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; echeng@uakron.edu

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

Academic Editor: Zhenyu Jia

Copyright © 2014 E. Cheng and Z. M. Ozsoyoglu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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 two pairs of 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 Φ_{ab} : the recursive approach [10] and the path-counting approach [16]. The recursive formulas [10] for Φ_{ab} and Φ_{aa} are

$$\begin{aligned}\Phi_{ab} &= \frac{1}{2} (\Phi_{fb} + \Phi_{mb}) \quad \text{if } a \text{ is not an ancestor of } b, \\ \Phi_{aa} &= \frac{1}{2} (1 + \Phi_{fm}) = \frac{1}{2} (1 + F_a),\end{aligned}\tag{1}$$

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 Φ_{ab} is

$$\Phi_{ab} = \sum_A \sum_{\langle P_{Aa}, P_{Ab} \rangle \in PP} \left(\frac{1}{2}\right)^{r+s+1} (1 + F_A),\tag{2}$$

where A is a common ancestor of a and b , PP is a set of non-overlapping path-pairs $\langle P_{Aa}, P_{Ab} \rangle$ from A to a and b , r is the length of the path P_{Aa} , s is the length of the path P_{Ab} , and F_A is the inbreeding coefficient of A . The path-pair $\langle P_{Aa}, P_{Ab} \rangle$ is *nonoverlapping* 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):

$$\begin{aligned}\Phi_{abc} &= \frac{1}{2} (\Phi_{fbc} + \Phi_{mbc}) \\ &\quad \text{if } a \text{ is not an ancestor of } b \text{ or } c, \\ \Phi_{aab} &= \frac{1}{2} (\Phi_{ab} + \Phi_{fmb}) \quad \text{if } a \text{ is not an ancestor of } b, \\ \Phi_{aaa} &= \frac{1}{4} (1 + 3\Phi_{fm}) = \frac{1}{4} (1 + 3F_a),\end{aligned}\tag{3}$$

$$\begin{aligned}\Phi_{abcd} &= \frac{1}{2} (\Phi_{fbcd} + \Phi_{mbcd}) \\ &\quad \text{if } a \text{ is not an ancestor of } b \text{ or } c \text{ or } d, \\ \Phi_{aabc} &= \frac{1}{2} (\Phi_{abc} + \Phi_{fmbc}) \\ &\quad \text{if } a \text{ is not an ancestor of } b \text{ or } c,\end{aligned}\tag{4}$$

$$\begin{aligned}\Phi_{aaab} &= \frac{1}{4} (\Phi_{ab} + 3\Phi_{fmb}) \\ &\quad \text{if } a \text{ is not an ancestor of } b, \\ \Phi_{aaaa} &= \frac{1}{8} (1 + 7\Phi_{fm}) = \frac{1}{8} (1 + 7F_a),\end{aligned}$$

$$\Phi_{ab,cd} = \frac{1}{2} (\Phi_{fb,cd} + \Phi_{mb,cd})$$

if a is not an ancestor of b or c or d ,

$$\Phi_{aa,bc} = \frac{1}{2} (\Phi_{bc} + \Phi_{fm,bc})$$

if a is not an ancestor of b or c ,

$$\Phi_{ab,ac} = \frac{1}{4} (2\Phi_{abc} + \Phi_{fb,mc} + \Phi_{mb,fc})$$

if a is not an ancestor of b or c ,

$$\Phi_{aa,ab} = \frac{1}{2} (\Phi_{ab} + \Phi_{fmb})$$

if a is not an ancestor of b ,

$$\Phi_{aa,aa} = \frac{1}{4} (1 + 3\Phi_{fm}) = \frac{1}{4} (1 + 3F_a).$$

(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_{abcd} 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. $\Phi_{ab,cd}$ 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 $\Phi_{abc} = 0$ if there is no common ancestor of a , b , and c . $\Phi_{abcd} = 0$ if there is no common ancestor of a , b , c , and d , and $\Phi_{ab,cd} = 0$ 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):

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1 & 2 & 2 & 1 & 1 & 1 \\ 4 & 0 & 2 & 0 & 2 & 0 & 2 & 1 & 0 \\ 8 & 0 & 4 & 0 & 2 & 0 & 2 & 1 & 0 \\ 8 & 0 & 2 & 0 & 4 & 0 & 2 & 1 & 0 \\ 16 & 0 & 4 & 0 & 4 & 0 & 2 & 1 & 0 \\ 4 & 4 & 2 & 2 & 2 & 2 & 1 & 1 & 1 \\ 16 & 0 & 4 & 0 & 4 & 0 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \Delta_3 \\ \Delta_4 \\ \Delta_5 \\ \Delta_6 \\ \Delta_7 \\ \Delta_8 \\ \Delta_9 \end{bmatrix} = \begin{bmatrix} 1 \\ 2\Phi_{aa} \\ 2\Phi_{bb} \\ 4\Phi_{ab} \\ 8\Phi_{aab} \\ 8\Phi_{abb} \\ 16\Phi_{aabb} \\ 4\Phi_{aa,bb} \\ 16\Phi_{ab,ab} \end{bmatrix}. \quad (6)$$

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

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 $P(A, a) \neq \text{NULL}$ if and only if A is an ancestor of a . P_{Aa} denotes a particular path from A to a , where $P_{Aa} \in P(A, a)$.

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

(5) *Nonoverlapping Path-Pair*. Given a path-pair $\langle P_{Aa}, P_{Ab} \rangle$, 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 $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$, where $P_{Aa} \in P(A, a)$, $P_{Ab} \in P(A, b)$, and $P_{Ac} \in P(A, c)$.

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

(8) $\text{Bi-}C(P_{Aa}, P_{Ab})$. It denotes all common individuals shared between P_{Aa} and P_{Ab} , except A .

(9) $\text{Tri-}C(P_{Aa}, P_{Ab}, P_{Ac})$. It denotes all common individuals shared among P_{Aa} , P_{Ab} , and P_{Ac} , except A .

(10) $\text{Quad-}C(P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad})$. It denotes all common individuals shared among P_{Aa} , P_{Ab} , P_{Ac} , and P_{Ad} , except A .

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

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

(13) *2-Overlap Path*. If s is a 2-overlap individual with respect to P_{Aa} and P_{Ab} , then both P_{Aa} 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

FIGURE 1: The 15 possible identity states for individuals a and b , grouped by their 9 condensed states. Lines indicate alleles that are IBD.

Path-pair1 $\begin{cases} A \rightarrow s \rightarrow e \rightarrow t \rightarrow a \\ A \rightarrow s \rightarrow e \rightarrow t \rightarrow b \end{cases}$	Path-pair3 $\begin{cases} A \rightarrow s \rightarrow e \rightarrow t \rightarrow a \\ A \rightarrow d \rightarrow f \rightarrow t \rightarrow b \end{cases}$
where $\{s, e, t\}$ are 2-overlap individuals.	where t is a crossover individual.
Path-pair2 $\begin{cases} A \rightarrow s \rightarrow e \rightarrow t \rightarrow a \\ A \rightarrow d \rightarrow b \end{cases}$	Path-pair4 $\begin{cases} A \rightarrow c \rightarrow t \rightarrow e \rightarrow a \\ A \rightarrow d \rightarrow f \rightarrow t \rightarrow b \end{cases}$
Non-overlapping path-pair	where t is a 2-overlap individual and e is a crossover individual.
Path-triple1 $\begin{cases} A \rightarrow s \rightarrow e \rightarrow t \rightarrow a \\ A \rightarrow d \rightarrow f \\ A \rightarrow c \end{cases}$	Path-triple4 $\begin{cases} A \rightarrow s \rightarrow e \rightarrow t \rightarrow a \\ A \rightarrow s \rightarrow f \rightarrow t \rightarrow b \\ A \rightarrow c \end{cases}$
Three independent paths	where t is a crossover individual; s is a 2-overlap individual and the overlap path is a root 2-overlap path.
Path-triple2 $\begin{cases} A \rightarrow s \rightarrow e \rightarrow t \rightarrow a \\ A \rightarrow d \rightarrow f \rightarrow t \rightarrow b \\ A \rightarrow c \end{cases}$	Path-triple5 $\begin{cases} A \rightarrow c \rightarrow e \rightarrow t \rightarrow a \\ A \rightarrow c \rightarrow e \rightarrow t \rightarrow b \\ A \rightarrow c \end{cases}$
where t is a crossover individual	where c is a 3-overlap individual; and $\{e, t\}$ are 2-overlap individuals and the overlap path is a root 2-overlap path
Path-triple3 $\begin{cases} A \rightarrow s \rightarrow e \rightarrow t \rightarrow a \\ A \rightarrow s \rightarrow e \rightarrow t \rightarrow b \\ A \rightarrow c \end{cases}$	Path-triple6 $\begin{cases} A \rightarrow c \rightarrow e \rightarrow t \rightarrow a \\ A \rightarrow c \rightarrow e \rightarrow t \rightarrow b \\ A \rightarrow c \end{cases}$
where $\{s, e, t\}$ are 2-overlap individuals and the overlap path is a root 2-overlap path.	where e is a crossover individual; t is a 2-overlap individual and the overlap path is not a root 2-overlap path; c is a 2-overlap individual and the overlap path is a root 2-overlap path

FIGURE 2: Examples of path-pairs and path-triples.

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 $A \rightarrow s \rightarrow e \rightarrow t$ 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 Φ_{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 Φ_{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.

Two individuals	Three individuals	Four individuals
Common ancestor	Triple-common ancestor	Quad-common ancestor
Path-pair	Path-triple	Path-quad
$Bi_C(P_{Aa}, P_{Ab})$	$Tri_C(P_{Aa}, P_{Ab}, P_{Ac})$	$Quad_C(P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad})$
N/A	2-Overlap individual	3-Overlap individual
N/A	2-Overlap path	3-Overlap path
N/A	Root 2-overlap path	Root 3-overlap path
N/A	Crossover individual	Crossover individual

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 Φ_{ab} is computed based on the inbreeding coefficient of A , modified by the length of each acceptable path-pair.

To compute Φ_{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 Φ_{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 Φ_{ab} , 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 Φ_{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 Φ_{abc} .

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

3. Results and Discussion

3.1. Path-Counting Formulas for Three Individuals. We first introduce a systematic method to generate all possible cases

FIGURE 4: A flowchart for path-counting formula derivation.

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 Φ_{abc} .

3.1.1. Cases for a Path-Pair. Given a path-pair $\langle P_{Aa}, P_{Ab} \rangle$ with $Bi_C(P_{Aa}, P_{Ab}) \neq NULL$, 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 $Bi_C(P_{Aa}, P_{Ab}) \neq NULL$, 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):

$$\begin{cases}
 \text{Case 1: } T \\
 \text{Case 2: } X^+ \\
 \text{Case 3: } TX^+ \\
 \text{Case 4: } T(X^+Y)^+, \\
 \text{Case 5: } T(X^+Y)^+X^+ \\
 \text{Case 6: } X^+Y \\
 \text{Case 7: } X^+(YX^+)^+ \\
 \text{Case 8: } X^+(YX^+)^+Y,
 \end{cases} \quad (7)$$

$$\left. \begin{array}{l}
 A \rightarrow s \rightarrow e \rightarrow t \rightarrow a \\
 A \rightarrow s \rightarrow e \rightarrow t \rightarrow b
 \end{array} \right\} \in T, \quad (8)$$

FIGURE 5: A path-pair level graphical representation of $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$.

where $\{s, e, t\}$ are 2-overlap individuals and the overlap path is a root 2-overlap path:

$$\left. \begin{array}{l} A \rightarrow s \rightarrow e \rightarrow t \rightarrow a \\ A \rightarrow s \rightarrow f \rightarrow t \rightarrow b \end{array} \right\} \in TX, \quad (9)$$

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

$$\left. \begin{array}{l} A \rightarrow s \rightarrow e \rightarrow t \rightarrow a \\ A \rightarrow d \rightarrow f \rightarrow t \rightarrow b \end{array} \right\} \in X, \quad (10)$$

where t is a crossover individual:

$$\left. \begin{array}{l} A \rightarrow c \rightarrow e \rightarrow t \rightarrow a \\ A \rightarrow s \rightarrow e \rightarrow t \rightarrow b \end{array} \right\} \in XY, \quad (11)$$

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 $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$, we represent each path as a node. The path-triple can be decomposed to three path-pairs (i.e., $\langle P_{Aa}, P_{Ab} \rangle$, $\langle P_{Aa}, P_{Ac} \rangle$, and $\langle P_{Ab}, P_{Ac} \rangle$). 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 $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$ consists of three independent paths. In Figure 2, *path-triple1* 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 $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$, consider the three path-pairs $\langle P_{Aa}, P_{Ab} \rangle$, $\langle P_{Aa}, P_{Ac} \rangle$, and $\langle P_{Ab}, P_{Ac} \rangle$, 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 $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$ has no contribution to Φ_{abc} .*

Proof. In [17], Nadot and Vaysseix proposed, from a genetic and biological point of view, that Φ_{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 .

FIGURE 6: Examples of pedigree and inheritance paths.

For the pedigree in Figure 6, let us consider the path-triple $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$ 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 $\langle P_{Ab}, P_{Ac} \rangle$, 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 $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$ with a 2-overlap edge between $\langle P_{Ab}, P_{Ac} \rangle$ (i.e., Case 6: XY). Therefore, such kind of path-triples has no contribution to Φ_{abc} . $\square$

Figure 6(b) shows one example of eligible inheritance paths corresponding to a pedigree graph. Each individual is represented by two allele nodes. The eligible inheritance paths in Figure 6(b) consist of red edges only.

Only Case 1, Case 2, and Case 3 do not have Y in the regular expression representation of a path-pair (see (7)); considering the scenarios S_1 – S_3 shown in Figure 5, an edge can have three options {Case 1: T ; Case 2: X ; Case 3: TX }.

3.1.3. Constructing Cases for a Path-Triple. For the scenarios S_1 – S_3 in Figure 5, we define two building blocks $\{B_1, B_2\}$ along with some rules in Figure 7 to generate acceptable cases. For B_1 , the edge can have three options {Case 1: T ; Case 2: X ; Case 3: TX }. For B_2 , we cannot allow both edges to be root overlap, because if two edges are root overlap, then

FIGURE 7: Building blocks $\{B_1, B_2\}$ and basic rules. FIGURE 8: A graphical illustration for obtaining T_3 .

P_{Aa} and P_{Ac} must share at least one common individual, except A, which contradicts the fact that P_{Aa} and P_{Ac} 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 to $S_3 = \{u_1 = B_2, u_2 = B_2, u_3 = B_2\}$, shown in Figure 8. For each u_i , we have a set of acceptable path-triples, denoted as R_i .

Considering the dependency among $\{R_1, R_2, R_3\}$, we use the natural join operator, denoted as $\bowtie$, operating on $\{R_1, R_2, R_3\}$ to generate all acceptable cases for S_3 . As a result, we obtain $T_3 = R_1 \bowtie R_2 \bowtie R_3$, where T_3 denotes the acceptable cases of the path-triple $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$ in the scenario S_3 .

For each scenario in Figure 5, we generate all acceptable cases for $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$. The scenario S_0 has no edges, and it shows that $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$ consists of three independent paths, while, for the other scenarios S_k ($k = 1, 2, 3$), the k edges can have two options:

- (1) all k edges belong to *crossover*; or
- (2) one edge belongs to *root 2-overlap*; the remaining $(k - 1)$ edges belong to *crossover*.

In summary, acceptable path-triples can have at most one root 2-overlap path, any number of crossover individuals, but zero 2-overlap path.

3.1.4. Splitting Operator. Considering the existence of root 2-overlap path and crossover in acceptable path-triples, we propose a splitting operator to transform a path-triple with crossover individuals to a noncrossover path-triple without changing the contribution from this path-triple to Φ_{abc} . The main purpose of using the splitting operator is to simplify the path-counting formula derivation process. We first use an example in Figure 9 to illustrate how the splitting operator

works. In Figure 9, there is a crossover individual s between P_{Aa} and P_{Ab} in the path triple $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$ in G_{k+1} . The splitting operator proceeds as follows:

- (1) split the node s to two nodes, s_1 and s_2 ;
- (2) transform the edges $s \rightarrow a'$ and $s \rightarrow b'$ to $s_1 \rightarrow a'$ and $s_2 \rightarrow b'$, respectively;
- (3) add two new edges, $s_2 \rightarrow a'$ and $s_1 \rightarrow b'$.

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} . $\square$

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

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

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

We call G' a *canonical graph* of G regarding Φ_{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 .

FIGURE 9: Transforming pedigree graph G_{k+1} having $k+1$ crossover to G_k having k crossover. FIGURE 10: A path-pair level graphical representation of $\langle P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad} \rangle$.

Proof (Sketch). The proof is by induction on the number of crossover individuals.

Induction hypothesis: assume that if G has k or less crossovers, there is a canonical graph G' for G .

In the induction step, let G_{k+1} be a graph with $k+1$ crossovers; let s be the lowest crossover between paths P_{Aa} and P_{Ab} in G_{k+1} . We apply the splitting operator on s in G_{k+1} and obtain G_k having k crossovers by Lemma 4. $\square$

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

$$\Phi_{abc} = \sum_A \left(\sum_{\text{Type 1}} \left(\frac{1}{2} \right)^{L_{\text{triple}}} \Phi_{AAA} + \sum_{\text{Type 2}} \left(\frac{1}{2} \right)^{L_{\text{triple}}+1} \Phi_{AA} \right), \quad (12)$$

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: $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$ has zero root 2-overlap, Type 2: $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$ has one root 2-overlap path P_{As} ending at the individual s

$$L_{\text{triple}} = \begin{cases} L_{P_{Aa}} + L_{P_{Ab}} + L_{P_{Ac}} & \text{for Type 1} \\ L_{P_{Aa}} + L_{P_{Ab}} + L_{P_{Ac}} - L_{P_{As}} & \text{for Type 2,} \end{cases} \quad (13)$$

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

For completeness, the path-counting formula for Φ_{aab} 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 $\langle P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad} \rangle$. Given a path-quad $\langle P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad} \rangle$ and $\text{Quad-C}(P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad}) = \emptyset$, the path-quad can have 11 scenarios S_0 – 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 $\text{Tri-C}(P_{Aa}, P_{Ab}, P_{Ac}) = \emptyset$, then it is equivalent to the scenario S_3 in Figure 8. Therefore, we only need to consider B_3 when $\text{Tri-C}(P_{Aa}, P_{Ab}, P_{Ac}) \neq \emptyset$.

3.2.2. Building Block-Based Cases Construction for $\langle P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad} \rangle$. 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 $S_i \in \{S_1, S_3\}$, it has only one building block, and all acceptable cases can be obtained directly. For $S_2 = \{u_1 = B_1, u_2 = B_1\}$, 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 B_3 , all three edges belong to root overlap (i.e., having root 3-overlap).

FIGURE 11: Building blocks for all scenarios of $\langle P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad} \rangle$.

TABLE 2: Largest subgraph of a scenario S_i ($4 \leq i \leq 10$ and $i \neq 6$).

S_i	S_4	S_5	S_7	S_8	S_9	S_{10}
S_j	S_3	S_3	S_6	S_5	S_7	S_9

For $S_6 = \{u_1 = B_3\}$, we obtain $T_6 = R_1$. For $S_i \in \{S_i \mid 4 \leq i \leq 10 \text{ and } i \neq 6\}$, we define the largest subgraph of S_i based on which we construct T_i .

Definition 7 (Largest Subgraph). Given a scenario S_i ($4 \leq i \leq 10$ and $i \neq 6$), the largest subgraph of S_i , denoted as S_j , is defined as follows:

- (1) S_j is a proper subgraph of S_i ;
- (2) if S_i contains B_3 , 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_i .

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

For a scenario S_i ($4 \leq i \leq 10$ and $i \neq 6$), let $\text{Diff}(S_i \setminus S_j)$ denote the set of building blocks in S_i but not in S_j , where S_j is the largest subgraph of S_i . Let $|E_i|$ and $|E_j|$ denote the number of edges in S_i and S_j , respectively. According to Table 2, we can conclude that $|E_i| - |E_j| = 1$. In order to leverage the dependency among building blocks, we consider only B_2 in $\text{Diff}(S_i \setminus S_j)$. For example, $\text{Diff}(S_5 \setminus S_3) = \{B_2\}$. Let T_3 denote all acceptable cases for S_3 . And let R_1 denote the set of acceptable cases for $\text{Diff}(S_5 \setminus S_3)$. Then, we can use S_3 and $\text{Diff}(S_5 \setminus S_3)$ to construct all acceptable cases for S_5 . Then, we apply this idea for constructing all acceptable cases for each S_i in Table 2.

Given a path-quad $\langle P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad} \rangle$, 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.

3.2.3. Path-Counting Formula for Φ_{abcd} . Now, we present the path-counting formula for Φ_{abcd} as follows:

$$\begin{aligned} \Phi_{abcd} = \sum_A \left(\sum_{\text{Type 1}} \left(\frac{1}{2} \right)^{L_{\text{quad}}} \Phi_{AAAA} \right. \\ \left. + \sum_{\text{Type 2}} \left(\frac{1}{2} \right)^{L_{\text{quad}}+1} \Phi_{AAA} \right. \\ \left. + \sum_{\text{Type 3}} \left(\frac{1}{2} \right)^{L_{\text{quad}}+2} \Phi_{AA} \right), \end{aligned} \quad (14)$$

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 : 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_{As} ending at s

$$\text{Type 3: } \begin{cases} \text{Case 1: two root 2-overlap paths } P_{As1}, \\ P_{As2} \text{ ending at } s_1 \text{ and } s_2, \text{ respectively} \\ \text{Case 2: one root 3-overlap path} \\ P_{At} \text{ ending at } t \\ \text{Case 3: one root 2-overlap path} \\ P_{As}, \text{ one root 3-overlap} \\ \text{path } P_{At} \text{ ending at } s \text{ and } t, \\ \text{respectively,} \end{cases}$$

$$L_{\text{quad}} = \begin{cases} L_{P_{Aa}} + L_{P_{Ab}} + L_{P_{Ac}} + L_{P_{Ad}} & \text{for Type 1} \\ L_{P_{Aa}} + L_{P_{Ab}} + L_{P_{Ac}} \\ + L_{P_{Ad}} - L_{P_{As}} & \text{for Type 2} \\ L_{P_{Aa}} + L_{P_{Ab}} + L_{P_{Ac}} + L_{P_{Ad}} \\ - L_{P_{As1}} - L_{P_{As2}} & \text{for Case 1} \in \text{Type 3} \\ L_{P_{Aa}} + L_{P_{Ab}} + L_{P_{Ac}} \\ + L_{P_{Ad}} - 2 * L_{P_{At}} & \text{for Case 2} \in \text{Type 3} \\ L_{P_{Aa}} + L_{P_{Ab}} + L_{P_{Ac}} + L_{P_{Ad}} \\ - L_{P_{At}} - L_{P_{As}} & \text{for Case 3} \in \text{Type 3,} \end{cases} \quad (15)$$

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

For completeness, the path-counting formulas for Φ_{aabc} and Φ_{aaab} are presented in Appendix A. The correctness of the path-counting formula for four individuals is proven in Appendix C.

FIGURE 12: Examples of 2-pair-path-quads for $\Phi_{ab,cd}$.

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 $\langle (P_{Sa}, P_{Sb}), (P_{Tc}, P_{Td}) \rangle$, where $P_{Sa} \in P(S, a)$, $P_{Sb} \in P(S, b)$, $P_{Tc} \in P(T, c)$, $P_{Td} \in P(T, d)$, S is a common ancestor of a and b , and T is a common ancestor of c and d . If $A = S = T$, 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 $\langle a, b \rangle$ and $\langle c, d \rangle$, 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 $r \in Bi_C(P_{Ai}, P_{Aj})$, where $i \in \{a, b\}$ and $j \in \{c, d\}$, we call r a *heter-overlap individual* when P_{Ai} and P_{Aj} 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 *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} .

t is a *homo-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_{Aa} 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 $\Phi_{ab,cd}$. Now, we present a path-pair level graphical 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_{Ar1} and P_{Ar2} , respectively.

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

TABLE 3: A summary of all cases for $\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle$.

$\langle P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad} \rangle$	$\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle$
Zero root 2-overlap and zero root 3-overlap	Zero root homo-overlap and zero root heter-overlap
One root 2-overlap path	One root homo-overlap and zero root heter-overlap Zero root homo-overlap and one root heter-overlap
Two root 2-overlap paths	Two root homo-overlaps and zero root heter-overlap Zero root homo-overlap and two root heter-overlaps
One root 3-overlap path	One root homo-overlap and two root heter-overlaps, and $h = r_1 = r_2$
One root 2-overlap and one root 3-overlap	One root homo-overlap and two root heter-overlaps, and $r_1 = r_2 \neq h$ One root homo-overlap and two root heter-overlaps, and $h = r_1 \neq r_2$

p_i is 0, denoted as $\text{gen}(p_i) = 0$. If an individual a has only one parent p , then we define $\text{gen}(a) = \text{gen}(p) + 1$. If an individual a has two parents f and m , we define $\text{gen}(a) = \text{MAX}\{\text{gen}(f), \text{gen}(m)\} + 1$.

The path-counting formula for $\Phi_{ab,cd}$ is as follows:

$$\begin{aligned}
 \Phi_{ab,cd} = & \sum_A \left(\sum_{\text{Type 1}} \left(\frac{1}{2} \right)^{L_{2\text{-pair}}} \Phi_{AAA} + \sum_{\text{Type 2}} \left(\frac{1}{2} \right)^{L_{2\text{-pair}}+1} \Phi_{AAA} \right. \\
 & + \sum_{\text{Type 3}} \left(\frac{1}{2} \right)^{L_{2\text{-pair}}+2} \Phi_{AA} \\
 & + \left. \sum_{\text{Type 4}} \left(\frac{1}{2} \right)^{L_{2\text{-pair}}+1} \Phi_{AA} \right) \\
 & + \sum_{(S,T) \in \text{Type 5}} \left(\frac{1}{2} \right)^{L_{(P_{Sa}, P_{Sb})} + L_{(P_{Tc}, P_{Td})} + 1} \Phi_{BB},
 \end{aligned} \tag{16}$$

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).

FIGURE 13: Scenarios of $\langle(P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad})\rangle$ at path-pair level.

Type 1: zero root homo-overlap and zero root hetero-overlap.

Type 2: zero root homo-overlap and one root hetero-overlap P_{Ar} ending at r ,

$$\text{Type 3: } \begin{cases} \text{zero root homo-overlap and two root} \\ \text{hetero-overlap } P_{Ar1} \text{ and } P_{Ar2} \text{ ending at} \\ r_1 \text{ and } r_2, \text{ respectively,} \\ \text{one root homo-overlap } P_{Ah} \text{ ending at } h \\ \text{and two root hetero-overlap } P_{Ar1} \text{ and } P_{Ar2} \\ \text{ending at } r_1 \text{ and } r_2, \text{ and } r_1 \neq r_2. \end{cases} \quad (17)$$

Type 4: one root homo-overlap P_{Ah} ending at h and two root hetero-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 $x \neq T$:

$$B = \begin{cases} S & \text{when } \text{gen}(S) < \text{gen}(T) \\ S & \text{when } \text{gen}(S) = \text{gen}(T) \\ & \text{and } T \text{ has two parents} \\ T & \text{otherwise,} \end{cases}$$

$$L_{2\text{-pair}} = \begin{cases} L_{P_{Aa}} + L_{P_{Ab}} + L_{P_{Ac}} + L_{P_{Ad}} & \text{for Type 1} \\ L_{P_{Aa}} + L_{P_{Ab}} + L_{P_{Ac}} + L_{P_{Ad}} - L_{P_{Ar}} & \text{for Type 2} \\ L_{P_{Aa}} + L_{P_{Ab}} + L_{P_{Ac}} + L_{P_{Ad}} - L_{P_{Ar1}} - L_{P_{Ar2}} & \text{for Type 3} \\ L_{P_{Aa}} + L_{P_{Ab}} + L_{P_{Ac}} + L_{P_{Ad}} - 2 * L_{P_{Ah}} & \text{for Type 4,} \end{cases} \quad (18)$$

$$L_{\langle P_{Sa}, P_{Sb} \rangle} = L_{P_{Sa}} + L_{P_{Sb}} \quad \text{for Type 5,}$$

$$L_{\langle P_{Tc}, P_{Td} \rangle} = L_{P_{Tc}} + L_{P_{Td}} \quad \text{for Type 5.}$$

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

$$\Phi_{ab,cd} = \sum_{(S,T) \in \text{Type 6}} \left(\frac{1}{2}\right)^{L_{\langle P_{Sa}, P_{Sb} \rangle} + L_{\langle P_{Tc}, P_{Td} \rangle}} \Phi_{SS} * \Phi_{TT}. \quad (19)$$

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.

$L_{\langle P_{Sa}, P_{Sb} \rangle}$ and $L_{\langle P_{Tc}, P_{Td} \rangle}$ 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 $\Phi_{aa,bc}$, $\Phi_{ab,ac}$, $\Phi_{ab,ab}$, and $\Phi_{aa,ab}$.

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 $\text{NC}(u)$, are assigned using a breadth-first-search traversal starting from r as follows.

- (1) If u is r then $\text{NC}(r)$ contains only one element: the empty string.
- (2) Otherwise, let u be a node with $\text{NC}(u)$, and $v_0, v_1, \dots, v_k$ be u 's children in sibling order; then for each x in $\text{NC}(u)$, a code xi^* is added to $\text{NC}(v_i)$, where $0 \leq i \leq k$, and $*$ 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,

FIGURE 14: The effect of pedigree size on computation efficiency improvement.

FIGURE 15: The effect of depth on computation efficiency improvement.

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 Φ_{aab} . For $\langle P_{Aa1}, P_{Aa2} \rangle$, we introduce a special case, where P_{Aa1} and P_{Aa2} are mergeable.

FIGURE 16: A path-pair level graphical representation of $\langle P_{Aa1}, P_{Aa2}, P_{Ab} \rangle$.

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_{Aa2} and P_{Ab} . 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 Φ_{aab} where a is not an ancestor of b :

$$\Phi_{aab} = \sum_A \left(\sum_{\text{Type 1}} \left(\frac{1}{2} \right)^{L_{\text{triple}}-1} \Phi_{AAA} + \sum_{\text{Type 2}} \left(\frac{1}{2} \right)^{L_{\text{triple}}} \Phi_{AA} + \sum_{\text{Type 3}} \left(\frac{1}{2} \right)^{L_{(P_{Aa1}, P_{Ab})}+1} \Phi_{AA} \right), \quad (\text{A.1})$$

where A : a common ancestor of a and b .

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: $\langle P_{Aa1}, P_{Aa2} \rangle$ is a nonoverlapping path-pair

$$L_{\text{triple}} = \begin{cases} L_{P_{Aa1}} + L_{P_{Aa2}} + L_{P_{Ab}} & \text{for Type 1} \\ L_{P_{Aa1}} + L_{P_{Aa2}} + L_{P_{Ab}} - L_{P_{As}} & \text{for Type 2,} \end{cases} \quad (\text{A.2})$$

$$L_{\langle P_{Aa1}, P_{Ab} \rangle} = L_{P_{Aa1}} + L_{P_{Ab}} \quad \text{for Type 3.}$$

For the sake of completeness, if a is an ancestor of b , there is no recursive formula for Φ_{aab} in [10], but we can use either the recursive formula for Φ_{abc} or the path-counting formula for Φ_{abc} to compute Φ_{a1a2b} . $\square$

A.2. Path-Counting Formula for Φ_{aabc} . 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 $\langle P_{Aa1}, P_{Ab}, P_{Ac} \rangle$.

If $\langle P_{Aa1}, P_{Aa2} \rangle$ is mergeable, 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 Φ_{aabc} where a is not an ancestor of b and c as follows:

$$\Phi_{aabc} = \sum_A \left(\sum_{\text{Type 1}} \left(\frac{1}{2} \right)^{L_{\text{quad}}-1} \Phi_{AAAA} + \sum_{\text{Type 2}} \left(\frac{1}{2} \right)^{L_{\text{quad}}} \Phi_{AAA} + \sum_{\text{Type 3}} \left(\frac{1}{2} \right)^{L_{\text{quad}}+1} \Phi_{AA} + \sum_A \left(\sum_{\text{Type 4}} \left(\frac{1}{2} \right)^{L_{\text{triple}}+1} \Phi_{AAA} + \sum_{\text{Type 5}} \left(\frac{1}{2} \right)^{L_{\text{triple}}+2} \Phi_{AA} \right), \quad (\text{A.3})$$

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

$$\text{Type 3: } \begin{cases} \text{Case 1: two root 2-overlap paths } P_{As1} \\ \text{and } P_{As2} \text{ ending at } s_1 \text{ and } s_2, \text{ respectively} \\ \text{Case 2: one root 3-overlap path } P_{At} \\ \text{ending at } t \\ \text{Case 3: one root 2-overlap} \\ \text{and one root 3-overlap paths} \\ P_{As} \text{ and } P_{At} \text{ ending at } s \text{ and } t, \\ \text{respectively.} \end{cases} \quad (\text{A.4})$$

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

$$L_{\text{quad}} = \begin{cases} L_{P_{Aa1}} + L_{P_{Aa2}} + L_{P_{Ab}} + L_{P_{Ac}} & \text{for Type 1} \\ L_{P_{Aa1}} + L_{P_{Aa2}} + L_{P_{Ab}} + L_{P_{Ac}} - L_{P_{As}} & \text{for Type 2} \\ L_{P_{Aa1}} + L_{P_{Aa2}} + L_{P_{Ab}} + L_{P_{Ac}} - L_{P_{As1}} - L_{P_{As2}} & \text{for Case 1} \in \text{Type 3} \\ L_{P_{Aa1}} + L_{P_{Aa2}} + L_{P_{Ab}} + L_{P_{Ac}} - L_{P_{At}} & \text{for Case 2} \in \text{Type 3} \\ L_{P_{Aa1}} + L_{P_{Aa2}} + L_{P_{Ab}} + L_{P_{Ac}} - L_{P_{At}} - L_{P_{As}} & \text{for Case 3} \in \text{Type 3,} \end{cases}$$

$$L_{\text{triple}} = \begin{cases} L_{P_{Aa1}} + L_{P_{Ab}} + L_{P_{Ac}} & \text{for Type 4} \\ L_{P_{Aa1}} + L_{P_{Ab}} + L_{P_{Ac}} - L_{P_{As}} & \text{for Type 5.} \end{cases} \quad (\text{A.5})$$

Note that if a is an ancestor of either b or c , or both of them, then the path-counting formula of Φ_{abcd} is applicable to compute Φ_{a1a2bc} .

A.3. Path-Counting Formula for Φ_{aaab} . A special case of $\langle P_{Aa1}, P_{Aa2}, P_{Aa3} \rangle$ for $\langle P_{Aa1}, P_{Aa2}, P_{Aa3}, P_{Ab} \rangle$ is introduced when $\langle P_{Aa1}, P_{Aa2}, P_{Aa3} \rangle$ is mergeable. With the existence of a mergeable path-triple, $\langle P_{Aa1}, P_{Aa2}, P_{Aa3}, P_{Ab} \rangle$ can be condensed to $\langle P_{Aa}, P_{Ab} \rangle$.

Definition A.3 (Mergeable Path-Triple). Given three paths P_{Aa1} , P_{Aa2} , and P_{Aa3} , they are *mergeable* if and only if they are completely identical.

Lemma A.4. Given a path-quad $\langle P_{Aa1}, P_{Aa2}, P_{Aa3}, P_{Ab} \rangle$, there must be at least one mergeable path-pair among $\langle P_{Aa1}, P_{Aa2} \rangle$, $\langle P_{Aa1}, P_{Aa3} \rangle$, $\langle P_{Aa2}, P_{Aa3} \rangle$.

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 $\langle P_{Aa1}, P_{Aa2}, P_{Aa3}, P_{Ab} \rangle$, there must be at least one mergeable path-pair among $\langle P_{Aa1}, P_{Aa2} \rangle$, $\langle P_{Aa1}, P_{Aa3} \rangle$, and $\langle P_{Aa2}, P_{Aa3} \rangle$. $\square$

For simplicity, we treat $\langle P_{Aa1}, P_{Aa2} \rangle$ as a default mergeable path-pair.

Now, we present the path-counting formula for Φ_{aaab} where a is not an ancestor of b as follows:

$$\begin{aligned} \Phi_{aaab} = \sum_A \left(\frac{3}{2} \left(\sum_{\text{Type 1}} \left(\frac{1}{2} \right)^{L_{\text{triple}}-1} \Phi_{AAA} \right. \right. \\ \left. \left. + \sum_{\text{Type 2}} \left(\frac{1}{2} \right)^{L_{\text{triple}}} \Phi_{AA} \right) \right. \\ \left. + \sum_{\text{Type 3}} \left(\frac{1}{2} \right)^{L_{\text{pair}}+2} \Phi_{AA} \right), \end{aligned} \quad (\text{A.6})$$

where A : a common ancestor of a and b .

When there is only one mergeable path-pair (let us consider $\langle P_{Aa1}, P_{Aa2} \rangle$ as the mergeable path-pair),

Type 1: $\langle P_{Aa1}, P_{Aa3}, P_{Ab} \rangle$ has zero root 2-overlap path,

Type 2: $\langle P_{Aa1}, P_{Aa3}, P_{Ab} \rangle$ has one root 2-overlap path P_{As} ending at s .

When $\langle P_{Aa1}, P_{Aa2}, P_{Aa3} \rangle$ is mergeable,

Type 3: $\langle P_{Aa}, P_{Ab} \rangle$ is nonoverlapping

$$L_{\text{triple}} = \begin{cases} L_{P_{Aa1}} + L_{P_{Aa3}} + L_{P_{Ab}} & \text{for Type 1} \\ L_{P_{Aa1}} + L_{P_{Aa3}} + L_{P_{Ab}} - L_{P_{As}} & \text{for Type 2,} \end{cases} \quad (\text{A.7})$$

$$L_{\text{pair}} = L_{P_{Aa}} + L_{P_{Ab}} \quad \text{for Type 3.}$$

Note that if a is an ancestor of b , we treat $\Phi_{aaab} = \Phi_{a1a2a3b}$. Then, we apply the path-counting formula for Φ_{abcd} to compute $\Phi_{a1a2a3b}$.

FIGURE 17: Dependency graph for different cases regarding Φ_{abc} and Φ_{aab} .

B. Proof for Path-Counting Formulas of Three Individuals

We first demonstrate that, for one triple-common ancestor A , the path-counting computation of Φ_{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 Φ_{abc} and Φ_{aab} , G can have 5 different cases:

$$\begin{aligned} \left. \begin{array}{l} \text{Case 2.1: } G \text{ does not have} \\ \text{any path-triples} \\ \langle P_{Aa1}, P_{Aa2}, P_{Ab} \rangle \\ \text{with root overlap} \\ \text{Case 2.2: } G \text{ has path-triples} \\ \langle P_{Aa1}, P_{Aa2}, P_{Ab} \rangle \\ \text{with root overlap} \\ \text{Case 2.3: } G \text{ has path-triples} \\ \langle P_{Aa1}, P_{Aa2}, P_{Ab} \rangle \\ \text{having mergeable} \\ \text{path-pair } \langle P_{Aa1}, P_{Aa2} \rangle \end{array} \right\} \Leftarrow \Phi_{aab}, \\ \\ \left. \begin{array}{l} \text{Case 3.1: } G \text{ does not have} \\ \text{any path-triples} \\ \langle P_{Aa}, P_{Ab}, P_{Ac} \rangle \\ \text{with root overlap} \\ \text{Case 3.2: } G \text{ has path-triples} \\ \langle P_{Aa}, P_{Ab}, P_{Ac} \rangle \\ \text{with root overlap} \end{array} \right\} \Leftarrow \Phi_{abc}. \end{aligned} \quad (\text{B.1})$$

Based on the 5 cases from Case 2.1 to Case 3.2, 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 Φ_{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 Φ_{ab} . Thus, we prove the correctness of Case 2.3 by transforming the case to Φ_{ab} ;

FIGURE 18: (a) c is a parent of a and b ; (b) no individual is a parent of another. FIGURE 19: (a) No individual is a parent of another; (b) c is an ancestor of a and b .

(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 Φ_{abc} , G does not have any path triples $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$ 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 Φ_{abc} , for Figure 18(a), $\Phi_{abc} = (1/2)\Phi_{cbc} = (1/2)^2\Phi_{ccc}$; for Figure 18(b), $\Phi_{abc} = (1/2)\Phi_{Abc} = (1/2)^2\Phi_{AAc} = (1/2)^3\Phi_{AAA}$.

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

For Figure 18(a), c is the only triple-common ancestor and we obtain $\Phi_{abc} = (1/2)^{L_{\langle P_{ca}, P_{cb}, P_{cc} \rangle}} \Phi_{ccc} = (1/2)^2\Phi_{ccc}$; for Figure 18(b), we obtain $\Phi_{abc} = (1/2)^{L_{\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle}} \Phi_{AAA} = (1/2)^3\Phi_{AAA}$.

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 $f \rightarrow a$ from G ,

then the new graph G^* has k edges. In terms of computing Φ_{fbc} , G^* satisfies the condition for induction hypothesis.

For Figure 19(a), $\Phi_{fbc} = \sum_{\text{Type 1}} (1/2)^{L_{\langle P_{Af}, P_{Ab}, P_{Ac} \rangle}} \Phi_{AAA}$. Based on the recursive formula (3), $\Phi_{abc} = (1/2)(\Phi_{fbc} + \Phi_{mbc})$ where f and m are parents of a . In G , a only has one parent f ; thus, it indicates $\Phi_{mbc} = 0$. Then, we can plug-in the path-counting formula for Φ_{fbc} to obtain

$$\begin{aligned} \Phi_{abc} &= \frac{1}{2} \Phi_{fbc} \\ &= \frac{1}{2} * \sum_{\text{Type 1}} \left(\frac{1}{2} \right)^{L_{\langle P_{Af}, P_{Ab}, P_{Ac} \rangle}} \Phi_{AAA} \\ &= \sum_{\text{Type 1}} \left(\frac{1}{2} \right)^{L_{\langle P_{Af}, P_{Ab}, P_{Ac} \rangle} + 1} \Phi_{AAA} \quad (\text{B.2}) \\ &\because L_{\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle} = L_{\langle P_{Af}, P_{Ab}, P_{Ac} \rangle} + 1 \\ &\therefore \Phi_{abc} = \sum_{\text{Type 1}} \left(\frac{1}{2} \right)^{L_{\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle}} \Phi_{AAA}. \end{aligned}$$

Similarly, for Figure 19(b), we obtain $\Phi_{abc} = \sum_{\text{Type 1}} (1/2)^{L_{\langle P_{cf}, P_{cb}, P_{cc} \rangle} + 1} \Phi_{ccc} = \sum_{\text{Type 1}} (1/2)^{L_{\langle P_{ca}, P_{cb}, P_{cc} \rangle}} \Phi_{ccc}$. Thus, it is true for $n = k + 1$. $\square$

B.1.2. Correctness Proof for Case 3.2

Case 3.2. For Φ_{abc} , G has path triples $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$ 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 .

FIGURE 20: (a) b is a parent of a , and c is a parent of b ; (b) b is a parent of a ; (c) no individual who is a parent of another.

Using the recursive formula (3) to compute Φ_{abc} : in Figure 20, for Figure 20(a), $\Phi_{abc} = (1/2)\Phi_{bbc} = (1/2)^2\Phi_{bc} = (1/2)^3\Phi_{cc}$; for Figure 20(b), $\Phi_{abc} = (1/2)\Phi_{bbc} = (1/2)^2\Phi_{bc} = (1/2)^4\Phi_{AA}$; for Figure 20(c), $\Phi_{abc} = (1/2)^2\Phi_{ssc} = (1/2)^3\Phi_{sc} = (1/2)^5\Phi_{AA}$.

Using the path-counting formula (12), if a path-triple $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$ has root overlap (i.e., Type 2), then the contribution of $\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle$ to Φ_{abc} can be computed as follows: $\sum_{\text{Type 2}} (1/2)^{L_{\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle} + 1} \Phi_{AA}$, where $L_{\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle} = L_{P_{Aa}} + L_{P_{Ab}} + L_{P_{Ac}} - L_{P_{As}}$ and s is the last individual of the root overlap path P_{As} .

For Figure 20(a), c is the only triple-common ancestor and we obtain $\Phi_{abc} = (1/2)^{L_{\langle P_{ca}, P_{cb}, P_{cc} \rangle} + 1} \Phi_{cc} = (1/2)^{2+1} \Phi_{cc} = (1/2)^3 \Phi_{cc}$. Similarly, for Figures 20(b) and 20(c), we obtain $\Phi_{abc} = (1/2)^4 \Phi_{AA}$ and $\Phi_{abc} = (1/2)^5 \Phi_{AA}$, respectively.

Induction Step. Let n denote the number of edges in G . Assume true for $n \leq k$, where $k \geq 2$. Show that it is true for $n = k + 1$.

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 p 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 Φ_{fbc} in G^* : $\Phi_{fbc} = \sum_{\text{Type 2}} (1/2)^{L_{\langle P_{Af}, P_{Ab}, P_{Ac} \rangle} + 1} \Phi_{AA}$.

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 Φ_{fbc} and obtain

$$\begin{aligned}
 \Phi_{abc} &= \frac{1}{2} \Phi_{fbc} \\
 &= \frac{1}{2} \sum_{\text{Type 2}} \left(\frac{1}{2} \right)^{L_{\langle P_{Af}, P_{Ab}, P_{Ac} \rangle} + 1} \Phi_{AA} \\
 &= \sum_{\text{Type 2}} \left(\frac{1}{2} \right)^{L_{\langle P_{Af}, P_{Ab}, P_{Ac} \rangle} + 1 + 1} \Phi_{AA} \\
 &\because L_{\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle} = L_{\langle P_{Af}, P_{Ab}, P_{Ac} \rangle} + 1 \\
 \therefore \Phi_{abc} &= \sum_{\text{Type 2}} \left(\frac{1}{2} \right)^{L_{\langle P_{Af}, P_{Ab}, P_{Ac} \rangle} + 1 + 1} \Phi_{AA} \\
 &= \sum_{\text{Type 2}} \left(\frac{1}{2} \right)^{L_{\langle P_{Aa}, P_{Ab}, P_{Ac} \rangle} + 1} \Phi_{AA}.
 \end{aligned} \tag{B.3}$$

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

In summary, it is true for $n = k + 1$. $\square$

FIGURE 21: (a) No individual who is a parent of another; (b) b is a parent of a ; (c) b is a parent of a and c is an ancestor of b .

B.1.3. Correctness Proof for Case 2.3

Case 2.3. For Φ_{aab} , the path-triples in the pedigree graph G have mergeable path-pair.

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 $\langle P_{Aa1}, P_{Aa2}, P_{Ab} \rangle \in \text{Type 3}$, which means that it has a mergeable path-pair, then the contribution of $\langle P_{Aa1}, P_{Aa2}, P_{Ab} \rangle$ to Φ_{aab} can be computed as follows: $\sum_{\text{Type 3}} (1/2)^{L_{\langle P_{Aa}, P_{Ab} \rangle} + 1} \Phi_{AA}$, where $L_{\langle P_{Aa}, P_{Ab} \rangle} = L_{P_{Aa}} + L_{P_{Ab}}$.

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

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

$\therefore a$ only has one parent f

$$\begin{aligned}
 \therefore \Phi_{aab} &= \frac{1}{2} (\Phi_{ab} + \Phi_{fmb}) \\
 &= \frac{1}{2} (\Phi_{ab} + 0) = \frac{1}{2} \Phi_{ab} \quad (\text{as } m \text{ is missing}).
 \end{aligned} \tag{B.4}$$

For Φ_{ab} , we use Wright's formula and obtain $\Phi_{ab} = \sum_P (1/2)^{L_{\langle P_{Aa}, P_{Ab} \rangle}} \Phi_{AA}$ where P denotes all nonoverlapping path-pairs $\langle P_{Aa}, P_{Ab} \rangle$.

Then, we have $\Phi_{aab} = (1/2)\Phi_{ab} = (1/2) \sum_P (1/2)^{L_{\langle P_{Aa}, P_{Ab} \rangle}} \Phi_{AA} = \sum_P (1/2)^{L_{\langle P_{Aa}, P_{Ab} \rangle} + 1} \Phi_{AA}$.

For Figure 22(b), we can also transform the computation of Φ_{aab} to Φ_{ab} .

In summary, it shows that the path-counting formula (A.1) is true for Case 2.3. $\square$

B.1.4. Correctness Proof for Cases 2.1 and 2.2. For Φ_{aab} , 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), Φ_{aab} can be transformed to $\Phi_{a_1 a_2 b}$, which is equivalent to the computation of Φ_{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 Φ_{aab} 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).

FIGURE 22: (a) b is not an ancestor of a ; (b) b is an ancestor of a .

Lemma A. Given a pedigree graph G and three individuals a, b, c having at least one trip-common ancestor, Φ_{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 $\text{Tri_C}(a, b, c, G) = \{A_i \mid 1 \leq i \leq k + 1\}$. Let A_1 be the most top triple-common ancestor such that there is no individual among the remaining ancestors $\{A_i \mid 2 \leq i \leq k + 1\}$ who is an ancestor of A_1 . Let $S(A_1)$ denote the contribution from A_1 to Φ_{abc} .

Because A_1 is the most top triple-common ancestor, there is no path-triple from $\{A_i \mid 2 \leq i \leq k + 1\}$ 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 $\text{Tri_C}(a, b, c, G') = \{A_i \mid 2 \leq i \leq k + 1\}$.

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 $\{A_i \mid 2 \leq i \leq k + 1\}$ who is a descendant of A_1 ;
- (2) there is at least one individual among $\{A_i \mid 2 \leq i \leq k + 1\}$ who is a descendant of A_1 .

For (1), since no individual among $\{A_i \mid 2 \leq i \leq k + 1\}$ 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 $\{A_i \mid 2 \leq i \leq k + 1\}$ to a, b , and c . It also means that the contribution from

A_1 to Φ_{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 .

$pt_i = \{t_a: A_1 \rightarrow \cdots \rightarrow a; t_b: A_1 \rightarrow \cdots \rightarrow b; t_c: A_1 \rightarrow \cdots \rightarrow c\}$, a path-triple from A_1 to a, b , and c .

If t_a, t_b , and t_c all pass through A_j , then the path-triple pt_i is not an eligible path-triple for Φ_{abc} . When we compute the contribution from A_1 to Φ_{abc} , we exclude all such path-triples where t_a, t_b , and t_c all pass through a lower triple-common ancestor. In other words, an eligible path-triple from A_1 regarding Φ_{abc} cannot have three paths all passing through a lower triple-common ancestor. Therefore, we know that that the contribution from A_1 to Φ_{abc} is independent of the contribution from the other triple-common ancestors. Summing up all contributions, we obtain $\Phi_{abc}(G) = \Phi_{abc}(G') + S(A_1)$. $\square$

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 Φ_{abcd} , Φ_{aabc} , and Φ_{aaab} ; there are 15 cases for a pedigree graph G :

$$\left. \begin{array}{l}
 \text{Case 2.1: } G \text{ has path-triples} \\
 \quad \langle P_{Aa_1}, P_{Aa_2}, P_{Ab} \rangle \\
 \quad \text{with zero root overlap} \\
 \text{Case 2.2: } G \text{ has path-triples} \\
 \quad \langle P_{Aa_1}, P_{Aa_2}, P_{Ab} \rangle \\
 \quad \text{with one root overlap} \\
 \text{Case 2.3: } G \text{ has path-pairs} \\
 \quad \langle P_{Aa}, P_{Ab} \rangle \\
 \quad \text{with zero root overlap}
 \end{array} \right\} \Longleftarrow \Phi_{aaab}$$

FIGURE 23: Dependency graph for different cases for four individuals.

$$\begin{array}{l}
 \left. \begin{array}{l}
 \text{Case 3.1: } G \text{ has path-quads} \\
 \langle P_{Aa_1}, P_{Aa_2}, P_{Ab}, P_{Ac} \rangle \\
 \text{with zero root overlap} \\
 \text{Case 3.2: } G \text{ has path-quads} \\
 \langle P_{Aa_1}, P_{Aa_2}, P_{Ab}, P_{Ac} \rangle \\
 \text{with one root 2-overlap} \\
 \text{Case 3.3.1: } G \text{ has path-quads} \\
 \langle P_{Aa_1}, P_{Aa_2}, P_{Ab}, P_{Ac} \rangle \\
 \text{with two root 2-overlap} \\
 \text{Case 3.3.2: } G \text{ has path-quads} \\
 \langle P_{Aa_1}, P_{Aa_2}, P_{Ab}, P_{Ac} \rangle \\
 \text{with one root 3-overlap} \\
 \text{Case 3.3.3: } G \text{ has path-quads} \\
 \langle P_{Aa_1}, P_{Aa_2}, P_{Ab}, P_{Ac} \rangle \\
 \text{with one root 2-overlap} \\
 \text{and one root 3-overlap} \\
 \text{Case 3.4: } G \text{ has path-triples} \\
 \langle P_{Aa}, P_{Ab}, P_{Ac} \rangle \\
 \text{with zero root overlap} \\
 \text{Case 3.5: } G \text{ has path-triples} \\
 \langle P_{Aa}, P_{Ab}, P_{Ac} \rangle \\
 \text{with one root overlap}
 \end{array} \right\} \Leftarrow \Phi_{abc}, \\
 \\
 \left. \begin{array}{l}
 \text{Case 4.1: } G \text{ has path-quads} \\
 \langle P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad} \rangle \\
 \text{with zero root overlap} \\
 \text{Case 4.2: } G \text{ has path-quads} \\
 \langle P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad} \rangle \\
 \text{with one root 2-overlap} \\
 \text{Case 4.3.1: } G \text{ has path-quads} \\
 \langle P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad} \rangle \\
 \text{with two root 2-overlap} \\
 \text{Case 4.3.2: } G \text{ has path-quads} \\
 \langle P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad} \rangle \\
 \text{with one root 3-overlap} \\
 \text{Case 4.3.3: } G \text{ has path-quads} \\
 \langle P_{Aa}, P_{Ab}, P_{Ac}, P_{Ad} \rangle \\
 \text{with one root 2-overlap} \\
 \text{and one root 3-overlap}
 \end{array} \right\} \Leftarrow \Phi_{abcd}.
 \end{array} \tag{C.1}$$

Then, we construct a dependency graph shown in Figure 23 for all cases for four individuals.

According to the dependency graph in Figure 23, the intermediate steps including Cases 3.4 and 3.5 are already

proved for the computation of Φ_{abc} . The correctness of the transformation from Case 4.2 to Case 3.4 can be proved based on the recursive formula for Φ_{abcd} and Φ_{abc} . 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 $\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle$ with zero root homo-overlap and zero root heter-overlap.

Case 4.2. G has $\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle$ with zero root homo-overlap and one root heter-overlap.

Case 4.3.1. G has $\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle$ with zero root homo-overlap and two root heter-overlap.

Case 4.3.2. G has $\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle$ with one root homo-overlap and two root heter-overlap.

Case 4.4. G has $\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle$ with one root homo-overlap and zero root heter-overlap.

Case 4.5. G has $\langle (P_{Aa}, P_{Ab}), (P_{Ac}, P_{Ad}) \rangle$ with two root homo-overlap and zero root heter-overlap.

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

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

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

Then, we construct a dependency graph for the cases relating to $\Phi_{ab,cd}$ 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 Φ_{abc} . The correctness of the transformation from Case 4.2 to Case 4.6 can be proved based on the recursive formula for $\Phi_{ab,cd}$ and $\Phi_{ab,ac}$. 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.

FIGURE 24: Dependency graph for different cases for two pairs of individuals.

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

- [1] Surgeon General's New Family Health History Tool Is Released, Ready for "21st Century Medicine", <http://compmed.com/category/people-helping-people/page/7/>.
- [2] M. Falchi, P. Forabosco, E. Mocci et al., "A genomewide search using an original pairwise sampling approach for large genealogies identifies a new locus for total and low-density lipoprotein cholesterol in two genetically differentiated isolates of Sardinia," *The American Journal of Human Genetics*, vol. 75, no. 6, pp. 1015–1031, 2004.
- [3] M. Ciullo, C. Bellenguez, V. Colonna et al., "New susceptibility locus for hypertension on chromosome 8q by efficient pedigree-breaking in an Italian isolate," *Human Molecular Genetics*, vol. 15, no. 10, pp. 1735–1743, 2006.
- [4] Glossary of Genetic Terms, National Human Genome Research Institute, <http://www.genome.gov/glossary/?id=148>.
- [5] C. W. Cotterman, *A calculus for statistico-genetics* [Ph.D. thesis], Columbus, Ohio, USA, Ohio State University, 1940, Reprinted in P. Ballonoff, Ed., *Genetics and Social Structure*, Dowden, Hutchinson & Ross, Stroudsburg, Pa, USA, 1974.
- [6] G. Malecot, *Les mathématique de l'hérédité*, Masson, Paris, France, 1948, Translated edition: *The Mathematics of Heredity*, Freeman, San Francisco, Calif, USA, 1969.
- [7] M. Gillois, "La relation d'identité en génétique," *Annales de l'Institut Henri Poincaré B*, vol. 2, pp. 1–94, 1964.
- [8] D. L. Harris, "Genotypic covariances between inbred relatives," *Genetics*, vol. 50, pp. 1319–1348, 1964.
- [9] A. Jacquard, "Logique du calcul des coefficients d'identité entre deux individus," *Population*, vol. 21, pp. 751–776, 1966.
- [10] G. Karigl, "A recursive algorithm for the calculation of identity coefficients," *Annals of Human Genetics*, vol. 45, no. 3, pp. 299–305, 1981.
- [11] B. Elliott, S. F. Akgul, S. Mayes, and Z. M. Ozsoyoglu, "Efficient evaluation of inbreeding queries on pedigree data," in *Proceedings of the 19th International Conference on Scientific and Statistical Database Management (SSDBM '07)*, July 2007.
- [12] B. Elliott, E. Cheng, S. Mayes, and Z. M. Ozsoyoglu, "Efficiently calculating inbreeding on large pedigrees databases," *Information Systems*, vol. 34, no. 6, pp. 469–492, 2009.
- [13] L. Yang, E. Cheng, and Z. M. Özsoyoğlu, "Using compact encodings for path-based computations on pedigree graphs," in *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine (ACM-BCB '11)*, pp. 235–244, August 2011.
- [14] E. Cheng, B. Elliott, and Z. M. Ozsoyoglu, "Scalable computation of kinship and identity coefficients on large pedigrees," in *Proceedings of the 7th Annual International Conference on Computational Systems Bioinformatics (CSB '08)*, pp. 27–36, 2008.
- [15] E. Cheng, B. Elliott, and Z. M. Özsoyoğlu, "Efficient computation of kinship and identity coefficients on large pedigrees," *Journal of Bioinformatics and Computational Biology (JBCB)*, vol. 7, no. 3, pp. 429–453, 2009.
- [16] S. Wright, "Coefficients of inbreeding and relationship," *The American Naturalist*, vol. 56, no. 645, 1922.
- [17] R. Nadot and G. Vaysseix, "Kinship and identity algorithm of coefficients of identity," *Biometrics*, vol. 29, no. 2, pp. 347–359, 1973.
- [18] E. Cheng, *Scalable path-based computations on pedigree data* [Ph.D. thesis], Case Western Reserve University, Cleveland, Ohio, USA, 2012.
- [19] V. Ollikainen, *Simulation Techniques for Disease Gene Localization in Isolated Populations* [Ph.D. thesis], University of Helsinki, Helsinki, Finland, 2002.
- [20] H. T. T. Toivonen, P. Onkamo, K. Vasko et al., "Data mining applied to linkage disequilibrium mapping," *The American Journal of Human Genetics*, vol. 67, no. 1, pp. 133–145, 2000.
- [21] W. Boucher, "Calculation of the inbreeding coefficient," *Journal of Mathematical Biology*, vol. 26, no. 1, pp. 57–64, 1988.