AncestryDNA Matching White Paper

Last updated January 10, 2025

Discovering genetic matches across a massive, expanding genetic database

Catherine A. Ball, Mathew J Barber, Jake Byrnes, Peter Carbonetto, Kenneth G. Chahine, Ross E. Curtis, Julie M. Granka, Eunjung Han, Eurie L. Hong, Amir R. Kermany, Natalie M. Myres, Keith Noto, Jianlong Qi, Kristin Rand, D. Barry Starr, Yong Wang, Lindsay Willmore, Aaron B. Wolf *(in alphabetical order)*

1. Introduction

AncestryDNA® conducts several genetic analyses to help customers find, preserve, and share their family history. Here we explain how we detect "matches" from DNA—more precisely, how we identify long chromosome segments shared by pairs of individuals that are suggestive of recent common ancestry. In the field of genetics, this is called "identity-by-descent" (IBD).

Once we identify IBD segments, we use this information to estimate how people are related to one another (e.g., first cousins). By drawing connections between relatives through their DNA, we offer the opportunity for AncestryDNA members to expand their documented pedigrees. Additionally, matching is an important building block for other AncestryDNA features such as ThruLines™ and Genetic Communities™.

In this paper, we describe the steps we take to identify and interpret segments of DNA that are identical-by-descent between individuals. We begin with an introduction to the key concepts behind DNA matching, explain the challenges in identifying matches, and finally we describe how we tackle the problem of detecting IBD in a large genetic database.

1.1. How DNA is inherited—a brief primer

To illustrate the concept of inheritance from a common ancestor, consider the small family in Figure 1.1. Humans have 22 pairs of chromosomes in which one chromosome is inherited from the father and one from the mother (the sex-linked chromosomes X and Y have a different inheritance pattern, and are not included in this example). In Figure 1.1, each family member is represented by a pair of just one of the 22 pairs of chromosomes (the two colored bars), but the same concepts we illustrate apply equally to all 22 pairs of chromosomes.

The chromosomes are shown in four colors—two shades of blue inherited from the father and two shades of red inherited from the mother.

Observe that each child inherits an equal amount of DNA (50%) from the mother (red) and the father (blue), since the child inherits one copy of each chromosome from each parent. Also, observe that each of the child's chromosomes is a mixture of each parent's two chromosome copies. Each child has one light and dark blue mixture from the father and one light and dark red mixture from the mother. This mixture is different in each child. The biological process responsible for the transmission of chromosomes from parents to child in this way is what is called meiosis. The random assortment of these chromosome fragments during meiosis is called recombination. The end result is that each child's DNA is a random mixture of DNA from his or her two parents.

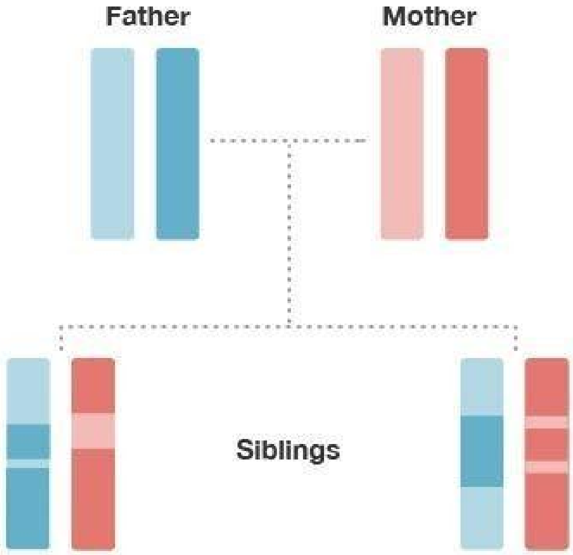


Figure 1.1: *Illustration of inheritance of DNA from parents to children. Each family member is represented by a pair of chromosomes inherited from their two parents. The chromosomes are colored to indicate DNA inherited from the same grandparent. The chromosomes of each child are a mixture of the chromosomes of his or her parents.*

Comparing the chromosomes of the siblings, lining them up from top to bottom (Figure 1.1), we observe that some regions of the chromosomes have the same color in each sibling. This indicates that they have almost identical sequences of DNA at those locations on their chromosome. These locations on the chromosome are called "identical-by-descent" (IBD) because they were inherited from a common ancestor (in this case, the common ancestor is the mother or the father).

When we compare less closely related individuals, they usually have shorter and fewer IBD segments. Figure 1.2 depicts the chromosome pairs for three 5th cousins sharing the same two common ancestors (great-great-great-great-grandparents). In this case, these three 5th cousins have each inherited only a small proportion of their DNA from the two common ancestors. Also, notice that because the transmission of DNA (through meiosis) has repeated several times over several generations, DNA from different common ancestors (red and blue) can end up on the same chromosome of an individual. Note that the gray portions of the chromosomes are inherited from other ancestors that are not shown in the diagram and may or may not contain segments that are IBD among the three 5th cousins.

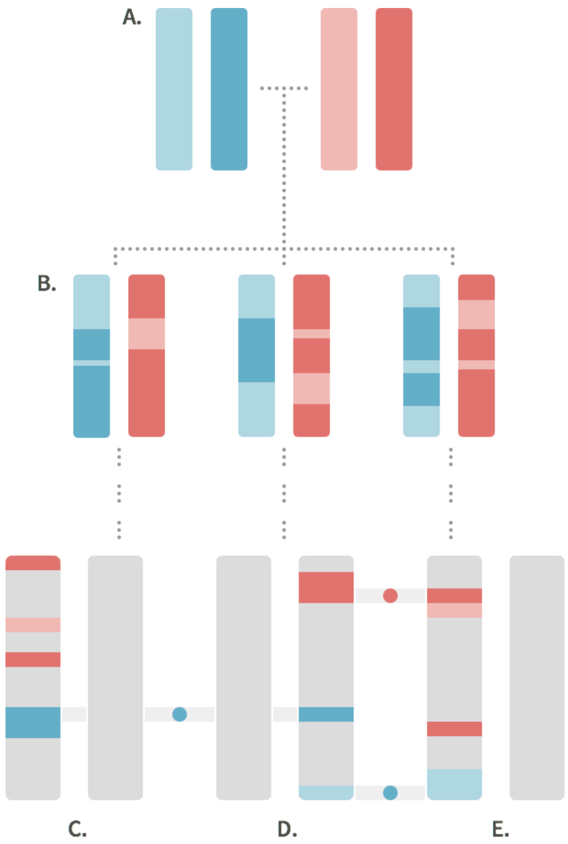


Figure 1.2: *Illustration of DNA that is identical-by-descent between distant cousins (C, D, E). Chromosomes of the common ancestors (A) and their children (B) are shown. Chromosomes of other intermediate generations are not shown in the diagram. The blue and red circles indicate chromosome segments that are IBD between the indicated chromosomes. See the caption of Figure 1.1 for more details.*

While the three 5th cousins in Figure 1.2 have all inherited some DNA from the common ancestors shown in the figure, only a few short segments of the chromosomes are actually identical in the same places on the chromosome of different cousins. In this example, we see that only 3 short chromosome segments, indicated by the blue and red circles, are IBD. One segment of DNA is shared by cousins C and D, and two segments are shared by cousins D and E. By contrast, cousins C and E, despite the fact that they are related through their great-great-great-great-grandparents (A), do not have *any* identical DNA that is IBD through these two common ancestors.

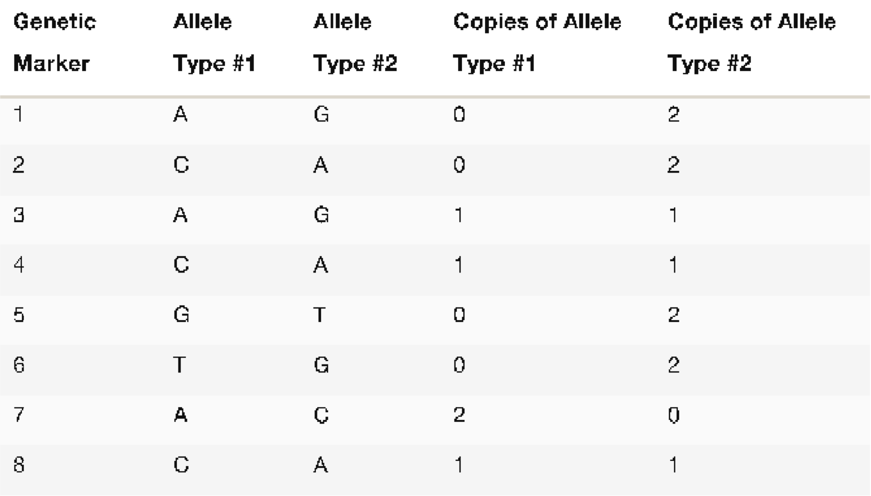
The first goal of DNA matching is to accurately identify the DNA segments on the 22 chromosome pairs that are identical-by-descent between pairs of individuals. Importantly, we would like to identify these IBD segments for every pair of customers in our database. Doing this accurately as well as efficiently for millions of people is not a trivial problem, and is an active area of research in the scientific community.

1.2. Genotype phasing

The first obstacle we face is that although DNA is transmitted from parent to child in long sequences, we do not have direct access to these exact sequences. (It is currently a prohibitively expensive and time-consuming process to read the exact DNA sequence inherited from each parent.) Instead, we only observe the unordered pairs of nucleotides—the basic building blocks of DNA, typically represented as A, T, G or C—at a small fraction of locations in the genome. This means that we only sample a small fraction of the complete DNA sequence, and we do not necessarily know which nucleotide came from the mother and which came from the father.

To better appreciate how this complicates identification of IBD, consider the genetic data in Table 1.3. This table illustrates how we represent customer genetic data in our database. At 8 specific DNA locations, or genetic markers, we have sampled the genotype from a single individual. The genotype is the pair of nucleotides present on the two chromosomes for an individual at a given genetic marker. (For more details on how these genetic markers are chosen, see *Ancestral Regions White Paper*). For example, at the first genetic marker, sometimes we observe individuals that have the “A” nucleotide (A stands for the nucleotide base adenine), and other times we observe individuals that have the “G” nucleotide (G refers to guanine). In other words, at this precise DNA location, we will either observe an A or G in an individual’s DNA. All the genetic markers we use are “polymorphic” (changing) in only a single nucleotide, hence they are called “single nucleotide polymorphisms,” or SNPs for short. At most SNPs, we observe only 2 possible nucleotides. Geneticists call these two possibilities “alleles.”

Since each person has two chromosome copies (one inherited from each parent), for a single individual we can either observe two A’s, two G’s, or an A and a G. In this example, at the first marker we observe two copies of the G allele in the person’s genotype. SNP observations are easily stored in our databases as 0’s, 1’s and 2’s, representing the number of times we observe a specified allele in the genotype.



Genetic Marker	Allele Type #1	Allele Type #2	Copies of Allele Type #1	Copies of Allele Type #2
1	A	G	0	2
2	C	A	0	2
3	A	G	1	1
4	C	A	1	1
5	G	T	0	2
6	T	G	0	2
7	A	C	2	0
8	C	A	1	1

Table 1.3: *Example of a small amount of genetic data from a single individual at 8 genetic markers. The genetic data are unordered pairs of nucleotides, or genotypes, which we can represent as numbers—0, 1 or 2—for the number of times each of the alleles is observed in the genotype.*

In Table 1.3, at genetic locations 1, 2, 5, 6 and 7, the mother and father have transmitted the same allele to the child. As a result, we can tell directly from the genotype the value on each of the two chromosomes (i.e., each chromosome has an G for marker 1). On the other hand, consider genetic marker 4. In this case, the individual's genotype is an A and a G; we do not know whether the A comes from the father and the G comes from the mother, or vice versa.

If we want to compare individual chromosomes to identify which segments are IBD, we need to know the sequence of alleles (letters) on each chromosome. This first requires that we determine the assignment of alleles to chromosomes; for example, we need to assign the A allele to mom or dad's chromosome and the T allele to the other parent's. We need to do the same for markers 3 and 8 as well. The process of determining the assignment of allele copies to chromosomes is called genotype phasing. In section 2, we describe our approach to this problem.

1.3. Finding matching segments

Once the phasing is complete—that is, once we have assigned the two allele copies of each genetic marker to each of an individual's two chromosomes—the second step is to identify identical DNA sequences between all pairs of individuals in the customer database. This is challenging because it involves comparing a very large number of sequences. AncestryDNA's database contains tens of millions of genotyped DNA instances, representing hundreds of *trillions* of pairs of individuals to check for matching segments. An additional complication is that the database is not static—it is continuously growing as more people take the AncestryDNA test.

Quantitative geneticists have developed very fast software such as GERMLINE (Gusev *et al.*, 2009) and Parente (Rodriguez *et al.*, 2015) to identify matches in a large number of genotype samples. We have developed our own implementation of GERMLINE, specifically designed to compute IBD in a growing database.

1.4. Assessing informativeness of matches for relationship estimation

Detecting matches enables us to estimate relationships between people. In general, the more identical segments of DNA shared between two people, the more likely it is that the two people share a recent common ancestor (refer to Figures 1.1 and 1.2). In practice, however, the IBD we detect may reflect other factors, such as selective pressures (Albrechtsen *et al.*, 2010), or more distant shared genealogy, in which case this IBD will confound the relationship estimates. An additional consideration is that since shorter IBD segments are difficult to identify accurately, a large proportion of shorter IBD segments that we detect could be false, and therefore could contribute errors to relationship estimation. In order to improve the accuracy of our relationship estimates, we have developed an approach to quantify the “informativeness” of IBD for estimating relationships. IBD segments that are expected to be less informative of recent relationships contribute less evidence to the relationship estimate. We describe this process, called “Timber,” in Section 4.

1.5. Estimating relationships

Finally, the fourth challenge is how to translate the identification of IBD segments to accurate relationship estimates. Identical twins are IBD across their entire genome, and parent-child pairs are IBD on half their chromosomes. Beyond this, however, due to the random process of meiosis and recombination, the exact relationship between two individuals is uncertain based on IBD alone. On average, more closely related people are IBD across a greater portion of their genomes, but the correspondence between amount of matching and the actual pedigree relationship is variable.

To develop a method for accurately estimating relationships from IBD, we use genetic data from thousands of pairs of individuals with known family relationships (either real people with documented pedigrees or simulated individuals with known pedigrees). Additionally, we use other information beyond IBD inferred from genetic data to ensure that our estimates of close relationships—specifically, parent-child and sibling relationships—are as accurate as possible. Methods for relationship estimation are detailed in section 5.

See Figure 1.4 for an overview of the matching and relationship analysis pipeline.

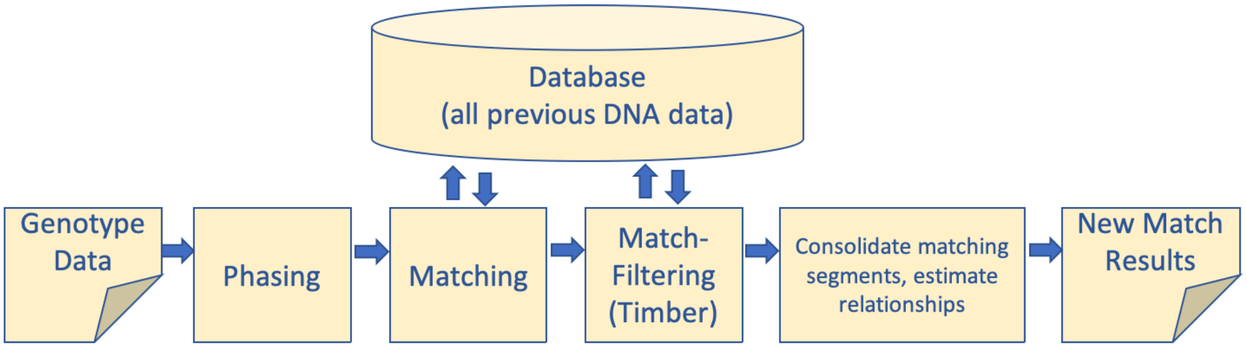


Figure 1.4: *Overview of the Matching and analysis pipeline.*

2. Genotype phasing algorithm

2.1. Introduction

As explained in section 1.2, genotypes alone often cannot tell us which allele copy was inherited from the father and which was inherited from the mother. One exception to this is when we have genotypes sampled from both parents and child (called a trio). In this case, since the laws of genetic inheritance tell us that alleles can only be transmitted from parent to child in specific ways, we can use this information to very accurately assign alleles to each of the two chromosome copies. However, since we cannot depend on all customers taking the AncestryDNA test with both their parents, we need a more sophisticated approach that can accurately determine the phase of the genotypes—the assignment of alleles to chromosome copies—without parental information.

Our strategy relies on two sequential processes. The first simultaneously phases the genotypes of a person using data from a large number of unrelated individuals. The basic principle is to prefer a phase that results in two sequences on each of the chromosomes that are also observed in many other samples. To do this, we have developed a modified strategy called "Underdog", based on the software BEAGLE (Browning and Browning, 2007). Underdog learns haplotype frequencies in a large number of AncestryDNA samples. Then, once we have learned haplotype frequencies, we use Underdog to quickly phase the genotypes of new customers.

As the number of individuals in the Ancestry DNA database has grown, we realized that we can substantially improve our phasing accuracy by leveraging the genotypes of related individuals through their shared DNA matches and segments of IBD. We refer to this phasing technology as SideView. This approach is based on two main assumptions. First, most of the time two individuals are related to each other through only one common ancestor (and therefore one familial line). Second, with a large enough database of individuals you can expect that the segments of IBD an individual shares with their matches will collectively overlap entire chromosomes and will include instances of IBD that span chromosomes. An important benefit of this approach is that phasing accuracy improves as more related samples are added to the database.

2.2. The Underdog genotype phasing algorithm

Underdog phasing is our approach to learn haplotype frequencies from a large number of genotyped samples that serve as a training set, and then quickly phase new genotype samples based on the learned haplotype frequencies. Underdog is a modification of the phasing algorithm BEAGLE (Browning and Browning, 2007), which builds a statistical model that summarizes the distribution of haplotypes in a population and uses the estimated haplotype distribution to estimate genotype phase. To make this computation feasible, we subdivide each chromosome into small segments (or "windows") of 500 SNPs each, and we separately build a haplotype-cluster model for each of these chromosome windows. The probability distribution over haplotypes in a window is defined using a Markov model (Browning, 2006).

More formally, within a single 500-SNP window, BEAGLE takes as input (1) a reference set *R* of genotypes previously phased with very high accuracy (e.g., genotypes belonging to trios), and (2) a query set *U* of unphased genotypes (see Appendix A). BEAGLE starts by randomly assigning phase to the genotypes *U*. Then it builds a new set of haplotype-cluster models from the randomly phased genotypes *U*, and the previously phased genotypes *R*. These haplotype-cluster models are then used to estimate a new (and hopefully more accurate) phase for genotypes *U*. The process iterates until the haplotype-cluster models converge to a solution. This final set of haplotype-cluster models is used to compute the most likely phase for each genotype in *U.* The final phased genotype sample is combined from the phase estimate in each window. For more details on the BEAGLE algorithm, consult the pseudocode in Appendix A-B and the original publication (Browning and Browning, 2007).

An important limitation of BEAGLE is that the computational expense of the model-building process increases with the size of *R* and *U* . Further, the output from BEAGLE cannot be easily reused to phase new genotype samples. To surmount these limitations, we propose an alternative approach: we learn haplotype-cluster models once from a large training set of phased genotypes, we store the learned models to file, then we use these models to quickly phase new genotype samples. We call this enhanced approach Underdog phasing, and describe it in more detail in appendix B. In section 2.4, we describe our experiments that demonstrate improvements in computational cost and phasing accuracy using Underdog.

2.3. SideView phasing algorithm

At a high level, SideView phases a person's genome by using overlapping segments of IBD from multiple DNA matches that represent opposite familial lines of descent (see Noto and Ruiz, 2022). Specifically, it finds sections of a person's (proband) DNA where they match two different individuals (matches) who do not match each other. This allows us to separate the proband's DNA into distinct haplotypes representing the DNA sequences inherited from each parent.

Our approach with SideView is essentially to identify segments of IBD and use them as surrogate parents, thereby enabling duo- or trio-phasing of those parts. Importantly, because of our database size, there are enough IBD segments that we can expect most IBD segments to overlap others on both sides of the family, and most sites to overlap multiple IBD segments. These provide information on which allele is part of a shared haplotype even if those IBD segments are shared between unphased diploids. Figure 2.1 shows a small illustration of the type of data we use.

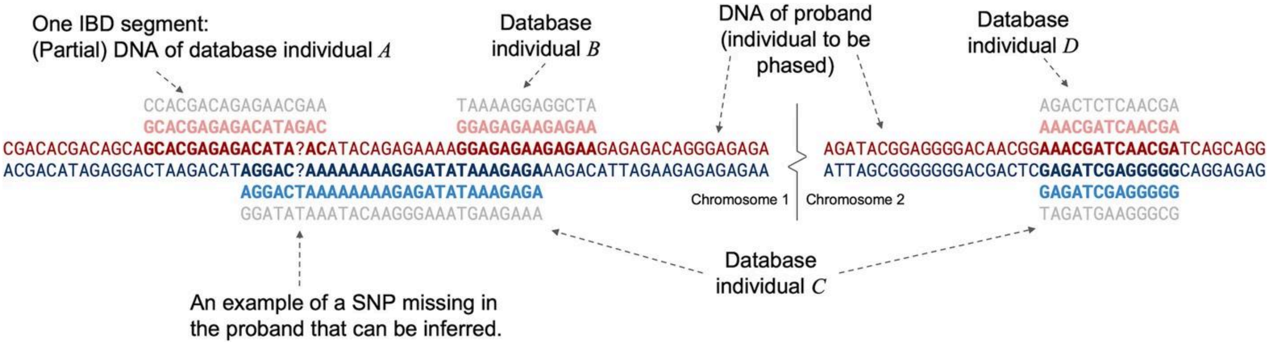

One IBD segment:
(Partial) DNA of database individual A
Database individual B
DNA of proband (individual to be phased)
Database individual D
CCACGACAGAGAACGAA
GCACGAGAGACATAGAC
TAAAAGGAGGCTA
GGAGAGAAGAGAA
AGACTCTCAACGA
AAACGATCAACGA
CGACACGACAGCAGCACGAGAGACATA?ACATACAGAGAAAAGGAGAGAAGAGAAGAGAGACAGGGAGAGA
AGATACGGAGGGGACAACGGAAACGATCAACGATCAGCAGG
ACGACATAGAGGACTAAGACATAGGAC?AAAAAAAAGAGATATAAAGAGAAAGACATTAGAAGAGAGAGAA
ATTAGCGGGGGGGACGACTCGAGATCGAGGGGGCAGGAGAG
AGGACTAAAAAAAAGAGATATAAAGAGA
GAGATCGAGGGGG
Chromosome 1
Chromosome 2
GGATATAAATACAAGGGAAATGAAGAAA
TAGATGAAGGGCG
Database individual C
An example of a SNP missing in the proband that can be inferred.

Figure 2.1: *An illustration of the SideView approach to phasing. The DNA of the proband consists of two haploid genotypes (red and blue) across all chromosomes. IBD segments also consist of two haplotypes, one of which is identical to one of the proband's haplotypes. Note that each IBD segment (partial diploid) is consistent with exactly one of the proband's haplotypes, and that we can infer that the same haplotype in the proband is shared with individual A and individual B, even though those segments do not overlap (because they both overlap with individual C). Individual C's two IBD segments are on different chromosomes and it is more likely than not that the haplotypes shared with individual C are inherited from the same parent (with dozens or hundreds of multi-chromosome IBD segments, inter-chromosome phasing becomes clearer). Reproduced from Noto and Ruiz, 2022.*

When IBD segments overlap, there is essentially only one way to assign each IBD segment to one parental haplotype or the other and thereby phase the proband's DNA. Given enough overlapping IBD segments, these blocks can extend to the full size of a chromosome. We refer to these blocks of overlapping IBD segments separated into two parental groups as subclusters.

A proband's DNA may be organized into several separately phased subclusters. In order to fully phase the proband's DNA, we must determine which parental group of each subcluster corresponds to the same parent in the other subclusters. We do this by organizing the subclusters into larger superclusters and align the parental groups of each subcluster. We illustrate this process in Figure 2.2.

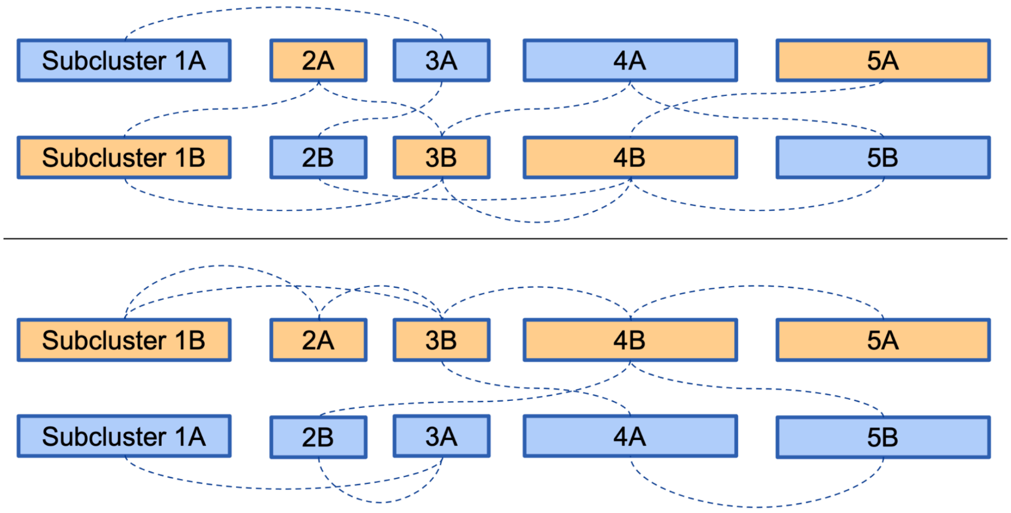


Figure 2.2: *An illustration of organizing subclusters for a proband's DNA into larger superclusters aligned with parental groups. At top, we show the proband's DNA is phased in five subclusters (1-5). The color indicates the true parent from which IBD segments are inherited. The letter A or B indicates which parental side each subcluster was originally assigned. Dotted lines indicate a connection between groups where the proband shares DNA with the same person in both groups. At bottom, we show the result of phasing these subclusters into one supercluster such that the number of connections between groups on the same side of the family is maximized. Reproduced from Noto and Ruiz, 2022.*

The primary mechanism for aligning the parental groups of subclusters on different chromosomes is based on individuals that share IBD segments across multiple subclusters. For example, in Figure 2.1, individual C shares DNA with the proband on chromosomes 1 and 2. It is generally more likely that the proband and individual C inherited both of the shared haplotypes from the same side of the family. When our IBD data consist of many instances where the proband shares DNA across multiple subclusters, we can phase the subclusters correctly by maximizing the number of instances where multi-segment IBD data are inherited from the same parent in a supercluster (Figure 2.2).

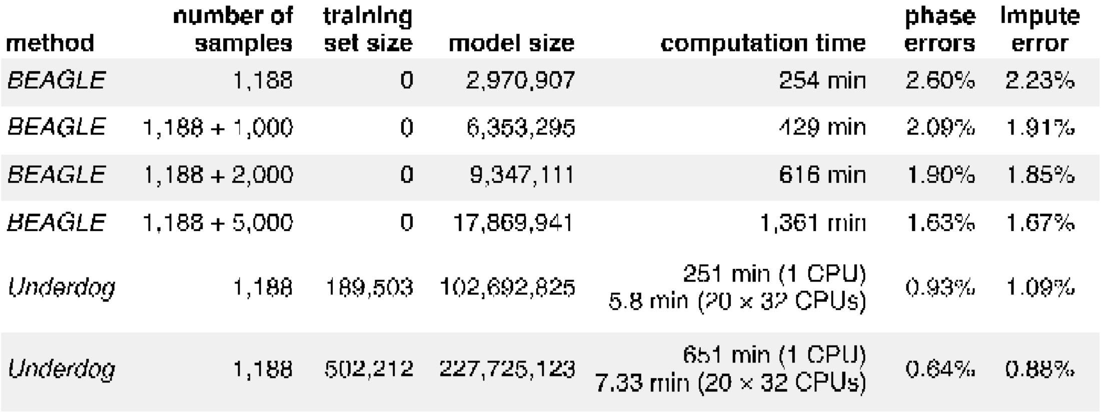
Depending on the amount and distribution of IBD data for a proband, it is possible that some parts of the genome do not overlap any IBD data and cannot be phased. In these rare cases, we default to the phase inferred by Underdog models.

2.4. Evaluation of genotype phasing algorithms

To evaluate the performance of our modified phasing algorithm Underdog, we compared its run time and phasing accuracy applied to datasets of different sizes against the runtime and accuracy of BEAGLE. We evaluated phasing accuracy on a test set of 1,188 unrelated individuals from our database that have been phased accurately because they each belong to a trio and were phased using parental information (that is, we used the genotypes of both parents to determine phase, but we do not include the parents in the test set available to BEAGLE and

Underdog). To assess phasing accuracy, we consider only genotypes that can be phased unambiguously in the trio. Another evaluation metric we use is impute error—the rate at which genotypes are incorrectly estimated when 1% of genotypes are set to missing uniformly at random.

Our evaluation showed that Underdog infers the phase of new genotype samples more accurately than BEAGLE—and with much lower computational cost—provided that we are able to make use of a very large panel of phased genotypes (Table 2.1). Underdog is able to achieve high accuracy because it can benefit from hundreds of thousands of samples. (As of 2020, customers taking the AncestryDNA test are phased using a panel of one million genotypes.)



method	number of samples	training set size	model size	computation time	phase errors	impute error
BEAGLE	1,188	0	2,970,907	254 min	2.60%	2.23%
BEAGLE	1,188 + 1,000	0	6,353,295	429 min	2.09%	1.91%
BEAGLE	1,188 + 2,000	0	9,347,111	616 min	1.90%	1.85%
BEAGLE	1,188 + 5,000	0	17,869,941	1,361 min	1.63%	1.67%
Underdog	1,188	189,503	102,692,825	251 min (1 CPU) 5.8 min (20 × 32 CPUs)	0.93%	1.09%
Underdog	1,188	502,212	227,725,123	651 min (1 CPU) 7.33 min (20 × 32 CPUs)	0.64%	0.88%

Table 2.1: *Results from an experiment comparing phasing accuracy using BEAGLE version 3.3.2 with data sets of different sizes against phasing accuracy using Underdog with a much larger reference panel of 189,503 samples, in which these samples were phased in large batches using HAPI-UR. These are results for chromosome 1 only. We run BEAGLE using default parameters, except we set n = 20 (this is the number of phasing estimates that are simulated for each genotype sample). Phasing error is evaluated in a test set with 1,188 trio-phased samples. Phasing error, or "switch-error rate," is calculated as the rate of disagreement between the estimated phase and the trio-phased haplotype, only for loci in which phase can be determined unambiguously; i.e., sites with at least one homozygous individual in the trio (Williams, 2012). “Model size” refers to the total number of haplotype-cluster model states across all chromosome windows. For Underdog, we show two computation times: the total time taken to complete the computation on a single CPU, and the computation time on a Hadoop cluster with 20 32-core compute nodes (we use the MapReduce framework; see Dean and Ghemawat, 2008).*

Separately, we evaluated the performance of our SideView phasing algorithm using a dataset of 30,000 child-parent-parent trios and a database of over 12 million genotypes. We used SideView in combination with the 12 million genotypes to phase the 30,000 child genomes, without using the parents genotype or IBD data. We then compared the resulting SideView phasing to trio phasing of the 30,000 child genomes based on the parents’ data. We find that SideView is able to separate the DNA inherited from each parent in our test set with an average accuracy over 95% (and a median accuracy of nearly 98%). Phase accuracy results are shown in Table 2.2 (Noto and Ruiz, 2022).

Criteria	SideView median (%)	SideView mean (%)
Global phase error	2.09	4.93
Phase switch error rate	0.26	0.33
Proportion of SNPs in 1 cM+ runs	99.01	98.64

***Table 2.2:** Results from an experiment evaluating the genomewide phase accuracy of SideView. Accuracy is measured using trio phase as the standard for true phase, using only SNPs where the phase can be unambiguously inferred from the trio (i.e., where at least one parent is homozygous). Global error is the rate at which the phase differs from trio phase (keeping only one haplotype assigned to one parent across the genome, but assuming the more favorable haplotype of two choices). Switch error rate is the frequency with which the phase of a heterozygous SNP differs from that of the previous heterozygous SNP with respect to trio phase. The third accuracy measure is the proportion of SNPs that belong to segments where there are no phase switches for at least 1 centimorgan. Some measures depend on SNP density, and we consider 416,176 SNPs across the autosome. The median global phase error of the pre-phased data is 48%, and the median phase switch rate and median proportion of SNPs in 1 cM+ runs is 1.08% and 95% respectively*

3. Detecting IBD

3.1. Matching Algorithm

Once we have estimated the phase of each genotype sample, we turn to the problem of finding IBD segments, or “matches,” shared by pairs of samples. This effectively reduces to the problem of finding long sequences (strings of A’s, T’s, G’s and C’s) that are identical in pairs of chromosomes. However, there are several practical issues that arise due to the peculiarities of genetic data, as well as the size of our data set, that make this problem more complex than it might first appear. In this section, we first describe our approach, then explain how this approach addresses some of the common problems in finding matches from phased genotype data.

Our general strategy is divided into 5 steps. We illustrate the individual steps in Figure 3.1.

1. Subdivide each chromosome into short segments, which we call “windows.” In our implementation, all windows contain exactly 96 SNPs. This number was chosen to balance computational cost and accuracy. (Note that these windows are not the same as the ones chosen for genotype phasing [see Figure 3.1, section B] and that we use 10 SNPs per window in the example in order to make it easier to follow.)
2. For each pair of individuals, identify windows in which the alleles at all SNPs in one of the individual’s two phased haplotypes are identical to all the alleles at the same positions in one of the other individual’s phased haplotypes. We call these “seed matches” (see Figure 3.1, section D).

3. For each seed match, we attempt to extend the seed match in both directions along the chromosome until (a) the beginning or end of the chromosome is reached, or (b) a homozygous mismatch is detected. A homozygous mismatch is a pair of genotypes at the same SNP that are incompatible regardless of how they are phased (for example, one individual has two A's and the other individual has two G's). The estimated IBD region is defined by the start and end positions of the SNPs included in the extended segment (see Figure 3.1, section D).
4. Calculate the length of the candidate matching segment in terms of genetic distance, measured in centimorgans (cM). Genetic distance is proportional to the expected rate of recombinations along that stretch of chromosome. Since individual chromosomes accumulate recombination events through successive generations of inheritance, IBD segments spanning large genetic distances suggest more recent inheritance. Below, we explain how we use the genetic distance of detected IBD segments to estimate relationships.
5. The segment is retained as part of a match in our database, provided that (i) the segment is at least 6 cM, (ii) the segment is part of a match with segment lengths that sum to at least 8 cM, and (iii) the match is not dismissed as identity by state (see Section 4.2).

The procedure we have outlined here is described in Gusev *et al.*, 2009.

As we described in step 2, we use the phased genotypes to identify seed matches. In the example (Figure 3.1), we identify 2 seed matches in 2 adjacent windows. Next, we extend the candidate IBD segment until a homozygous mismatch is encountered. In the example, the error in the estimated phase here does not prevent SNPs in this window from being included in the IBD segment. This illustrates the importance of not relying solely on the haplotype sequences identified in the genotype phasing step to identify IBD segments. Although our phasing is very accurate overall, even small amounts of phasing error will confound detection of long segments that are IBD. Our solution is to use the phased genotypes to suggest initial candidates (seed matches), then in step 3, we use the unphased genotype data to extend the matches. In this example, the matching segment is extended across most SNPs shown in the figure, and is nearly identical to the length of the ground-truth IBD segment.

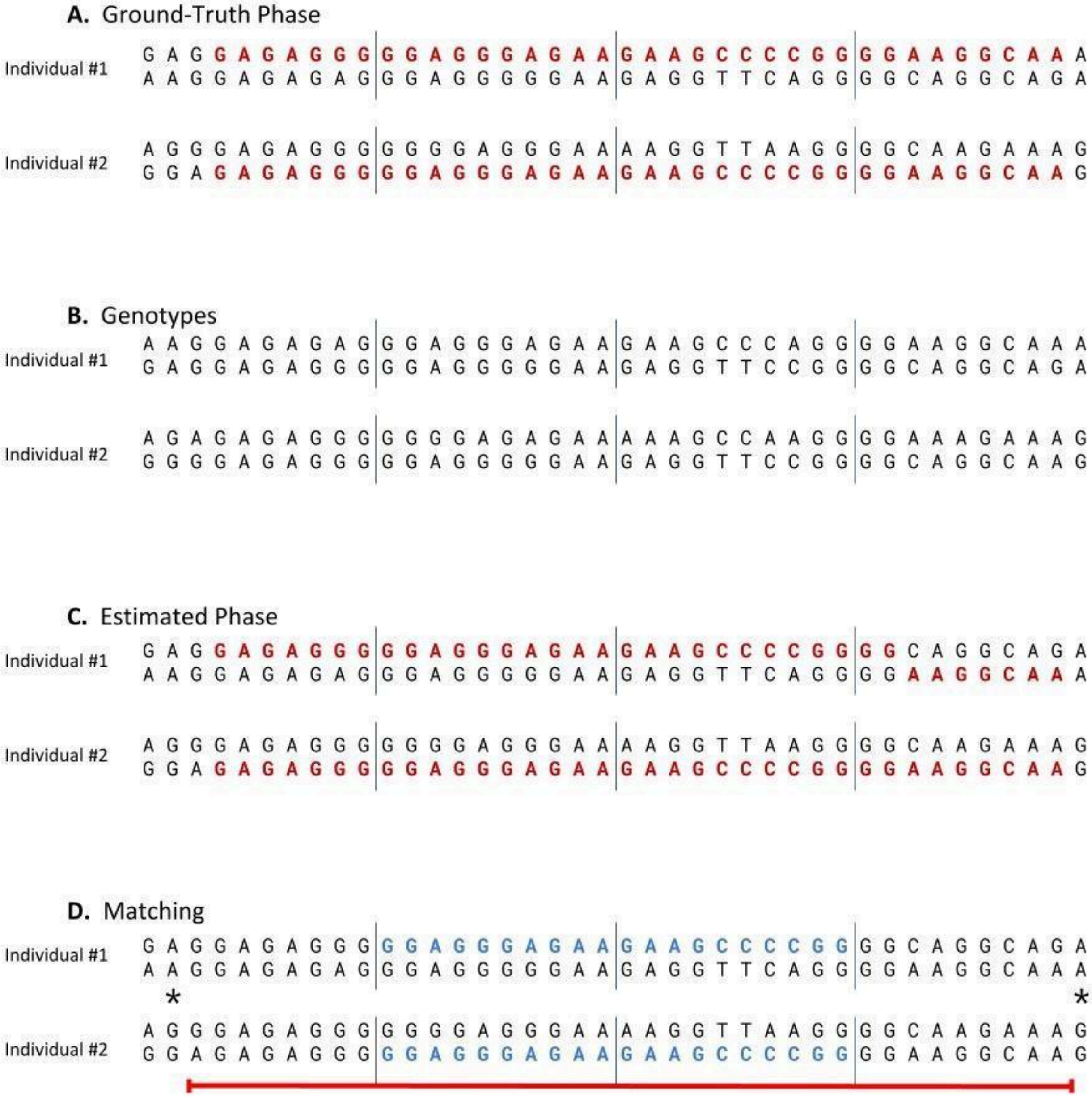

A. Ground-Truth Phase
Individual #1
G A G G A G A G G G | G G A G G G A G A A | G A A G C C C C G G | G G A A G G C A A A
A A G G A G A G A G | G G A G G G G G A A | G A G G T T C A G G | G G C A G G C A G A
Individual #2
A G G G A G A G G G | G G G G A G G G A A | A A G G T T A A G G | G G C A A G A A A G
G G A G A G A G G G | G G A G G G A G A A | G A A G C C C C G G | G G A A G G C A A G
B. Genotypes
Individual #1
A A G G A G A G A G | G G A G G G A G A A | G A A G C C C A G G | G G A A G G C A A A
G A G G A G A G G G | G G A G G G G G A A | G A G G T T C C G G | G G C A G G C A G A
Individual #2
A G A G A G A G G G | G G G G A G A G A A | A A A G C C A A G G | G G A A A G A A A G
G G G G A G A G G G | G G A G G G G G A A | G A G G T T C C G G | G G C A G G C A A G
C. Estimated Phase
Individual #1
G A G G A G A G G G | G G A G G G A G A A | G A A G C C C C G G | G G C A G G C A G A
A A G G A G A G A G | G G A G G G G G A A | G A G G T T C A G G | G G A A G G C A A A
Individual #2
A G G G A G A G G G | G G G G A G G G A A | A A G G T T A A G G | G G C A A G A A A G
G G A G A G A G G G | G G A G G G A G A A | G A A G C C C C G G | G G A A G G C A A G
D. Matching
Individual #1
G A G G A G A G G G | G G A G G G A G A A | G A A G C C C C G G | G G C A G G C A G A
A A G G A G A G A G | G G A G G G G G A A | G A G G T T C A G G | G G A A G G C A A A
*
*
Individual #2
A G G G A G A G G G | G G G G A G G G A A | A A G G T T A A G G | G G C A A G A A A G
G G A G A G A G G G | G G A G G G A G A A | G A A G C C C C G G | G G A A G G C A A G

Figure 3.1: *IBD detection example in two DNA samples at 40 consecutive genetic markers (SNPs). In A, we show the (unobserved) ground-truth sequences at the 40 SNPs, highlighting in red the pair of sequences that are IBD. B shows the genotype data—unordered pairs of alleles at 40 SNPs—that are available in our data. Note that genotypes "AG" and "GA" are identical because the order of the alleles in the genotype is not informative. These genotypes are subdivided into 4 windows each containing, for illustration only, 10 SNPs. C shows the genotype phase—assignment of the alleles to the two chromosome copies—that is estimated by Underdog, highlighting in red the same IBD segment in A. Observe that Underdog incorrectly phases the 7 right-most SNPs in the IBD segment. D shows the results of matching by GERMLINE to the phased genotypes shown in C. First, two windows containing seed matches are identified. The seed matches, highlighted in light blue, are identical sequences within a window. Second, beginning with one of the seed matches, the matching segment is extended in both directions until a homozygous mismatch is identified. The homozygous mismatches are indicated with an asterisk (*). The*

final IBD segment spans 37 SNPs, as indicated by the orange bar. This is nearly identical to the SNPs spanned by the ground-truth IBD segment (shown in A). The only error is the inclusion of an additional SNP on the left-hand side that is reached before a homozygous mismatch.

An important feature of our method is that we do not keep track of all matching segments; in step 5, we filter out a candidate match if its genetic distance is less than 8 cM. The cutoff of 8 cM was chosen after considering several factors. The first factor is data storage. Since the number of matching segments grows exponentially with decreasing length, we dramatically reduce the storage requirements of our matching database by increasing the cutoff. A second, and more critical, factor is that the accuracy of IBD detection drops rapidly with decreasing IBD length—that is, the shorter the length of the detected IBD segment (expressed in genetic distance), the less likely it is that the detected chromosome segment is truly inherited from a common ancestor.

To illustrate the phenomenon of decreasing accuracy with decreasing IBD length, we examine concordance of matching between parent and child using the described IBD detection strategy. Typically, if two individuals, X and Z, are IBD across a given chromosome segment, then we would expect that Z is also IBD with at least one of the parents of X. Therefore, we can assess accuracy of IBD detection by quantifying concordance of IBD between parents and child; more accurate IBD detection should yield better parent-child concordance.

Figure 3.2 summarizes IBD detected in 20,000 matches chosen so that for every match between individuals X and Z, there is a corresponding match detected between individuals Y and Z, such that Y is a parent of X. As expected, most of the points in the scatterplot cluster around the diagonal (the dotted orange line); for these points, the amount of IBD detected in the child is nearly identical to the amount of IBD detected in the parent. However, as we move toward the bottom-left corner of the plot, more and more points are distributed away from the diagonal This shows that concordance is not as strong for smaller amounts of IBD. (Note that the smaller number of points away from the diagonal near 5 cM is an artifact due to the fact that we are only looking at pairs with total IBD at least 5 cM.)

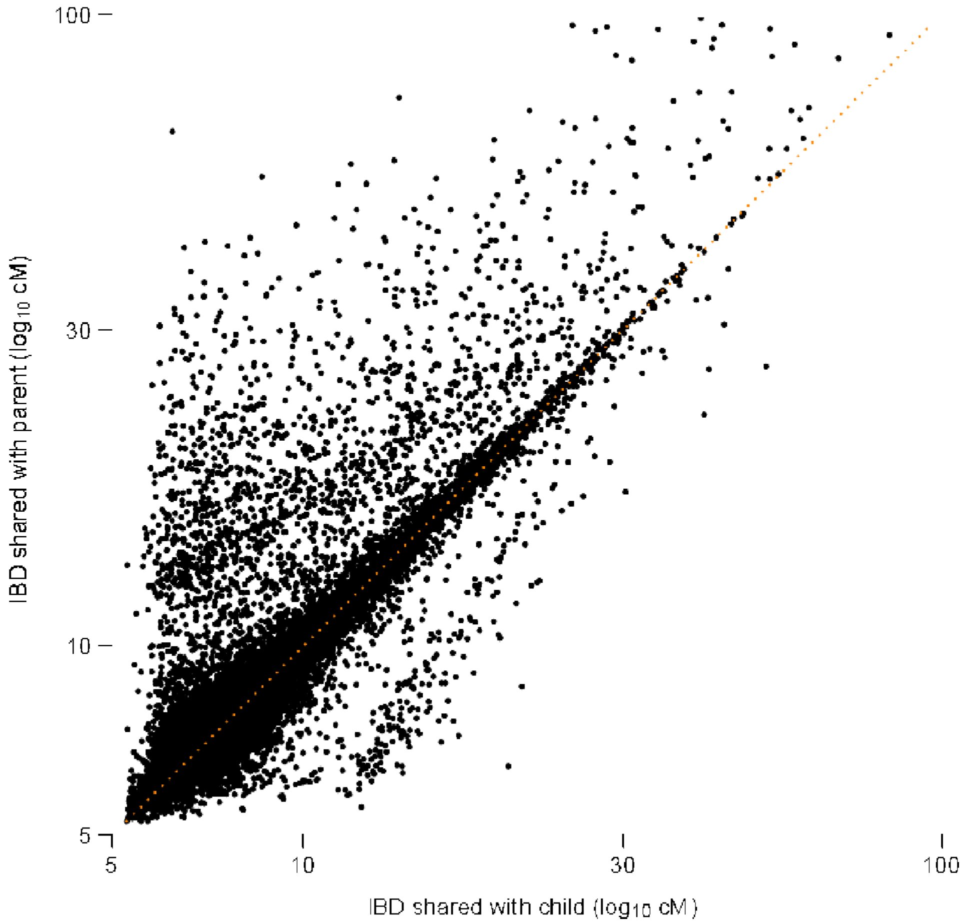


Figure 3.2: *Concordance of matching between child and parents. Each point in the scatterplot corresponds to triple (X,Y,Z) such that individuals X and Z share IBD > 5 cM, individuals Y and Z share IBD > 5 cM, and individual Y is a parent of X. A total of 20,000 such triples are plotted in this figure. The horizontal and vertical axes give the total IBD detected (in cM). Note that IBD is shown on the logarithmic scale and only for IBD < 100 cM.*

We take a second look at this concordance in Figure 3.3. Here, we quantify concordance by counting the number of times that IBD is shared with the mother, father, or both parents, stratified by total IBD length in the child in cM. (We do not compare exact locations of IBD segments, only total IBD length between pairs of individuals.) As the length of the detected IBD segment between child X and individual Y decreases, it is less likely that we also detect IBD > 6 cM between individual Y and one of X's parents. This indicates that detection of smaller amounts of shared IBD is less accurate. In other experiments, Durand *et al.* (2014) have shown that GERMLINE is particularly inaccurate for IBD segments less than 4 cM.

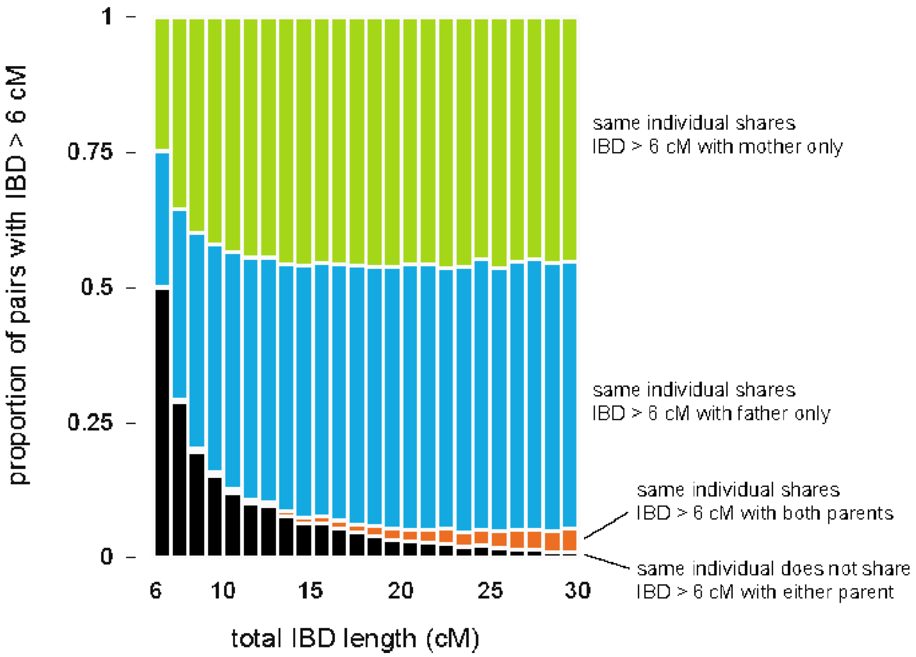


Figure 3.3: *Concordance of matching between child and parents. For a given total IBD length between child X and individual Y, we count the number of times that we detect IBD with this length and compare this to the number of times that we detect IBD (with total length > 6 cM) between the father of X (blue) and with the mother of X (green), and with both parents (orange). This figure is compiled from matching results on 16,178 mother-father-child trios.*

One complication is that accurate detection of IBD requires that we have a high density of SNPs in all regions of the genome. The array technology that we use to acquire the genotype data yields high-density SNP data across most of the genome, but there are some genomic regions with unusually low SNP density. This means that any matches that overlap these SNP-poor regions will be less reliable. To counteract this problem, we discount these matches by reducing their total length (in cM).

Another complication is that the identification of seed matches quickly becomes intractable as the number of DNA samples grows. We use hashing to avoid explicitly comparing every pair of haplotypes in each 96-SNP window (that would be billions of billions of comparisons). More precisely, we implement a hash function, *f(h,w),* that maps a character string *h* and window identifier *w* to an integer value. It has the property that if two different individuals have identical strings in the same window, they will have the same value of *f(h,w).* This makes it possible to quickly identify exact matches in a scalable fashion. Since the number of seed matches within a window is typically a very small proportion of the total number of chromosome pairs, hashing yields extremely fast detection of seed matches.

GERMLINE is able to efficiently and accurately identify IBD segments suggestive of recent common inheritance in a large database of genotypes. However, we cannot use GERMLINE directly for detecting matches among AncestryDNA genotype samples because GERMLINE was not designed to efficiently detect IBD in a growing database. Therefore, we have developed our own implementation.

4. Adjusting IBD for relationship estimation

4.1. Motivation

IBD detected between two genotype samples can be used to estimate a pedigree relationship because more closely related people have, on average, more DNA that is IBD. To improve the accuracy of this estimate, we first apply a simple algorithm that de-emphasizes the evidence from detected IBD (see section 3) that is less likely to be informative of close relationships. We call this algorithm “Timber.”

To understand the motivation behind this algorithm, it is instructive to examine matching results aggregated over a large number of samples. In Figure 4.1, we show aggregated matching results for three individuals selected from our database. For each of the 96-SNP windows used for IBD detection, Figure 4.1 shows the total number of IBD segments longer than 6 cM that were detected in pairs (*i*, *j*), in which *i* is the selected individual, and *j* is an individual in from a reference panel of over 300,000 genotypes (the Timber reference panel). Section A illustrates a common case in which IBD is detected in individual *i* with only a very small proportion of samples in the Timber reference panel within any given region of the genome. This reflects our expectation that very few pairs of individuals in the AncestryDNA database are closely related. By comparison, individual B has a substantially higher rate of matching with the Timber reference panel. Many factors could explain the different genome-wide rates of IBD shared by individuals A and B. For example, if we assume that IBD detection is equally accurate in individuals A and B, then demographic or historical factors could explain the different rates of matching; for example, one hypothesis could be that individual B’s ancestors have lived in the United States for a longer period of time, whereas individual A’s ancestors are more recent immigrants to the United States. Under this scenario, we would be more likely to find other relatives of individual B than individual A since, as of this writing, the vast majority of people who have taken the AncestryDNA test are from the United States. This illustrates a trend that we have observed more generally: the overall pattern of IBD can differ substantially from one individual to the next, and these differences may reflect different ancestral origins.

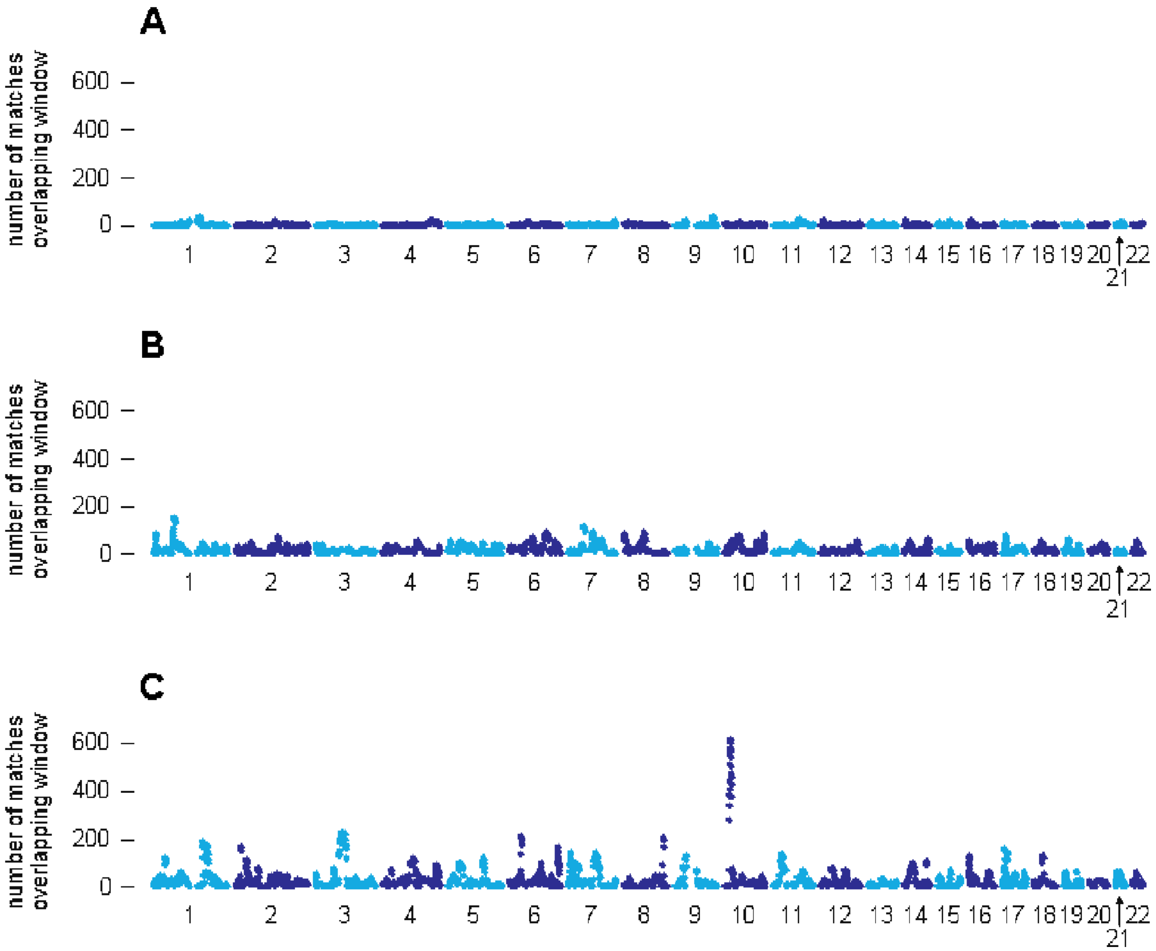


Figure 4.1. *A, B, and C show (separately for three individuals) match counts in all 96-SNP windows across the genome. More specifically, in each window on autosomal chromosomes 1 through 22, we count the number of times that the window overlaps an IBD segment detected between the given individual (labeled A, B, or C) and individuals included in a reference panel of 325,932 genotypes.*

Next, consider the individual in section C, who has a higher rate of matching than both individuals A and B. In addition, the matching rate is highly variable across the genome; certain regions, such as a region near the centromere of chromosome 3, and a region on chromosome 10, overlap with an unusually large number of detected IBD segments. If all detected IBD is due to inheritance from recent common ancestors, it is extremely unlikely that we would observe such excessive IBD in specific regions of the genome. This suggests that many of these spikes in IBD are unlikely to reflect recent inheritance from common ancestors. Instead, these spikes more likely reflect other demographic factors, or identity by state (IBS) (see, for example, Albrechtsen *et al.*, 2010). The implication is that IBD detected in regions with high rates of matching is expected to be less useful for estimating recent relationships.

Motivated by these observations, we have developed a procedure, Timber, that uses match counts aggregated over thousands of samples to inform relationship estimation. The strategy is to analyze matching results accumulated over a large number of genotype samples, then identify, separately for each individual, regions of the genome with unusually high rates of matching. Once we have identified these regions, we reduce the genetic distance of detected IBD segments overlapping these regions. We call these adjusted distances “Timber scores.” Since individuals can vary widely in genome-wide patterns of matching, as we observed in Figure 4.1, we run this analysis separately for each genotype sample. In the next section, we describe the Timber algorithm in greater detail.

4.2. The Timber algorithm

To compute Timber scores for all IBD segments, we take the following steps:

1. Select the Timber reference set, denoted by $\boldsymbol{R}$. Our reference set contains over 300,000 genotype samples.
2. Subdivide the genome into windows. Here, we use the same 96-SNP windows used to detect IBD. Let n be the number of windows.
3. For each sample i, and for each window, count the number of matches detected in GERMLINE between sample i and $i' \in \boldsymbol{R}$ that overlap the window. We represent these counts as a vector, $\mathbf{C}_i = (\mathbf{C}_{i,1},\ \mathbf{C}_{i,2},\ \ldots,\ \mathbf{C}_{i,n})$.
4. For each sample i, compute weights $\mathbf{W}_i = <\mathbf{W}_{i,1},\ \mathbf{W}_{i,2},\ \ldots,\ \mathbf{W}_{i,n}> = f(\mathbf{C}_i)$, in which each weight $\mathbf{W}_{i,j}$ is a number between 0 and 1, and f is a probability density function fitted to the matching data $\mathbf{C}_i$ for sample i. (Here we do not discuss the specification of this model, and the procedure for fitting this model to the data.)
5. Compute the Timber score for each matching segment. Let g be a matching segment detected in pair (i, i'), and let $j \in g$ be the set of all windows j that overlap segment g .

   The Timber score for segment g is defined as $TimberScore_g = \sum_{\in j \ g} dist(j) \times \mathbf{W}_{i,j} \times \mathbf{W}_{i',j}$, in which $dist(j)$ is the genetic distance spanned by the SNPs assigned to window j.

See Appendix C for a description of these same steps in pseudocode.

When an IBD segment does not overlap a region with an unusually high rate of matching, the final Timber score is nearly identical to the original length of the IBD segment. On the other hand, when some of the windows overlapping the segment exhibit an abnormally high rate of matching with the Timber reference panel, the Timber score will be smaller than the original genetic distance of the IBD segment.

One drawback to this procedure is that it considers each window in isolation, ignoring the information from neighboring windows on the same chromosome. To illustrate why this can be a limitation, consider the case when IBD between two individuals spans a large proportion of chromosome 1. In this case, we can usually be confident that the detected IBD was inherited from a recent common ancestor, and therefore it would not make sense to de-emphasize IBD which overlaps regions on the chromosome with an unusually high rate of matching. Thus, Timber is most useful for shorter IBD segments for which we have less confidence in the result. Therefore, we only apply Timber to matches with total IBD less than 90 cM.

In summary, we have used our large genetic database to identify unusual matching patterns, and by quantifying these unusual patterns, we adjust the relationship evidence separately for each individual. Timber improves relationship estimates for more distant relatives, such as 5th or 6th cousins, by downweighting the evidence from regions that are less likely to be informative of close relationships.

5. Estimating familial relationships from IBD

5.1. Background

As explained in section 1.1, more distantly related individuals (e.g., fifth cousins) are expected to inherit a smaller proportion of their genome from shared ancestors than more closely related individuals (e.g., first cousins). As we have also discussed, these chromosomal segments inherited from a common ancestor are said to be identical-by-descent (IBD). We have devoted much of this document to describing how we analyze an individual’s genotype to detect all IBD segments (greater than 8 cM) in our database in a way that balances accuracy and computational efficiency.

The final step in our analysis is to use the amount of detected IBD between a pair of individuals, following the Timber adjustments described in the previous section, to estimate a pedigree relationship for each pair of individuals who share one or more IBD segments. More specifically, the objective of relationship estimation is to infer, as accurately as possible, the number of meioses (see Figure 5.1) separating two individuals.

In Figure 5.1, we illustrate how the number of reproductive events, or number of meioses (see section 1.1), corresponds to a pedigree relationship. In Section A, two meioses separate two (full) siblings; each meiosis is indicated by a dotted line joining a child and parent in the pedigree diagram. In section B, the most distantly related individuals in the pedigree are a pair of third cousins, in which the two common ancestors are great-great-great-grandparents of the individual on the left and great-grandparents of the individual on the right, respectively. The two third cousins are separated by 8 meioses.

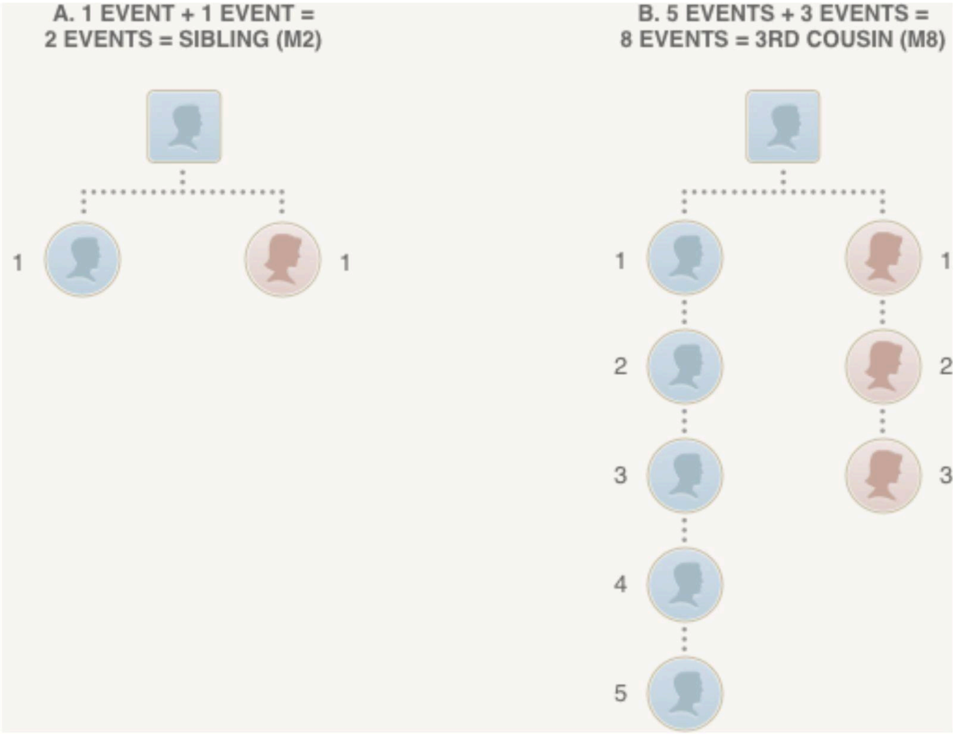


Figure 5.1. *Two examples illustrate the correspondence between pedigree relationship and number of reproductive events (meioses). Reproductive events are indicated by dotted lines between individuals in the pedigree diagram. Note that only one of two parents are shown. A, shows the pedigree for two (full) siblings sharing the same two parents (only one parent is shown). B, shows the pedigree for a more extended family in which the two most distantly related individuals are third cousins.*

Since transmission of DNA from parents to child is inherently a random process (explained in section 1.1), the amount of the genome that is IBD between two siblings can vary. As the number of reproductive events separating two individuals increases, so does the number of random transmissions, leading to greater variation in the proportion of the genome that is inherited from common ancestors. Therefore, we face inherently more uncertainty in estimating more distant relationships. We explore these concepts in greater detail in the next section.

We also note that when two individuals share only one common ancestor (e.g. half siblings), the expected amount of shared DNA is less than when individuals share two common ancestors (e.g. full siblings). To determine the expected amount of DNA shared for a half-relationship, we consider the distribution of shared DNA one meiosis level higher. Therefore, half siblings would share DNA expected with 3 meiosis (M3) relationships rather than 2 (M2).

5.2. Method for estimating relationships

To characterize the relationship between the amount of shared IBD and number of separating meioses, we studied the amount of shared IBD inferred from the genotypes of individuals with known relationships. With the large number of AncestryDNA customers who have consented to share their genetic data for research and development of new and improved algorithms and created pedigrees, we have collected a vast network of annotated relationships across the range of shared cM. Although there is the potential for errors when using user-reported data, the sheer volume of the dataset overwhelms the inaccuracies.

We utilized millions of match pairs with relationships defined by the user-reported pedigree information for this analysis. The relationships ranged from twins to fourth cousins and from 6 to 3500 shared cM (as detected using the IBD analysis described in previous sections). While some match pairs may share more than two common ancestors, only the relationship through the most recent ancestor was used in this analysis. Due to the difficulty of annotating half relationships through user-reported pedigree information, only match pairs with full relationships were used. The meiosis level between two individuals was determined based on the generation of the most recent common ancestor to both individuals in the match pair. Because we utilize more information beyond shared cM to estimate close relationships (twins, parent and child, and full siblings), these relationships are included in the analysis for information only. The details of how we estimate close relationships are described later in this section.

The IBD distribution is summarized in Figure 5.2. As discussed above, we observe that the amount of IBD decreases, on average, for more distant relationships. We also observe much greater overlap toward the bottom of Figure 5.2 when the shared cM is smaller. As a result, given smaller amounts of IBD detected, we are typically more uncertain about the exact relationship that explains the detected IBD. These results include the adjusted IBD length after running the Timber algorithm (described in Section 4). The Timber adjustments explain the dip in the M7 and M8 distributions and the increase in the M10 distribution around 70-90 shared cM.

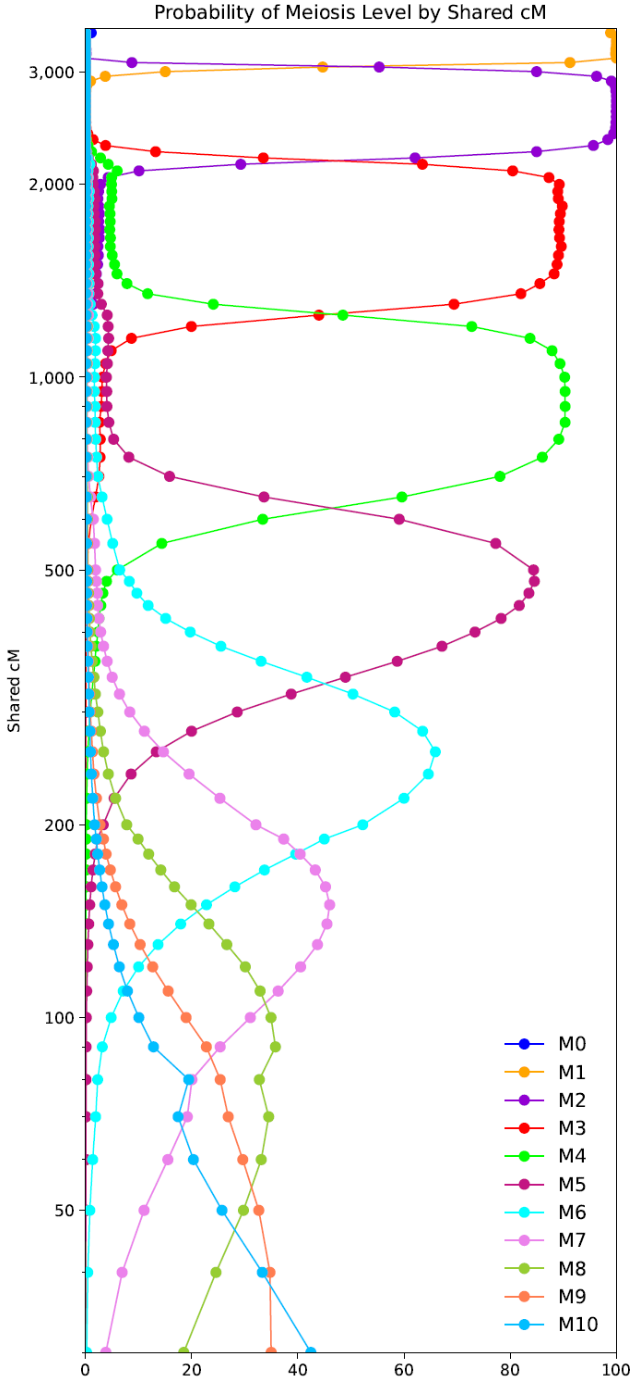


Figure 5.2. *Distribution of total IBD, in cM, detected in pairs corresponding to different annotated relationships, grouped by number of separating meioses. One meiosis (M1) corresponds to parent-child relationships, two meioses (M2) corresponds to (full) siblings, and so on. The distributions for 11 or more separating meioses were not included in this analysis. Note that total IBD lengths—the vertical axis in the plot—are shown on the logarithmic scale, and only IBD greater than 30 cM is shown. The meiosis level is indicated with an M followed by the number of separating meioses..*

The relationship estimation provided to AncestryDNA customers uses the distributions displayed in Figure 5.2. The number of separating meioses that is most likely given the length of IBD detected between a pair of related individuals (assuming they are separated by 10 or fewer meioses) corresponds to the meiosis distribution with the highest probability for the amount of shared cM. For a given number of meioses, the corresponding shared cM interval is extended across the locations on the vertical axis where the corresponding distribution is to the right of the other curves. The meiosis levels with lower probabilities (to the left of the curve with maximum probability) for the shared cM are used to predict other possible relationships for the match pair.

Beyond the intervals illustrated in Figure 5.2, it is also important to consider the uncertainty in a particular relationship estimate. For example, consider the case when two individuals are estimated to share 1,000 cM IBD. According to our empirical data, it is very likely that these two individuals are separated by exactly 4 reproductive events, such as first cousins (see Figure 5.2). Therefore, we could report this relationship estimate with high confidence. On the other hand, consider the case when two individuals share 650 cM IBD. In this situation, we cannot be certain whether the two individuals are separated by 4 or 5 reproductive events; for example, they could be first cousins, or first cousins once removed. This uncertainty is accentuated for more distant relationships and demonstrated by the greater amount of overlap of the corresponding curves in Figure 5.2. We account for greater uncertainty in more distant relationships when delivering estimates to customers by reporting a range of possible relationships (e.g., second to third cousins).

Once we have made a prediction based on estimated IBD, we take an additional step to ensure highly accurate estimates of close relationships—specifically, pairs separated by at most 3 meioses. Although our estimates of close relationships are already expected to be highly accurate based on IBD alone, additional factors not accounted for in our simulations, such as unusually high phasing error, can occasionally contribute to errors in our relationship estimates. Therefore, we take an additional step to closely re-examine matches with a significant amount of sharing in order to detect and correct these errors. If a match is determined to share more than 90cM, we scan the genome without regard to a “seed” of identical phase in order to find any segments that may have been missed, which results in a more accurate estimate of total sharing.

We also measure IBD2, places where a person shares DNA with another in the same part of the genome on both sides of the family–i.e., there are two identical sequences of DNA at a location on the genome instead of just one. IBD2 can occur when people are related on both sides of the family, such as is the case for full siblings (where IBD2 makes up approximately 25% of the genome) and identical twins (where IBD2 makes up 100% of the genome).

When combined with IBD1, the measure described in Section 3, IBD2 improves separation of close pedigree relationships, thereby augmenting our ability to accurately estimate these relationships. Figure 5.3 shows the empirical distribution of two matching statistics–total detected IBD, and an additional statistic that provides an estimate of the proportion of the genome that is IBD2. With total IBD alone (the vertical axis in Figure 5.3), we can determine with near-perfect accuracy whether a pair of individuals are parent-child or full siblings. By contrast, full siblings and half siblings show a great deal of overlap in total IBD shared, so we cannot determine as accurately whether a pair of individuals are full siblings or half siblings. However, when we consider the total IBD and IBD2 statistics jointly, in Figure 5.3 we observe that these data clearly separate parent-child pairs from full siblings, and greatly improve the separation of full siblings and half siblings. Therefore, by using both matching statistics simultaneously, we achieve nearly 100% accuracy in distinguishing close relationships–identical twins, parent-child, full siblings, and half siblings.

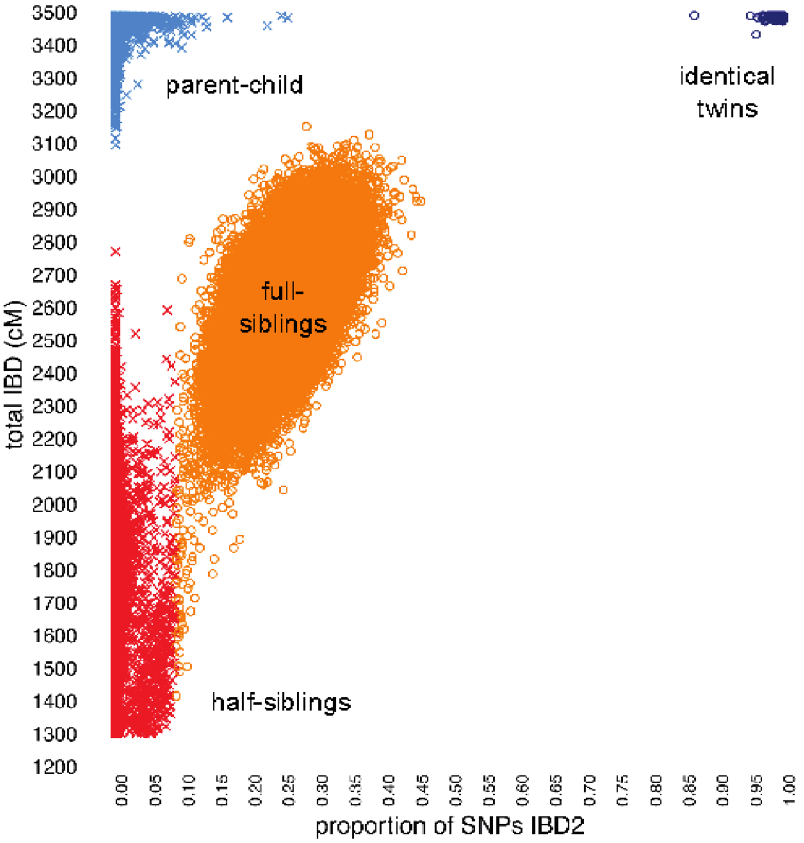


Figure 5.3. *Empirical distribution of two matching statistics in approximately 400,000 pairs (i, j), in which total IBD shared between i and j is greater than 1,300 cM. Each point corresponds to a pair (i, j), and is colored by the final relationship estimate. The vertical axis shows the total detected IBD between i and j, in cM. The horizontal axis shows an additional matching statistic—the proportion of SNPs within 200-SNP segments in which the genotypes at all 200 SNPs are identical in i and j. This additional statistic gives an estimate of the proportion of the genome that is IBD across both haplotypes (IBD2).*

5.3. Estimating Number of Shared Segments

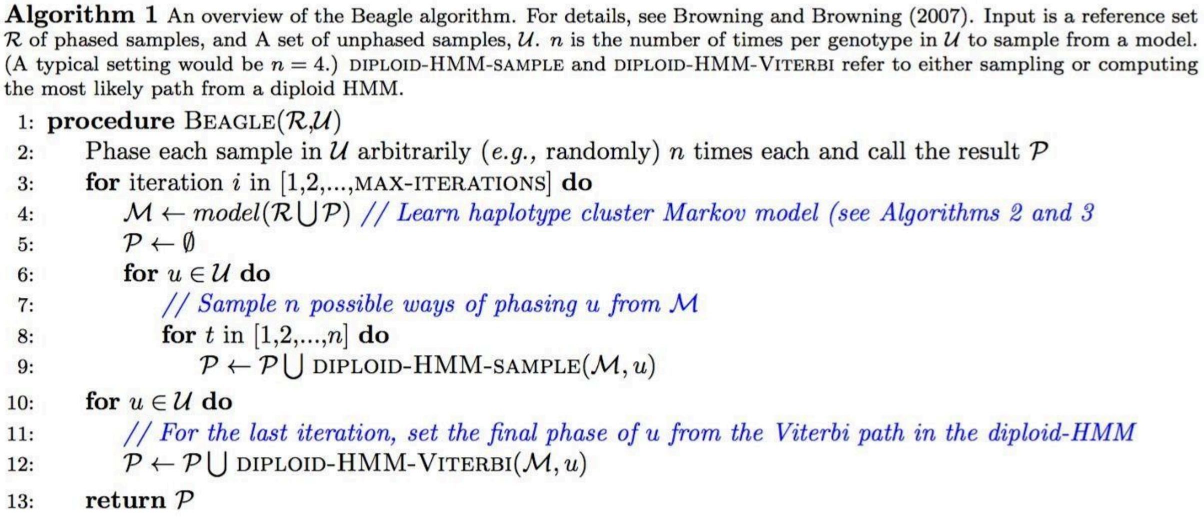
In addition to the length of each DNA match, we also estimate the number of segments that two people share (commonly referred to as the *number of shared segments*). Sometimes, the segments we find are close enough together to make it difficult to determine if the sharing consists of one long segment or multiple shorter ones. Errors in the genotyping process are rare, but do occur many times in an individual's data (out of hundreds of thousands of SNPs analyzed), which could cause a shared segment to "break" in the middle. We therefore post-process each individual match to judge whether groups of shared segments on the same chromosome appear to be broken by a single SNP mismatch that can be explained away as a genotype error. We recombine those segments for the final estimate of the number of shared segments for each match.

6. Summary and future plans

In this technical document, we have given an overview of our algorithms for phasing genotypes, detecting IBD, and estimating relationships in the AncestryDNA database. Our aim in developing these algorithms is to help AncestryDNA customers gain insight into how they are related to other people who have taken the AncestryDNA test. Each relationship estimate delivered to an AncestryDNA customer may yield a genealogical discovery.

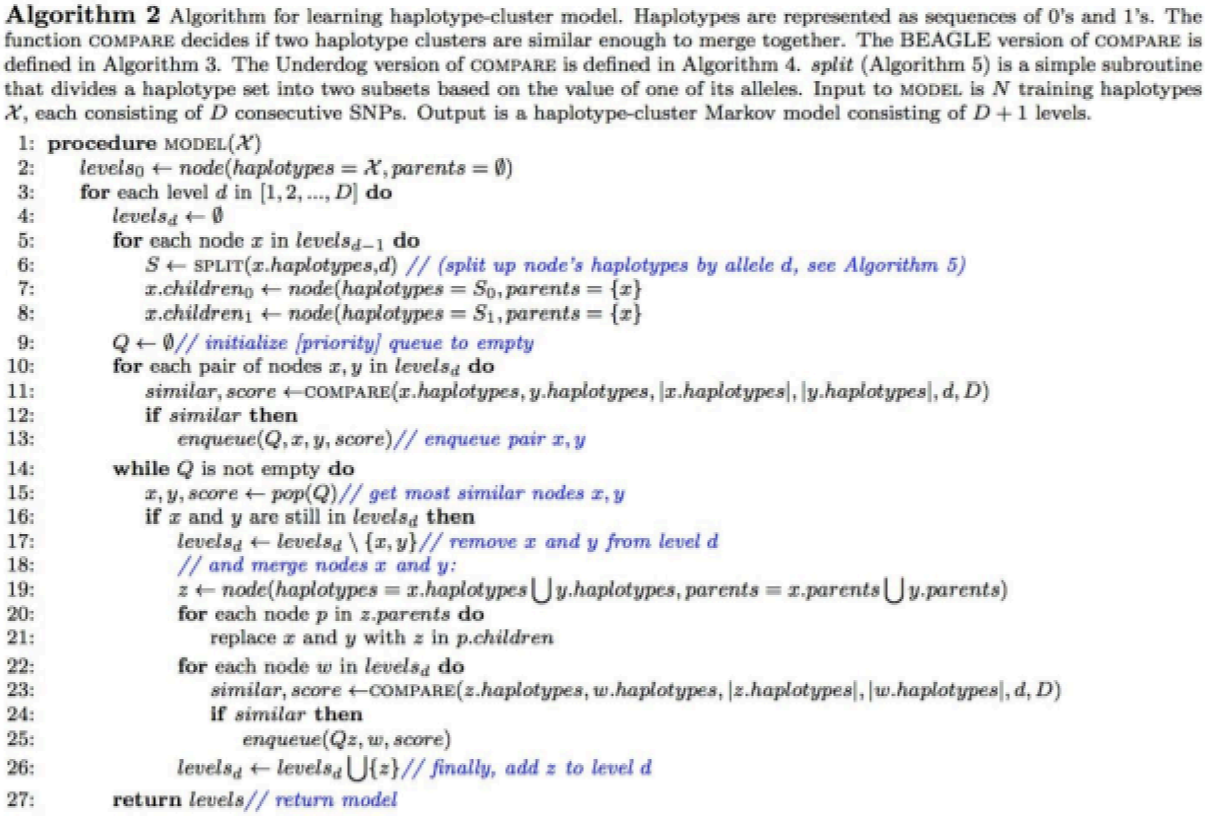
Some of the technical advances we have described here, such as accurate genotype phasing, have been achieved by developing algorithms that can scale to the massive amount of genetic data from our AncestryDNA customers. In addition, several advancements have been made possible by the large number of customers that have consented to share their genetic data for research and development of new and improved algorithms. Therefore, we expect further improvements in DNA matching as the AncestryDNA database grows further.

Appendix A. BEAGLE genotype phasing algorithm pseudocode



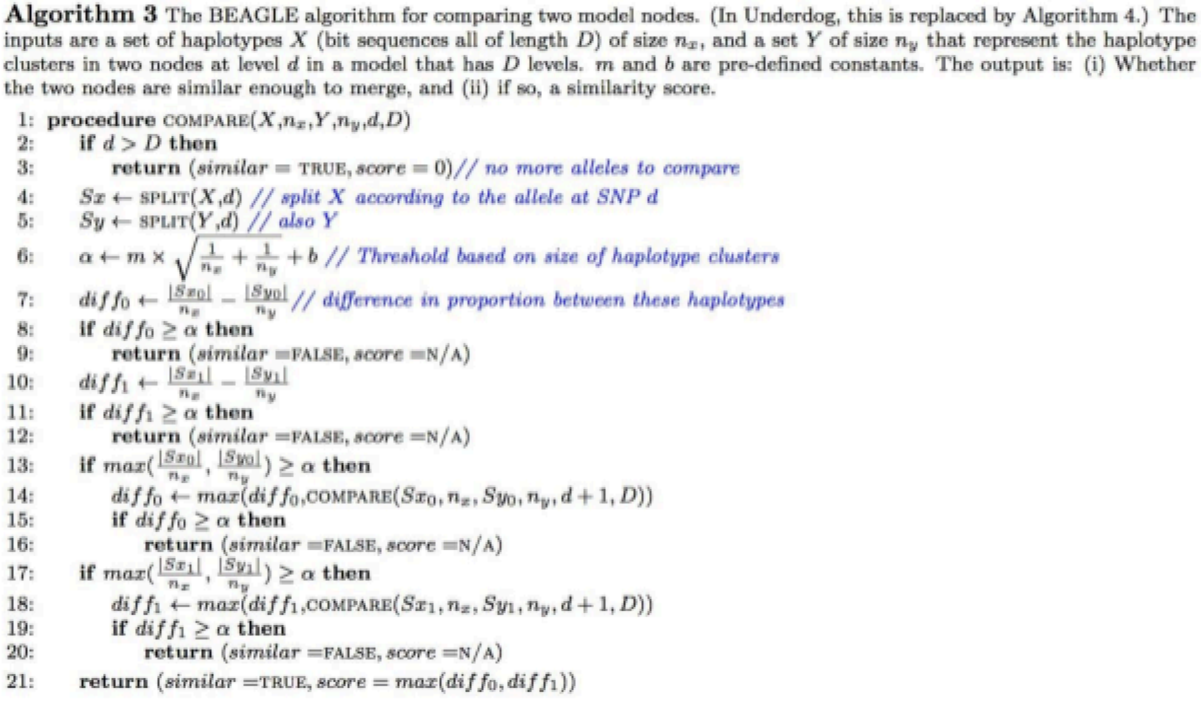
Algorithm 1 An overview of the Beagle algorithm. For details, see Browning and Browning (2007). Input is a reference set $\mathcal{R}$ of phased samples, and A set of unphased samples, $\mathcal{U}$. n is the number of times per genotype in $\mathcal{U}$ to sample from a model. (A typical setting would be $n = 4$.) DIPLOID-HMM-SAMPLE and DIPLOID-HMM-VITERBI refer to either sampling or computing the most likely path from a diploid HMM.

```
procedure BEAGLE($\mathcal{R}$,$\mathcal{U}$)
    Phase each sample in $\mathcal{U}$ arbitrarily ($e.g.$, randomly) $n$ times each and call the result $\mathcal{P}$
    for iteration $i$ in [1,2,...,MAX-ITERATIONS] do
        $\mathcal{M} \leftarrow model(\mathcal{R} \bigcup \mathcal{P})$ // Learn haplotype cluster Markov model (see Algorithms 2 and 3
        $\mathcal{P} \leftarrow \emptyset$
        for $u \in \mathcal{U}$ do
            // Sample $n$ possible ways of phasing $u$ from $\mathcal{M}$
            for $t$ in [1,2,...,$n$] do
                $\mathcal{P} \leftarrow \mathcal{P} \bigcup$ DIPLOID-HMM-SAMPLE($\mathcal{M}, u$)
    for $u \in \mathcal{U}$ do
        // For the last iteration, set the final phase of u from the Viterbi path in the diploid-HMM
        $\mathcal{P} \leftarrow \mathcal{P} \bigcup$ DIPLOID-HMM-VITERBI($\mathcal{M}, u$)
    return $\mathcal{P}$
```



Algorithm 2 Algorithm for learning haplotype-cluster model. Haplotypes are represented as sequences of 0's and 1's. The function COMPARE decides if two haplotype clusters are similar enough to merge together. The BEAGLE version of COMPARE is defined in Algorithm 3. The Underdog version of COMPARE is defined in Algorithm 4. *split* (Algorithm 5) is a simple subroutine that divides a haplotype set into two subsets based on the value of one of its alleles. Input to MODEL is N training haplotypes $\mathcal{X}$, each consisting of D consecutive SNPs. Output is a haplotype-cluster Markov model consisting of $D+1$ levels.

```
procedure MODEL($\mathcal{X}$)
    $levels_0 \leftarrow node(haplotypes = \mathcal{X}, parents = \emptyset)$
    for each level $d$ in $[1, 2, ..., D]$ do
        $levels_d \leftarrow \emptyset$
        for each node $x$ in $levels_{d-1}$ do
            $S \leftarrow$ SPLIT($x.haplotypes$,d) // (split up node's haplotypes by allele d, see Algorithm 5)
            $x.children_0 \leftarrow node(haplotypes = S_0, parents = \{x\}$
            $x.children_1 \leftarrow node(haplotypes = S_1, parents = \{x\}$
        $Q \leftarrow \emptyset$// initialize [priority] queue to empty
        for each pair of nodes $x, y$ in $levels_d$ do
            $similar, score \leftarrow$COMPARE($x.haplotypes, y.haplotypes, |x.haplotypes|, |y.haplotypes|, d, D$)
            if $similar$ then
                $enqueue(Q, x, y, score)$// enqueue pair $x, y$
        while $Q$ is not empty do
            $x, y, score \leftarrow pop(Q)$// get most similar nodes $x, y$
            if $x$ and $y$ are still in $levels_d$ then
                $levels_d \leftarrow levels_d \setminus \{x, y\}$// remove $x$ and $y$ from level $d$
                // and merge nodes $x$ and $y$:
                $z \leftarrow node(haplotypes = x.haplotypes \bigcup y.haplotypes, parents = x.parents \bigcup y.parents)$
                for each node $p$ in $z.parents$ do
                    replace $x$ and $y$ with $z$ in $p.children$
                for each node $w$ in $levels_d$ do
                    $similar, score \leftarrow$COMPARE($z.haplotypes, w.haplotypes, |z.haplotypes|, |w.haplotypes|, d, D$)
                    if $similar$ then
                        $enqueue(Qz, w, score)$
                $levels_d \leftarrow levels_d \bigcup \{z\}$// finally, add $z$ to level $d$
        return $levels$// return model
```



Algorithm 3 The BEAGLE algorithm for comparing two model nodes. (In Underdog, this is replaced by Algorithm 4.) The inputs are a set of haplotypes X (bit sequences all of length D) of size n_x, and a set Y of size n_y that represent the haplotype clusters in two nodes at level d in a model that has D levels. m and b are pre-defined constants. The output is: (i) Whether the two nodes are similar enough to merge, and (ii) if so, a similarity score.

```
procedure COMPARE($X$,$n_x$,$Y$,$n_y$,$d$,$D$)
    if $d > D$ then
        return ($similar$ = TRUE, $score = 0$)// no more alleles to compare
    $Sx \leftarrow$ SPLIT($X$,$d$) // split $X$ according to the allele at SNP $d$
    $Sy \leftarrow$ SPLIT($Y$,$d$) // also $Y$
    $\alpha \leftarrow m \times \sqrt{\frac{1}{n_x} + \frac{1}{n_y}} + b$ // Threshold based on size of haplotype clusters
    $diff_0 \leftarrow \frac{|Sx_0|}{n_x} - \frac{|Sy_0|}{n_y}$// difference in proportion between these haplotypes
    if $diff_0 \geq \alpha$ then
        return ($similar$ =FALSE, $score$ =N/A)
    $diff_1 \leftarrow \frac{|Sx_1|}{n_x} - \frac{|Sy_1|}{n_y}$
    if $diff_1 \geq \alpha$ then
        return ($similar$ =FALSE, $score$ =N/A)
    if $max(\frac{|Sx_0|}{n_x}, \frac{|Sy_0|}{n_y}) \geq \alpha$ then
        $diff_0 \leftarrow max(diff_0$,COMPARE($Sx_0, n_x, Sy_0, n_y, d+1, D$))
        if $diff_0 \geq \alpha$ then
            return ($similar$ =FALSE, $score$ =N/A)
    if $max(\frac{|Sx_1|}{n_x}, \frac{|Sy_1|}{n_y}) \geq \alpha$ then
        $diff_1 \leftarrow max(diff_1$,COMPARE($Sx_1, n_x, Sy_1, n_y, d+1, D$))
        if $diff_1 \geq \alpha$ then
            return ($similar$ =FALSE, $score$ =N/A)
    return ($similar$ =TRUE, $score = max(diff_0, diff_1)$)
```

Appendix B. Underdog genotype phasing algorithm

Our primary aim is to learn haplotype-cluster models from large training sets and use them to phase samples efficiently and accurately. Here we introduce some modifications to BEAGLE so that the algorithm is better suited to this aim. Our new algorithm is called Underdog.

BEAGLE only represents haplotypes that actually appear in the training examples. However, since we would like to phase new genotype samples that do not necessarily appear in the training set, we set the transition probability for allele *a* at a given SNP to

$$P(a) = \frac{1}{2}\gamma + \frac{n_a}{n_a + n_{\bar{a}}}(1-\gamma)$$

(Eq. B1)

where n_a is the number of times allele *a* is observed in training data, and $n_{\bar{a}}$ is the number of times the other allele is observed. This is compared with the BEAGLE formula shown in Algorithm 3. Here, γ is a positive number between 0 and 1. To illustrate the rationale for this choice of transition probability, consider the bottom state of level 2 in Figure 2.1. Instead of having only one transition (to the bottom state in level 3) with 100% probability, we add a second transition for the blue allele (also to the bottom state in level 3) that is visited with probability γ. We define all transition probabilities in the haplotype-cluster model in this way. These transition probabilities are only noticeably different from the transition probabilities in BEAGLE when one allele occurs very infrequently in the training set within a given cluster of haplotypes. With this modification, Underdog allows for genotype phase based on haplotypes that did not appear in the training set.

Although the BEAGLE haplotype-cluster models are intended to be parsimonious, building these models from hundreds of thousands of haplotypes can still yield very large models with millions of states, making it difficult to phase genotype samples in a reasonable amount of time. To address this problem, we first observe that although there is typically a large number of possible ways of phasing a sample, most of these possibilities are extremely unlikely conditioned on a specific haplotype-cluster model. In other words, most of the probability mass is typically concentrated on a small subset of paths through the HMM. To avoid considering all possible paths (which is computationally expensive), at a given level *d* we retain the smallest number of states such that the probability of being in one of those states is greater than 1 - ε. Even for small values of ε, this heuristic dramatically decreases the computational cost of sampling from the HMM, and computing the most likely phase using the Viterbi algorithm (Figure B1), while incurring very few additional phasing errors.

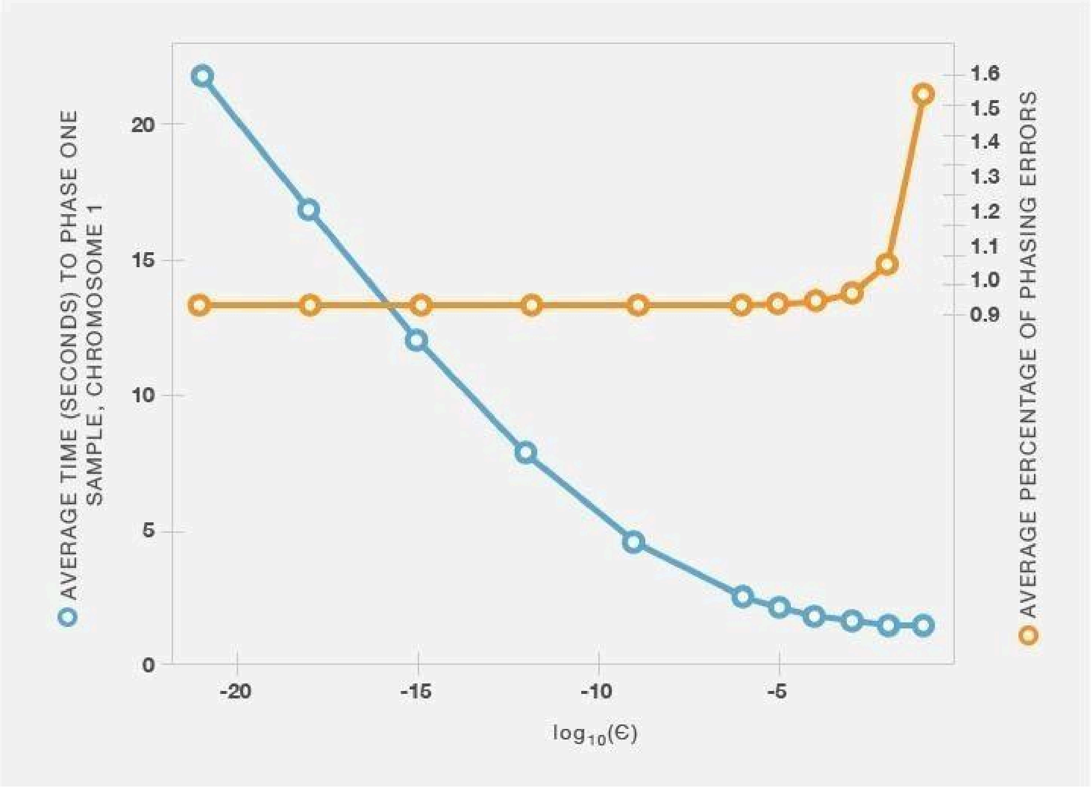


Figure B1: *Relationship between choice of HMM parameter ε and average computation time for phasing a genotype sample (based on chromosome 1 only). If we set ε = 0, the average sample phasing time is 63 seconds, and the average phasing error rate is 0.93%. For choices of ε that are larger, but not too large, we achieve comparable phasing accuracy with a dramatic reduction in computational expense. Note that the computation time here does not include file input/output, nor the time taken to merge the phasing results from multiple windows.*

The second modification we make to BEAGLE concerns the criterion for deciding whether two haplotype clusters (*i.e.*, nodes of the haploid Markov model) should be merged during model learning (see Algorithm 4). Since the standard method is overly confident for frequencies that are close to 0 or 1, we regularize the estimates using a symmetric beta distribution as a prior. Specifically, haplotype clusters *x* and *y* are not merged unless the following condition is satisfied for some haplotype *h*:

$$\frac{(\tilde{p}_x^{(h)} - \tilde{p}_y^{(h)})^2}{\frac{\tilde{p}_x^{(h)}(1-\tilde{p}_x^{(h)})}{n_x} + \frac{\tilde{p}_y^{(h)}(1-\tilde{p}_y^{(h)})}{n_y}} \geq C$$

(Eq. B2)

where n_x and n_y are the sizes of clusters x and y. The posterior allele frequency estimates in this formula are

$$\begin{aligned}\tilde{p}_x^{(h)} &= \frac{n_x(h)+\alpha}{n_x+\alpha+\beta}\\ \tilde{p}_y^{(h)} &= \frac{n_y(h)+\alpha}{n_y+\alpha+\beta}\end{aligned}$$

(Eq. B3)

where $n_x(h)$ and $n_y(h)$ are the numbers of haplotypes that begin with haplotype h. We set the parameters of the Beta prior (the prior counts), α and β, to 0.5. Compare this criterion to the one used in Browning (2006), (also refer to Algorithm 3), which merges two clusters unless the following relation holds for some h :

$$|\hat{p}_x^{(h)} - \hat{p}_y^{(h)}| \geq \sqrt{n_x^{-1} + n_y^{-1}}$$

(Eq. B4)

where $\hat{p}_x^{(h)}$ is the proportion of haplotypes in cluster x with that begin with haplotype h , and $\hat{p}_y^{(h)}$ is the proportion of haplotypes in cluster y that begin with h . We evaluated the phasing accuracy of the algorithm using a few different values for constant C and settled on $C = 20$.

Algorithm 4 is the modified version of BEAGLE's procedure (Algorithm 3) that applies Eq. B2 to merging haplotypes during model building.

For computational efficiency, on each chromosome we estimate the genotype phase within 500-SNP windows separately. This can result in a loss of phasing accuracy at the beginning and end of each window because information outside the window is ignored, and therefore there is less information about the genotypes at the two extremities of the window. To address this problem, we learn haplotype-cluster models in overlapping windows; specifically, we use 500-SNP windows in which two adjacent windows on the same chromosome overlap by 100 SNPs. Since the final phasing estimates produced in the two windows may disagree in the overlapping portion, it is not immediately clear how to combine the phasing estimates from adjacent windows. We propose a simple solution to this problem. First, we select the SNP nearest the midpoint of the overlapping portion at which the genotype is heterozygous (that is, the two allele copies are not the same). We call this the "switch-point SNP." We then join the sequences from the overlapping windows that share the same allele at this switch-point SNP. For example, in Figure B2 we join the top sequence in the left-hand window with the bottom sequence in the right-hand window because they are both estimated to carry the blue allele at the selected switch-point SNP.

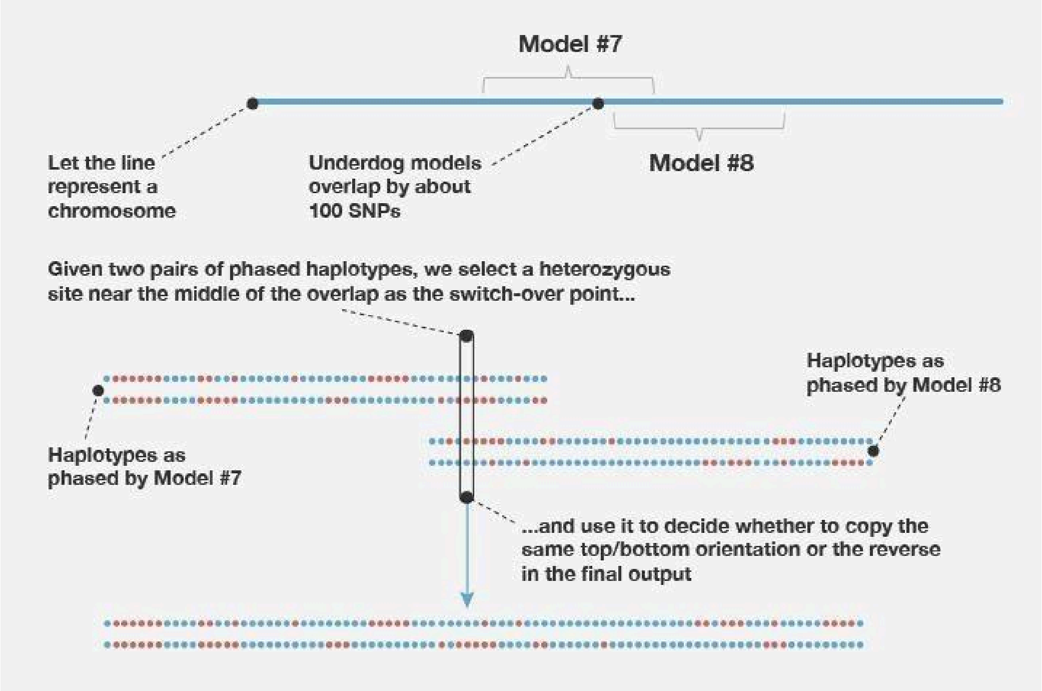
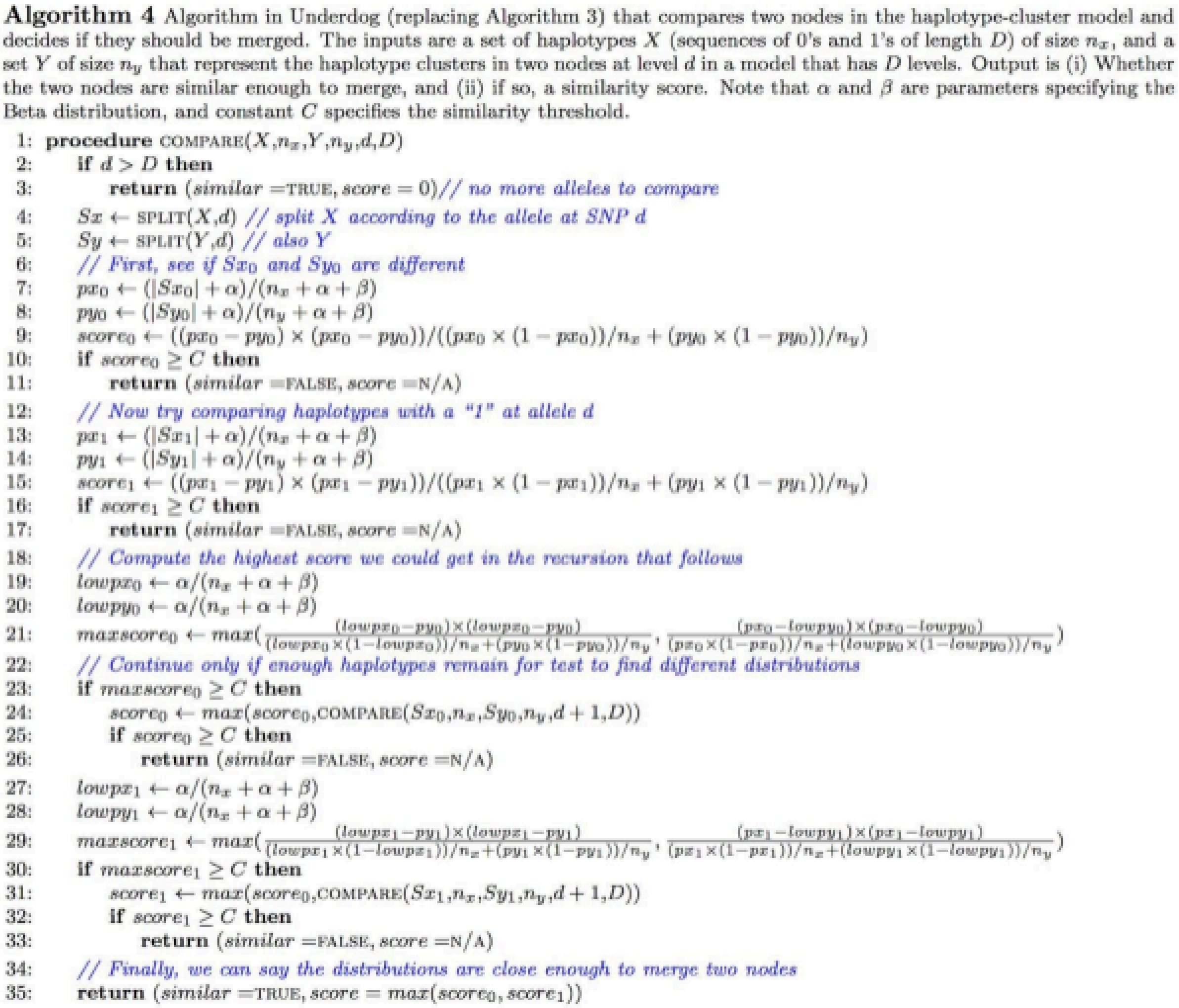
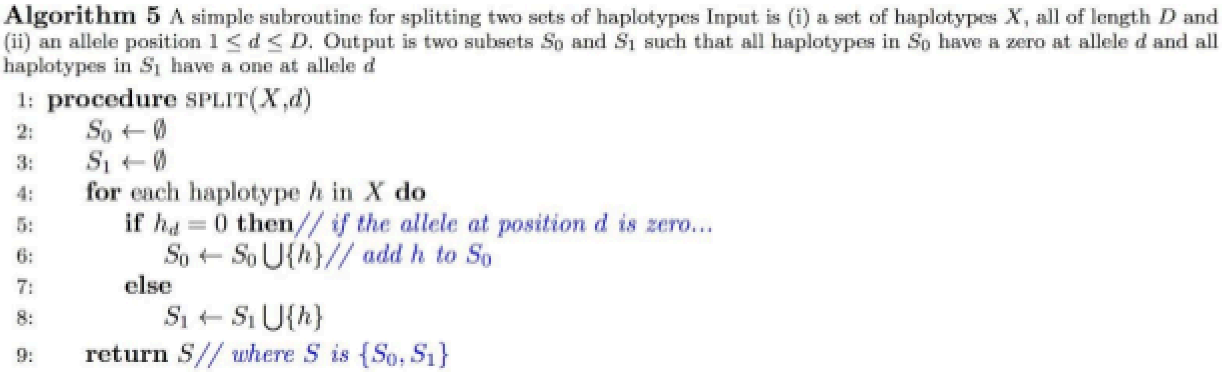

Model #7
Let the line represent a chromosome
Underdog models overlap by about 100 SNPs
Model #8
Given two pairs of phased haplotypes, we select a heterozygous site near the middle of the overlap as the switch-over point...
Haplotypes as phased by Model #8
Haplotypes as phased by Model #7
...and use it to decide whether to copy the same top/bottom orientation or the reverse in the final output

Figure B2: *Underdog learns haplotype-cluster models in overlapping windows. This figure illustrates how we obtain the final genotype phase from these overlapping windows.*



Algorithm 4 Algorithm in Underdog (replacing Algorithm 3) that compares two nodes in the haplotype-cluster model and decides if they should be merged. The inputs are a set of haplotypes X (sequences of 0's and 1's of length D) of size n_x, and a set Y of size n_y that represent the haplotype clusters in two nodes at level d in a model that has D levels. Output is (i) Whether the two nodes are similar enough to merge, and (ii) if so, a similarity score. Note that α and β are parameters specifying the Beta distribution, and constant C specifies the similarity threshold.

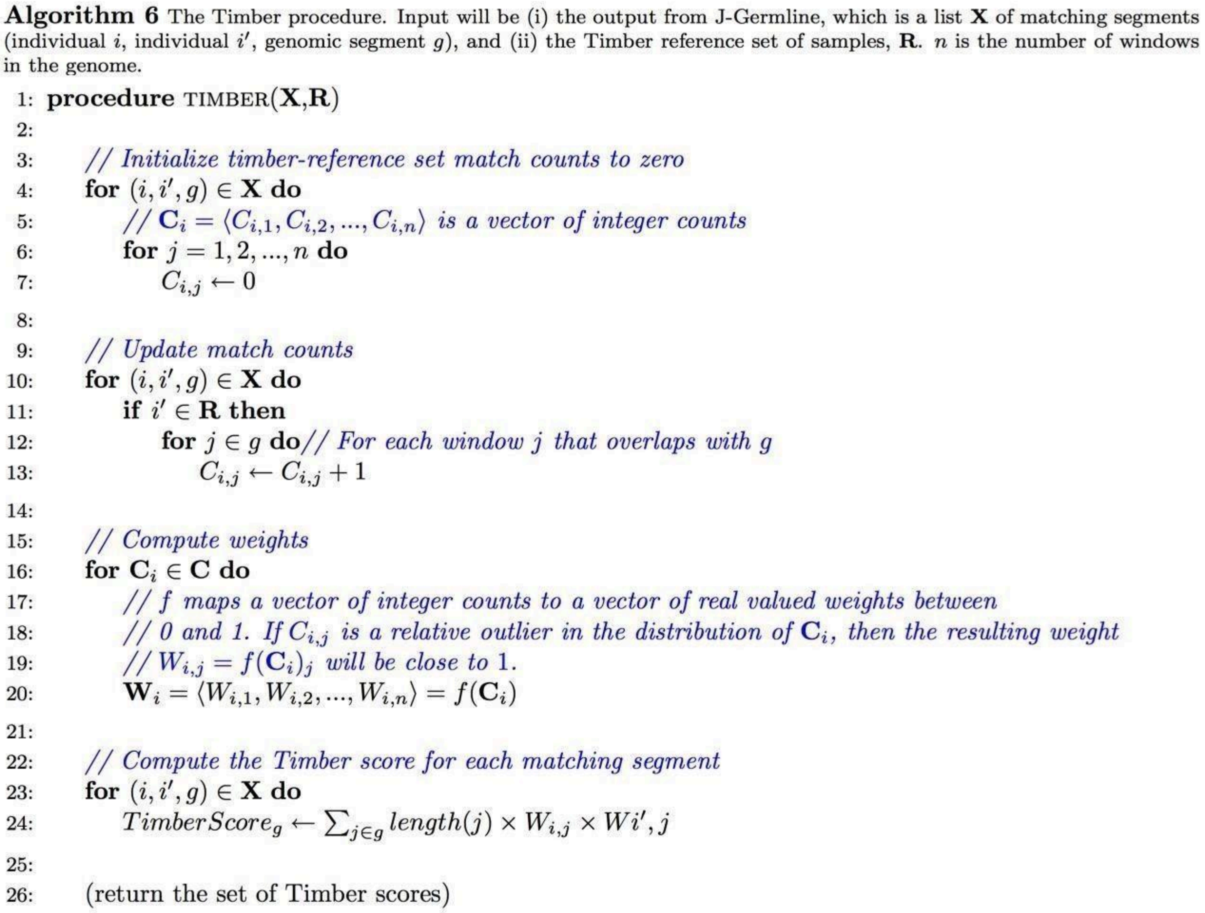
```
procedure COMPARE($X$,$n_x$,$Y$,$n_y$,$d$,$D$)
    if $d > D$ then
        return ($similar$ =TRUE, $score = 0$)// no more alleles to compare
    $Sx \leftarrow$ SPLIT($X$,$d$) // split $X$ according to the allele at SNP $d$
    $Sy \leftarrow$ SPLIT($Y$,$d$) // also $Y$
    // First, see if $Sx_0$ and $Sy_0$ are different
    $px_0 \leftarrow (|Sx_0| + \alpha)/(n_x + \alpha + \beta)$
    $py_0 \leftarrow (|Sy_0| + \alpha)/(n_y + \alpha + \beta)$
    $score_0 \leftarrow ((px_0 - py_0) \times (px_0 - py_0))/((px_0 \times (1 - px_0))/n_x + (py_0 \times (1 - py_0))/n_y)$
    if $score_0 \geq C$ then
        return ($similar$ =FALSE, $score$ =N/A)
    // Now try comparing haplotypes with a "1" at allele $d$
    $px_1 \leftarrow (|Sx_1| + \alpha)/(n_x + \alpha + \beta)$
    $py_1 \leftarrow (|Sy_1| + \alpha)/(n_y + \alpha + \beta)$
    $score_1 \leftarrow ((px_1 - py_1) \times (px_1 - py_1))/((px_1 \times (1 - px_1))/n_x + (py_1 \times (1 - py_1))/n_y)$
    if $score_1 \geq C$ then
        return ($similar$ =FALSE, $score$ =N/A)
    // Compute the highest score we could get in the recursion that follows
    $lowpx_0 \leftarrow \alpha/(n_x + \alpha + \beta)$
    $lowpy_0 \leftarrow \alpha/(n_x + \alpha + \beta)$
    $maxscore_0 \leftarrow max(\frac{(lowpx_0 - py_0)\times(lowpx_0 - py_0)}{(lowpx_0\times(1-lowpx_0))/n_x+(py_0\times(1-py_0))/n_y}, \frac{(px_0 - lowpy_0)\times(px_0 - lowpy_0)}{(px_0\times(1-px_0))/n_x+(lowpy_0\times(1-lowpy_0))/n_y})$
    // Continue only if enough haplotypes remain for test to find different distributions
    if $maxscore_0 \geq C$ then
        $score_0 \leftarrow max(score_0,$COMPARE$(Sx_0,n_x,Sy_0,n_y,d + 1,D))$
        if $score_0 \geq C$ then
            return ($similar$ =FALSE, $score$ =N/A)
    $lowpx_1 \leftarrow \alpha/(n_x + \alpha + \beta)$
    $lowpy_1 \leftarrow \alpha/(n_x + \alpha + \beta)$
    $maxscore_1 \leftarrow max(\frac{(lowpx_1 - py_1)\times(lowpx_1 - py_1)}{(lowpx_1\times(1-lowpx_1))/n_x+(py_1\times(1-py_1))/n_y}, \frac{(px_1 - lowpy_1)\times(px_1 - lowpy_1)}{(px_1\times(1-px_1))/n_x+(lowpy_1\times(1-lowpy_1))/n_y})$
    if $maxscore_1 \geq C$ then
        $score_1 \leftarrow max(score_0,$COMPARE$(Sx_1,n_x,Sy_1,n_y,d + 1,D))$
        if $score_1 \geq C$ then
            return ($similar$ =FALSE, $score$ =N/A)
    // Finally, we can say the distributions are close enough to merge two nodes
    return ($similar$ =TRUE, $score = max(score_0, score_1)$)
```



Algorithm 5 A simple subroutine for splitting two sets of haplotypes Input is (i) a set of haplotypes X, all of length D and (ii) an allele position $1 \leq d \leq D$. Output is two subsets S_0 and S_1 such that all haplotypes in S_0 have a zero at allele d and all haplotypes in S_1 have a one at allele d

```
procedure SPLIT($X$,$d$)
    $S_0 \leftarrow \emptyset$
    $S_1 \leftarrow \emptyset$
    for each haplotype $h$ in $X$ do
        if $h_d = 0$ then// if the allele at position $d$ is zero...
            $S_0 \leftarrow S_0 \bigcup \{h\}$// add $h$ to $S_0$
        else
            $S_1 \leftarrow S_1 \bigcup \{h\}$
    return $S$// where $S$ is $\{S_0, S_1\}$
```

Appendix C. The Timber IBD adjustment algorithm



Algorithm 6 The Timber procedure. Input will be (i) the output from J-Germline, which is a list **X** of matching segments (individual i, individual i', genomic segment g), and (ii) the Timber reference set of samples, **R**. n is the number of windows in the genome.

```
procedure TIMBER($\mathbf{X}$,$\mathbf{R}$)

    // Initialize timber-reference set match counts to zero
    for $(i, i', g) \in \mathbf{X}$ do
        // $\mathbf{C}_i = \langle C_{i,1}, C_{i,2}, ..., C_{i,n} \rangle$ is a vector of integer counts
        for $j = 1, 2, ..., n$ do
            $C_{i,j} \leftarrow 0$

    // Update match counts
    for $(i, i', g) \in \mathbf{X}$ do
        if $i' \in \mathbf{R}$ then
            for $j \in g$ do// For each window $j$ that overlaps with $g$
                $C_{i,j} \leftarrow C_{i,j} + 1$

    // Compute weights
    for $\mathbf{C}_i \in \mathbf{C}$ do
        // $f$ maps a vector of integer counts to a vector of real valued weights between
        // 0 and 1. If $C_{i,j}$ is a relative outlier in the distribution of $\mathbf{C}_i$, then the resulting weight
        // $W_{i,j} = f(\mathbf{C}_i)_j$ will be close to 1.
        $\mathbf{W}_i = \langle W_{i,1}, W_{i,2}, ..., W_{i,n} \rangle = f(\mathbf{C}_i)$

    // Compute the Timber score for each matching segment
    for $(i, i', g) \in \mathbf{X}$ do
        $TimberScore_g \leftarrow \sum_{j \in g} length(j) \times W_{i,j} \times Wi', j$

    (return the set of Timber scores)
```

References

- Albrechtsen, I. Moltke, R. Nielsen (2010). Natural selection and the distribution of identity-by-descent in the human genome. *Genetics* 186, 295–308.
- S. R. Browning (2006). Multilocus associate mapping using variable-length Markov chains. *American Journal of Human Genetics* 78, 903–913.
- S. R. Browning, B. L. Browning (2007). Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *American Journal of Human Genetics* 81, 1084–1096.
- B. L. Browning, S. R. Browning (2013). Improving the accuracy and efficiency of identity-by-descent detection in population data. *Genetics* 194, 459–471.
- J. Dean, S. Ghemawat (2008). Mapreduce: simplified data processing on large clusters. *Communications of the ACM* 51, 107–113.
- E. Y. Durand, N. Eriksson, C. Y. Mclean (2014). Reducing pervasive false-positive identical-by-descent segments detected by large-scale pedigree analysis. *Molecular Biology and Evolution* 31, 2212–2222.

- Gusev, J. K. Lowe, M. Stoffel, M. J. Daly, D. Altshuler, J. L. Breslow, J. M. Friedman, I. Pe'er (2009). Whole population, genome-wide mapping of hidden relatedness. *Genome Research* 19, 318–326.
- J. A. Nelder, R. Mead (1965). A simplex algorithm for function minimization. *Computer Journal* 7, 308–313.
- K. Noto, Y. Wang, R. Curtis, J. Granka, M. Barber, J. Byrnes, N. Myres, P. Carbonetto, A. Kermany, C. Han, C. A. Ball, K. G. Chahine (2014). Underdog: a fully-supervised phasing algorithm that learns from hundreds of thousands of samples and phases in minutes. Invited Talk, *64th Annual Meeting of the American Society of Human Genetics*.
- K. Noto, L. Ruiz (2022). Accurate genome-wide phasing from IBD data. *BMC Bioinformatics* 23, 502.
- S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. R. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. W. De Bakker, M. J. Daly, P. C. Sham. PLINK: A tool set for whole-genome association and population-based linkage analyses (2007). *American Journal of Human Genetics* 81, 559–575.
- L. R. Rabiner (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286.
- J. M. Rodriguez, S. Bercovici, L. Huang, R. Frostig, S. Batzoglou (2015). Parente2: a fast and accurate method for detecting identity by descent. *Genome Research* 25, 280–289.
- Ron, Y. Singer, N. Tishby (1998). On the learnability and usage of acyclic probabilistic finite automata. *Journal of Computer and System Sciences* 56, 133–152.
- L. Williams, N. Patterson, J. Glessner, H. Hakonarson, D. Reich (2012). Phasing of many thousands of genotyped samples. *American Journal of Human Genetics* 91, 238–251.