa2} \) are completely identical.

Next, we present a graphical representation of \( \langle P_{Aa1}, P_{Aa2}, P_{Ab} \rangle \) in Figure 16.

Lemma A.2. For \( S_2 \) and \( S_3 \) in Figure 16, \( \langle P_{Aa1}, P_{Aa2} \rangle \) cannot be a mergeable path-pair.

Proof. For \( S_2 \) and \( S_3 \), if \( \langle P_{Aa1}, P_{Aa2} \rangle \) is mergeable, then any common individual \( s \) between \( P_{Aa1} \) and \( P_{Ab} \) is also a shared individual between \( P_{Ab} \) and \( P_{Aa2} \). It means \( s \in \text{Tri}_C(P_{Aa1}, P_{Aa2}, P_{Ab}) \) which contradicts the fact that \( \text{Tri}_C(P_{Aa1}, P_{Aa2}, P_{Ab}) = \emptyset \).

Considering all three scenarios in Figure 16, only \( S_1 \) can have a mergeable path-pair \( \langle P_{Aa1}, P_{Aa2} \rangle \) by Lemma A.2. Now, we present our path-counting formula for \( \Phi_{abc} \) where \( a \) is not an ancestor of \( b \) and \( c \) as follows:

\[
\Phi_{abc} = \sum_A \left( \sum_{\text{Type 1}} \left( \frac{1}{2} \right)^{L_{\text{quad}} - 1} \Phi_{AAA} + \sum_{\text{Type 2}} \left( \frac{1}{2} \right)^{L_{\text{quad}}} \Phi_{AA} + \sum_{\text{Type 3}} \left( \frac{1}{2} \right)^{L_{\text{quad}} + 1} \Phi_{AA} \right),
\]

where \( A \): a common ancestor of \( a, b, c \), and \( d \).

When \( \langle P_{Aa1}, P_{Aa2} \rangle \) is not mergeable,

Type 1: \( \langle P_{Aa1}, P_{Aa2}, P_{Ab} \rangle \) has no root 2-overlap.

Type 2: \( \langle P_{Aa1}, P_{Aa2}, P_{Ab} \rangle \) has one root 2-overlap path \( P_{As} \) ending at the individual \( s \).

When \( \langle P_{Aa1}, P_{Aa2} \rangle \) is mergeable,

Type 3: \( \langle P_{Aa1}, P_{Ab} \rangle \) is a nonoverlapping path-pair

\[
L_{\text{triple}} = \begin{cases} L_{P_{Aa1}} + L_{P_{Aa2}} + L_{P_{Ab}} & \text{for Type 1} \\ L_{P_{Aa1}} + L_{P_{Aa2}} + L_{P_{Ab}} - L_{P_{As}} & \text{for Type 2} \\ L_{P_{Aa1}} + L_{P_{Aa2}} + L_{P_{Ab}} & \text{for Type 3} \end{cases}
\]

(A.2)

For the sake of completeness, if \( a \) is an ancestor of \( b \), there is no recursive formula for \( \Phi_{abc} \) in [10], but we can use either the recursive formula for \( \Phi_{abc} \) or the path-counting formula for \( \Phi_{abc} \) to compute \( \Phi_{abc} \).

A.2. Path-Counting Formula for \( \Phi_{abc} \). Given a path-quad \( \langle P_{Aa1}, P_{Aa2}, P_{Ab}, P_{Ac} \rangle \), if \( \langle P_{Aa1}, P_{Aa2} \rangle \) is not mergeable, then we process the path-quad as equivalent to \( \langle P_{Aa1}, P_{Aa2} \rangle \), and if \( \langle P_{Aa1}, P_{Aa2} \rangle \) is mergeable, then the path-quad \( \langle P_{Aa1}, P_{Aa2}, P_{Ab}, P_{Ac} \rangle \) can be condensed to scenarios for \( \langle P_{Aa1}, P_{Ab}, P_{Ac} \rangle \). Now, we present a path-counting formula for \( \Phi_{abc} \) where \( a \) is an ancestor of \( b \) and \( c \) as follows:

\[
\Phi_{abc} = \sum_A \left( \sum_{\text{Type 1}} \left( \frac{1}{2} \right)^{L_{\text{quad}} - 1} \Phi_{AAAA} + \sum_{\text{Type 2}} \left( \frac{1}{2} \right)^{L_{\text{quad}}} \Phi_{AA} + \sum_{\text{Type 3}} \left( \frac{1}{2} \right)^{L_{\text{quad}} + 1} \Phi_{AA} \right),
\]

where \( A \): a quad-common ancestor of \( a, b, c \), and \( d \).

When \( \langle P_{Aa1}, P_{Aa2} \rangle \) is not mergeable,

Type 1: zero root 2-overlap and zero root 3-overlap path;

Type 2: one root 2-overlap path \( P_{As} \) ending at \( s \)

Case 1: two root 2-overlap paths \( P_{Aa1} \) and \( P_{Aa2} \) ending at \( s_1 \) and \( s_2 \), respectively

Case 2: one root 3-overlap path \( P_{As} \) ending at \( t \)

Case 3: one root 2-overlap and one root 3-overlap paths \( P_{As} \) and \( P_{Ac} \) ending at \( s \) and \( t \), respectively.

(A.4)

When \( \langle P_{Aa1}, P_{Aa2} \rangle \) is mergeable,

Type 4: \( \langle P_{Aa1}, P_{Ab}, P_{Ac} \rangle \) has zero root 2-overlap path;

Type 5: \( \langle P_{Aa1}, P_{Ab}, P_{Ac} \rangle \) has one root 2-overlap path \( P_{As} \) ending at \( s \)

\[
L_{\text{quad}} = \begin{cases} L_{P_{Aa1}} + L_{P_{Aa2}} + L_{P_{As}} + L_{P_{Ac}} & \text{for Type 1} \\ L_{P_{Aa1}} + L_{P_{Aa2}} + L_{P_{Ab}} + L_{P_{As}} & \text{for Type 2} \\ -L_{P_{As}} & \text{for Case 1 \( \in \) Type 3} \\ -L_{P_{Aa1}} - L_{P_{Aa2}} + L_{P_{Ab}} + L_{P_{As}} & \text{for Case 2 \( \in \) Type 3} \\ L_{P_{Aa1}} + L_{P_{Aa2}} + L_{P_{Ab}} + L_{P_{As}} & \text{for Case 3 \( \in \) Type 3} \\ -L_{P_{As}} & \text{for Case 3 \( \in \) Type 3} \end{cases}
\]

(A.5)
Note that if \( a \) is an ancestor of either \( b \) or \( c \), or both of them, then the path-counting formula of \( \Phi_{abcd} \) is applicable to compute \( \Phi_{\text{abc}} \).

A.3. Path-Counting Formula for \( \Phi_{ab} \). A special case of \( \langle P_{A_a}, P_{A_b}, P_{A_{ab}} \rangle \) for \( \langle P_{A_a}, P_{A_{ab}}, P_{A_{ab}} \rangle, P_{A_{ab}} \rangle \) is introduced when \( \langle P_{A_a}, P_{A_{ab}}, P_{A_{ab}} \rangle \) is mergeable. With the existence of a mergeable path-triple, \( \langle P_{A_a}, P_{A_{ab}}, P_{A_{ab}}, P_{A_{ab}} \rangle \) can be condensed to \( \langle P_{A_a}, P_{A_{ab}} \rangle \).

Definition A.3 (Mergeable Path-Triple). Given three paths \( P_{A_{a1}}, P_{A_{a2}}, \) and \( P_{A_{a3}} \), they are mergeable if and only if they are completely identical.

Lemma A.4. Given a path-quad \( \langle P_{A_{a1}}, P_{A_{a2}}, P_{A_{a3}}, P_{a_{ab}} \rangle \), there must be at least one mergeable path-pair among \( \langle P_{A_{a1}}, P_{A_{a2}} \rangle, \langle P_{A_{a2}}, P_{A_{a3}} \rangle, \langle P_{A_{a1}}, P_{A_{a3}} \rangle, \) and \( \langle P_{A_{a2}}, P_{A_{a3}} \rangle \).

Proof. For an individual \( a \) with two parents \( f \) and \( m \), the paternal allele of the individual \( a \) is transmitted from \( f \) and the maternal allele is transmitted from \( m \). At allele level, only two descent paths starting from an ancestor are allowed. For a path-quad \( \langle P_{A_{a1}}, P_{A_{a2}}, P_{A_{a3}}, P_{a_{ab}} \rangle \), there must be at least one mergeable path-pair among \( \langle P_{A_{a1}}, P_{A_{a2}} \rangle, \langle P_{A_{a2}}, P_{A_{a3}} \rangle, \langle P_{A_{a1}}, P_{A_{a3}} \rangle, \) and \( \langle P_{A_{a2}}, P_{A_{a3}} \rangle \).

For simplicity, we treat \( \langle P_{A_{a1}}, P_{A_{a2}} \rangle \) as a default mergeable path-pair.

Now, we present the path-counting formula for \( \Phi_{ab} \) where \( a \) is not an ancestor of \( b \) as follows:

\[
\Phi_{ab} = \sum_A \left( \frac{3}{2} \left( \sum_{\text{Type 1}} \left( \frac{1}{2} \right)^{L_{\text{triple}}-1} \Phi_{AAA} \right) + \sum_{\text{Type 2}} \left( \frac{1}{2} \right)^{L_{\text{triple}}} \Phi_{AA} \right) + \sum_{\text{Type 3}} \left( \frac{1}{2} \right)^{L_{\text{pair}}+2} \Phi_{AA},
\]

where \( A \): a common ancestor of \( a \) and \( b \).

When there is only one mergeable path-pair (let us consider \( \langle P_{A_{a1}}, P_{A_{a2}} \rangle \) as the mergeable path-pair),

Type 1: \( \langle P_{A_{a1}}, P_{A_{a2}}, P_{a_{ab}} \rangle \) has zero root 2-overlap path,
Type 2: \( \langle P_{A_{a1}}, P_{A_{a2}}, P_{a_{ab}} \rangle \) has one root 2-overlap path \( P_{A_{a}} \) ending at \( s \).

When \( \langle P_{A_{a1}}, P_{A_{a2}}, P_{a_{ab}} \rangle \) is mergeable,

Type 3: \( \langle P_{A_{a1}}, P_{a_{ab}} \rangle \) is nonoverlapping

\[
L_{\text{triple}} = \begin{cases} L_{p_{a_{ab}}} + L_{p_{a_{ab}}} + L_{p_{a_{ab}}} & \text{for Type 1} \\ L_{p_{a_{ab}}} + L_{p_{a_{ab}}} + L_{p_{a_{ab}}} - L_{p_{a_{ab}}} & \text{for Type 2} \end{cases},
\]

\[
L_{\text{pair}} = L_{p_{a_{ab}}} + L_{p_{a_{ab}}} \text{ for Type 3}.
\]

Note that if \( a \) is an ancestor of \( b \), we treat \( \Phi_{ab} = \Phi_{a_1a_2a_3b} \). Then, we apply the path-counting formula for \( \Phi_{abcd} \) to compute \( \Phi_{a_1a_2a_3b} \).

![Figure 17: Dependency graph for different cases regarding \( \Phi_{abc} \) and \( \Phi_{ab} \).](image)

B. Proof for Path-Counting Formulas of Three Individuals

We first demonstrate that, for one triple-common ancestor \( A \), the path-counting computation of \( \Phi_{abc} \) is equivalent to the computation using recursive formulas. Then, we prove the correctness of the path-counting computation for multiple triple-common ancestors.

B.1. One Triple-Common Ancestor. Considering the different types of path-triples starting from a triple-common ancestor \( A \) in a pedigree graph \( G \) contributing to \( \Phi_{abc} \) and \( \Phi_{ab} \), \( G \) can have 5 different cases:

- **Case 1:** \( G \) does not have any path-triples \( \langle P_{A_a}, P_{A_{ab}}, P_{a_{ab}} \rangle \) with root overlap
- **Case 2:** \( G \) has path-triples \( \langle P_{A_a}, P_{A_{ab}}, P_{a_{ab}} \rangle \) with root overlap
- **Case 3:** \( G \) has path-triples \( \langle P_{A_a}, P_{A_{ab}}, P_{a_{ab}} \rangle \) having mergeable path-pair \( \langle P_{A_a}, P_{A_{ab}} \rangle \)

Based on the 5 cases from Case 1 to Case 2, we first construct a dependency graph shown in Figure 17, consistent with the recursive formulas (3), (4), and (5) for the generalized kinship coefficients for three individuals.

Then, we take the following steps to prove the correctness of the path-counting formulas (12) and (A.1):

(i) for \( \Phi_{ab} \), the correctness of the path-counting formula (i.e., Wright's formula) is proven in [21]. For Case 2.1 and Case 2.2, the correctness is proven based on the correctness of Cases 3.1 and 3.2;

(ii) for Case 2.3, it has no cycle but only depends on \( \Phi_{ab} \). Thus, we prove the correctness of Case 2.3 by transforming the case to \( \Phi_{ab} \).
(iii) for Cases 3.1 and 3.2, the correctness is proven by induction on the number of edges, \( n \), in the pedigree graph \( G \).

B.1.1. Correctness Proof for Case 3.1

Case 3.1. For \( \Phi_{abc} \), \( G \) does not have any path triples \( \langle P_{Ab}, P_{Ab}, P_{Ac} \rangle \) with root overlap.

Proof (Basis). There are two basic scenarios: (i) one individual is a parent of another; (ii) no individual is a parent of another, among \( a, b, \) and \( c \).

Using the recursive formula (3) to compute \( \Phi_{abc} \), for Figure 18(a), \( \Phi_{abc} = (1/2)\Phi_{bca} = (1/2)^{2}\Phi_{ccc} \); for Figure 18(b), \( \Phi_{abc} = (1/2)\Phi_{cbe} = (1/2)^{2}\Phi_{Aac} = (1/2)^{3}\Phi_{AAA} \).

Using the path-counting formula (12), if a path-triple \( \langle P_{Ab}, P_{Ab}, P_{Ac} \rangle \) has no root overlap (i.e., Type I), then the contribution of \( \langle P_{Ab}, P_{Ab}, P_{Ac} \rangle \) to \( \Phi_{abc} \) can be computed as follows: \( \sum_{\text{Type I}} (1/2)^{1-(P_{Ab}/P_{Ab})} \Phi_{AAA} = L_{P_{Ab}P_{Ab}P_{Ac}} \).

For Figure 18(a), \( \Phi_{abc} = (1/2)^{3}\Phi_{AAA} \) and for Figure 18(b), \( \Phi_{abc} = (1/2)^{2}\Phi_{aaa} \).

Induction Step. Let \( n \) denote the number of edges in \( G \). Assume true for \( n \leq k \), where \( k \geq 2 \). Then, we show it is true for \( n = k + 1 \).

For Figures 19(a) and 19(b), among \( a, b, \) and \( c \), let \( a \) be the individual having the longest path starting from their triple-common ancestor in the pedigree graph \( G \) with \( (k + 1) \) edges. If we remove the node \( a \) and cut the edge \( f \to a \) from \( G \), then the new graph \( G' \) has \( k \) edges. In terms of computing \( \Phi_{fbc} \), \( G' \) satisfies the condition for induction hypothesis.

For Figure 19(a), \( \Phi_{fbc} = \sum_{\text{Type I}} (1/2)^{1-(P_{Ab}/P_{Ab})} \Phi_{AAA} = \sum_{\text{Type I}} (1/2) \Phi_{AAA} \).

Thus, it is true for \( n = k + 1 \).

B.1.2. Correctness Proof for Case 3.2

Case 3.2. For \( \Phi_{abc} \), \( G \) has path triples \( \langle P_{Ab}, P_{Ab}, P_{Ac} \rangle \) with root overlap.

Proof (Basis). There are three basic scenarios: (i) there are two individuals who are parents of another; (ii) there is only one individual who is parent of another; (iii) there is no individual who is a parent of another, among \( a, b, \) and \( c \).
Using the recursive formula (3) to compute \( \Phi_{abc} \): in Figure 20, for Figure 20(a), \( \Phi_{abc} = (1/2)\Phi_{abc} = (1/2)^2\Phi_{bc} = (1/2)^3\Phi_{cc} \); for Figure 20(b), \( \Phi_{abc} = (1/2)\Phi_{abc} = (1/2)^2\Phi_{bc} = (1/2)^3\Phi_{cc} \); for Figure 20(c), \( \Phi_{abc} = (1/2)^2\Phi_{ac} = (1/2)^3\Phi_{kc} = (1/2)^5\Phi_{AA}. \)

Using the path-counting formula (12), if a path-triple \( \langle P_a, P_{ab}, P_{bc} \rangle \) has root overlap (i.e., Type 2), then the contribution of \( \langle P_a, P_{ab}, P_{bc} \rangle \) to \( \Phi_{abc} \) can be computed as follows:

\[ L_{\langle P_a, P_{ab}, P_{bc} \rangle} = L_{P_a} + L_{P_{ab}} + L_{P_{bc}} \]

For Figure 20(a), \( c \) is the only triple-common ancestor and we obtain \( \Phi_{abc} = (1/2)^3\Phi_{cc} = (1/2)^{2+1}\Phi_{cc} = (1/2)^3\Phi_{cc} \). Similarly, for Figures 20(b) and 20(c), we obtain \( \Phi_{abc} = (1/2)^3\Phi_{AA} \) and \( \Phi_{abc} = (1/2)^3\Phi_{AA} \), respectively.

**Induction Step.** Let \( n \) denote the number of edges in \( G \). Assume true for \( n \leq k \), then show that it is true for \( k + 1 \).

For Figures 21(a), 21(b), and 21(c), among \( a, b, \) and \( c \), let \( a \) be the individual who has the longest path and let \( b \) be a parent of \( a \). Then, we cut the edge \( p \rightarrow a \) from \( G \) and obtain a new graph \( G' \) which satisfies the condition of induction hypothesis. For Figure 21(a), we use the path-counting formula for \( \Phi_{bc} \) in \( G' \) to obtain:

\[ \Phi_{abc} = \sum_{\text{Type 2}} (1/2)^{L_{\langle P_a, P_{ab}, P_{bc} \rangle} + 1} \Phi_{AA}. \]

In \( G \), \( f \) is the only parent of \( a \), according to the recursive formula (3), we have \( \Phi_{abc} = (1/2)\Phi_{fbc} \). Then, we can plug-in the \( \Phi_{fbc} \) and obtain:

\[ \Phi_{abc} = (1/2)^{L_{\langle P_a, P_{ab}, P_{bc} \rangle} + 1} \Phi_{AA}. \]

For Figures 21(b) and 21(c), we take the same steps as we calculate \( \Phi_{abc} \) for Figure 21(a).

**Proof.** Considering the relationship between \( a \) and \( b \), \( G \) has two scenarios: (i) \( b \) is not an ancestor of \( a \); (ii) \( b \) is an ancestor of \( a \). Using the path-counting formula (A.1), if a path-triple \( \langle P_{ab1}, P_{ab2}, P_{abk} \rangle \) is Type 3, which means it has a mergeable path-pair, then the contribution of \( \langle P_{ab1}, P_{ab2}, P_{abk} \rangle \) to \( \Phi_{abc} \) can be computed as follows:

\[ \sum_{\text{Type 3}} (1/2)^{L_{\langle P_{ab1}, P_{ab2}, P_{abk} \rangle} + 1} \Phi_{AA} \]

Using the recursive formula (4), we obtain \( \Phi_{abc} = (1/2)(\Phi_{ab} + \Phi_{fmb}) \).

**B.1.3. Correctness Proof for Case 2.3**

Case 2.3. For \( \Phi_{abc} \), the path-triples in the pedigree graph \( G \) have mergeable path-pair.

For Figure 22(a), \( A \) is a common ancestor of \( a \) and \( b \).

\[ \therefore \text{only has one parent } f \]

\[ \Phi_{abc} = \left(1/2\right)\left(\Phi_{ab} + \Phi_{fmb}\right) \]

**B.1.4. Correctness Proof for Cases 2.1 and 2.2**

For \( \Phi_{abc} \), when there is no path-triple having mergeable path-pair, (i.e., the path-triple belongs to either Case 2.1 or Case 2.3), \( \Phi_{abc} \) can be transformed to \( \Phi_{a,b} \), which is equivalent to the computation of \( \Phi_{abc} \) for Cases 3.1 and 3.2. The correctness of our path-counting formula for Cases 3.1 and 3.2 is proven. Thus, we obtain the correctness for \( \Phi_{abc} \) when the path-triple belongs to either Case 2.1 or Case 2.2.

**B.2. Multiple Triple-Common Ancestors.** Now, we provide the correctness proof for multiple triple-common ancestors regarding the path-counting formulas (12) and (A.1).
Lemma A. Given a pedigree graph $G$ and three individuals $a$, $b$, $c$ having at least one trip-common ancestor, $\Phi_{abc}$ is correctly computed using the path counting formulas (12) and (A.1).

Proof. Proof by induction on the number of triple-common ancestors.

Basis. $G$ has only one triple-common ancestor of $a$, $b$, and $c$.

The correctness of (12) and (A.1) for $G$ with only one triple-common ancestor of $a$, $b$, and $c$ is proven in the previous section.

Induction Hypothesis. Assume that if $G$ has $k$ or less triple-common ancestors of $a$, $b$, and $c$, (12) and (A.1) are correct for $G$.

Induction Step. Now, we show that it is true for $G$ with $k + 1$ triple-common ancestors of $a$, $b$, and $c$.

Let $\text{Tri}_C(a, b, c, G)$ denote all triple-common ancestors of $a$, $b$, and $c$ in $G$, where $\text{Tri}_C(a, b, c, G) = \{A_i | 1 \leq i \leq k + 1\}$. Let $A_1$ be the most top triple-common ancestor such that there is no individual among the remaining ancestors $\{A_i | 2 \leq i \leq k + 1\}$ who is an ancestor of $A_1$. Let $S(A_1)$ denote the contribution from $A_1$ to $\Phi_{abc}$.

Because $A_1$ is the most top triple-common ancestor, there is no path-triple from $\{A_i | 2 \leq i \leq k + 1\}$ to $a$, $b$, and $c$ which passes through $A_1$. Then, we can remove $A_1$ from $G$ and delete all out-going edges from $A_1$ and obtain a new graph $G'$ which has $k$ triple-common ancestors of $a$, $b$, and $c$. It means $\text{Tri}_C(a, b, c, G') = \{A_i | 2 \leq i \leq k + 1\}$.

For the new graph $G'$, we can apply our induction hypothesis and obtain $\Phi_{abc}(G')$.

For the most top triple-common ancestor $A_1$, there are two different cases considering its relationship with the other triple-common ancestors:

1. There is no individual among $\{A_i | 2 \leq i \leq k + 1\}$ who is a descendant of $A_1$;
2. There is at least one individual among $\{A_i | 2 \leq i \leq k + 1\}$ who is a descendant of $A_1$.

For (1), since no individual among $\{A_i | 2 \leq i \leq k + 1\}$ is a descendant of $A_1$, the set of path-triples from $A_1$ to $a$, $b$, and $c$ is independent of the set of path-triples from $\{A_i | 2 \leq i \leq k + 1\}$ to $a$, $b$, and $c$. It also means that the contribution from $A_1$ to $\Phi_{abc}$ is independent of the contribution from the other triple-common ancestors.

Summing up all contributions, we can obtain $\Phi_{abc}(G) = \Phi_{abc}(G') + S(A_1)$.

For (2), let $A_j$ be one descendant of $A_1$. Now both $A_1$ and $A_j$ can reach $a$, $b$, and $c$.

For (2), let $A_j$ be one descendant of $A_1$. Now both $A_1$ and $A_j$ can reach $a$, $b$, and $c$.

C. Proof for Four Individuals and Two Pairs of Individuals

Here, we give a proof sketch for the correctness of path counting formulas for four individuals. First of all, for four individuals in a pedigree graph $G$, we present all different cases based on which we construct a dependency graph. The correctness of the path-counting formulas for two-pair individuals can be proved similarly.

C.1. Proof for Four Individuals. Consider the existence of different types of path-quads regarding $\Phi_{abcd}$, $\Phi_{abc}$, and $\Phi_{aab}$; there are 15 cases for a pedigree graph $G$:

Case 2.1: $G$ has path-triples $\langle P_{\alpha\beta}, P_{\alpha\gamma}, P_{\beta\gamma}\rangle$ with zero root overlap

Case 2.2: $G$ has path-triples $\langle P_{\alpha\beta}, P_{\alpha\gamma}, P_{\beta\gamma}\rangle$ with one root overlap

Case 2.3: $G$ has path-pairs $\langle P_{\alpha\beta}, P_{\beta\gamma}\rangle$ with zero root overlap

FIGURE 22: (a) b is not an ancestor of a; (b) b is an ancestor of a.
Case 3.1: $G$ has path-quads
$\langle P_{ab}, P_{ac}, P_{bc}, P_{bc} \rangle$
with zero root overlap

Case 3.2: $G$ has path-quads
$\langle P_{ab}, P_{ac}, P_{bc}, P_{bc} \rangle$
with one root 2-overlap

Case 3.3.1: $G$ has path-quads
$\langle P_{ab}, P_{ac}, P_{bc}, P_{bc} \rangle$
with two root 2-overlap

Case 3.3.2: $G$ has path-quads
$\langle P_{ab}, P_{ac}, P_{bc}, P_{bc} \rangle$
with one root 3-overlap

Case 3.3.3: $G$ has path-quads
$\langle P_{ab}, P_{ac}, P_{bc}, P_{bc} \rangle$
with one root 2-overlap

Case 3.4: $G$ has path-triples
$\langle P_{ab}, P_{bc}, P_{bc} \rangle$
with zero root overlap

Case 3.5: $G$ has path-triples
$\langle P_{ab}, P_{bc}, P_{bc} \rangle$
with one root overlap

Case 4.1: $G$ has path-quads
$\langle P_{ab}, P_{bc}, P_{bc}, P_{bc} \rangle$
with zero root overlap

Case 4.2: $G$ has path-quads
$\langle P_{ab}, P_{bc}, P_{bc}, P_{bc} \rangle$
with one root 2-overlap

Case 4.3.1: $G$ has path-quads
$\langle P_{ab}, P_{bc}, P_{bc}, P_{bc} \rangle$
with two root 2-overlap

Case 4.3.2: $G$ has path-quads
$\langle P_{ab}, P_{bc}, P_{bc}, P_{bc} \rangle$
with one root 3-overlap

Case 4.3.3: $G$ has path-quads
$\langle P_{ab}, P_{bc}, P_{bc}, P_{bc} \rangle$
with one root 2-overlap

proven for the computation of $\Phi_{abc}$. The correctness of the transformation from Case 4.2 to Case 3.4 can be proved based on the recursive formula for $\Phi_{abc}$ and $\Phi_{abcd}$. Similarly, we can obtain the transformation from Case 4.3.1 to Case 3.5.

C.2. Proof for Two Pairs of Individuals. Consider the existence of different types of 2-pair-path-pair regarding $\Phi_{ab,cd}$; there are 9 cases which are listed as follows.

Case 4.1. $G$ has $\langle (P_{ab}, P_{bc}, P_{bc}, P_{bc}) \rangle$ with zero root homom-overlap and zero root heter-overlap.

Case 4.2. $G$ has $\langle (P_{ab}, P_{bc}, P_{bc}, P_{bc}) \rangle$ with zero root homom-overlap and one root heter-overlap.

Case 4.3.1. $G$ has $\langle (P_{ab}, P_{bc}, P_{bc}, P_{bc}) \rangle$ with zero root homom-overlap and zero root heter-overlap.

Case 4.3.2. $G$ has $\langle (P_{ab}, P_{bc}, P_{bc}, P_{bc}) \rangle$ with one root homom-overlap and two root heter-overlap.

Case 4.4. $G$ has $\langle (P_{ab}, P_{bc}, P_{bc}, P_{bc}) \rangle$ with one root homom-overlap and zero root heter-overlap.

Case 4.5. $G$ has $\langle (P_{ab}, P_{bc}, P_{bc}, P_{bc}) \rangle$ with two root homom-overlap and zero root heter-overlap.

Case 4.6. $G$ has path-triples $\langle P_{ab}, P_{bc}, P_{bc} \rangle$ with zero root overlap.

Case 4.7. $G$ has path-triples $\langle P_{ab}, P_{bc}, P_{bc} \rangle$ with one root overlap.

Case 4.8. $G$ has path-pairs $\langle P_{cd}, P_{cd} \rangle$ with zero root overlap.

Then, we construct a dependency graph for the cases relating to $\Phi_{abcd}$ in Figure 24.

According to the dependency graph in Figure 24, Cases 4.6, 4.7, and 4.8 are the intermediate steps which already are proved for the computation of $\Phi_{abcd}$. The correctness of the transformation from Case 4.2 to Case 4.6 can be proved based on the recursive formula for $\Phi_{abcd}$ and $\Phi_{abcd}$. Similarly, we can obtain the transformation from Cases 4.3.1 and 4.3.2 to Case 4.7 as well as from Case 4.4 to Case 4.8 accordingly.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.
Acknowledgments

The authors thank Professor Robert C. Elston, Case School of Medicine, for introducing to them the identity coefficients and referring them to the related literature [7, 10, 17]. This work is partially supported by the National Science Foundation Grants DBI 0743705, DBI 0849956, and CRI 0551603 and by the National Institute of Health Grant GM088823.

References
