Imputation of Missing Genotypes from Sparse to High Density Using Long-Range Phasing

Imputation of Missing Genotypes From Sparse to High Density Using Long-Range Phasing

Hans D. Daetwyler,*,†,‡,[1] George R. Wiggans,[§] Ben J. Hayes,[§] John A. Woolliams,[†] and Mike E. Goddard*,**

*Biosciences Research Division, Department of Primary Industries, Bundoora 3083, Australia, [†]The Roslin Institute and R(D)SVS, The University of Edinburgh, Roslin EH25 9RG, United Kingdom, [‡]Animal Breeding and Genomics Centre, Wageningen University, 6700 AH Wageningen, The Netherlands, [§]Animal Improvement Programs Laboratory, Agricultural Research Service, U.S. Department of Agriculture, Beltsville, Maryland 20705-2350, and **Faculty of Land and Environment, University of Melbourne, Parkville 3010, Australia

ABSTRACT Related individuals share potentially long chromosome segments that trace to a common ancestor. We describe a phasing algorithm (ChromoPhase) that utilizes this characteristic of finite populations to phase large sections of a chromosome. In addition to phasing, our method imputes missing genotypes in individuals genotyped at lower marker density when more densely genotyped relatives are available. ChromoPhase uses a pedigree to collect an individual's (the proband) surrogate parents and offspring and uses genotypic similarity to identify its genomic surrogates. The algorithm then cycles through the relatives and genomic surrogates one at a time to find shared chromosome segments. Once a segment has been identified, any missing information in the proband is filled in with information from the relative. We tested ChromoPhase in a simulated population consisting of 400 individuals at a marker density of 1500/M, which is approximately equivalent to a 50K bovine single nucleotide polymorphism chip. In simulated data, 99.9% loci were correctly phased and, when imputing from 100 to 1500 markers, more than 87% of missing genotypes were correctly imputed. Performance increased when the number of generations available in the pedigree increased, but was reduced when the sparse genotype contained fewer loci. However, in simulated data, ChromoPhase correctly imputed at least 12% more genotypes than fastPHASE, depending on sparse marker density. We also tested the algorithm in a real Holstein cattle data set to impute 50K genotypes in animals with a sparse 3K genotype. In these data 92% of genotypes were correctly imputed in animals with a genotyped sire. We evaluated the accuracy of genomic predictions with the dense, sparse, and imputed simulated data sets and show that the reduction in genomic evaluation accuracy is modest even with imperfectly imputed genotype data. Our results demonstrate that imputation of missing genotypes, and potentially full genome sequence, using long-range phasing is feasible.

SINGLE nucleotide polymorphism (SNP) arrays from sparse to high density are now available in many species. The genotypes resulting from high-throughput methods are unphased and, therefore, the paternal or maternal source of each allele is unknown. Knowledge of parental origin or haplotype information can be useful in the analysis of complex traits, such as quantitative trait loci (QTL) detection (*e.g.*, Meuwissen and Goddard 2000), genomic selection (*e.g.*, Meuwissen *et al.* 2001; Calus *et al.* 2008; Villumsen and Janss 2009), and detection of imprinting (*e.g.*, Reik and Walter 2001; Wood and Oakey 2006).

Many methods for resolving haplotypes have been proposed and they fall into two broad categories: those that use known relationships between individuals to perform a linkage analysis (*e.g.*, Elston and Stewart 1971; Lander and Green 1987; Weeks *et al.* 1995; Windig and Meuwissen 2004) and those that rely on linkage disequilibrium among the SNP in a population without known relationships (*e.g.*, Clark 1990; Scheet and Stephens 2006; Tier 2006; Browning and Browning 2009).

More recently, another feature of finite population genomics together with the availability of denser marker maps have fostered new phasing approaches, as demonstrated by Kong *et al.* (2008). Population characteristics such as geographical proximity can result in a high probability that

doi: 10.1534/genetics.111.128082
Manuscript received April 5, 2011; accepted for publication June 9, 2011
Supporting information is available online at http://www.genetics.org/content/suppl/2011/06/24/genetics.111.128082.DC1.
[1]Corresponding author: Biosciences Research Division, Department of Primary Industries, 1 Park Dr., Bundoora, Victoria 3083, Australia. E-mail: hans.daetwyler@dpi.vic.gov.au

individuals within a given population share a common ancestor not many generations ago. Similarly, in commercial animal populations, selective breeding has reduced effective population sizes by limiting the number of parents, again causing individuals to share one or more common ancestors in the past few generations. If individuals share a common ancestor n generations ago, they are likely to have shared chromosome segments of average length approximately $1/n$ M. Provided that n is not too large and with dense genotyping of markers, these segments will contain many markers and so it should be possible to recognize them and distinguish them from short segments that are identical-by-state (IBS) but do not trace to the common ancestor, without complex likelihood calculations. This leads to a phasing approach based on the key observation of Kong *et al.* (2008) that if animals have nonconflicting homozygote genotypes over a long string of consecutive loci, they have at least one long haplotype in common. The requirement of a long string of loci leads to a high probability that the common long haplotype has originated in a common ancestor.

Kong *et al.* (2008) called their method long-range phasing, but the principle of comparing long stretches of chromosomes between individuals to identify common segments can also be used to impute and phase missing genotypes or even to impute genotypes on individuals that have not been genotyped at all. One particularly useful application is to impute dense genotypes on individuals with sparse genotypes using dense genotype information on their relatives. Then, for example, genomic predictions for selection in livestock or crop species could be made on the imputed genotypes, at the cost of genotyping the low-density markers. In the extreme, full genome sequences could be imputed for individuals that have been genotyped at moderate density, provided they had enough relatives that had been fully sequenced (Goddard 2008).

Here we describe a computationally efficient algorithm (ChromoPhase) that can phase whole chromosomes and simultaneously impute missing genotypes if a haplotype has been observed in the densely genotyped population. We use a similar approach to that of Kong *et al.* (2008) and its extension (Hickey *et al.* 2011), but whereas their focus is on genotypes, we use haplotypes more explicitly. We also use the pedigree to identify whether a relative is likely to share a part of an individual's paternal or maternal chromosome in addition to animals that are genomically similar, which we call genomic surrogates. In addition, while Kong *et al.* (2008) requires operational extensions to impute missing genotype information, our algorithm already addresses the objective of long-range phasing to both phase and impute loci simultaneously.

Methods

ChromoPhase has the objective of inferring paternal and maternal gametes for a set of individuals on the basis of a subset (possibly a complete subset) of all individuals with dense genotypes. It relies upon the same principle as Kong *et al.* (2008) in that it makes use of the potentially long chromosome segments that related animals share. These segments are particularly long when individuals are closely related, as during meiosis the expected number of crossovers is one per Morgan of chromosome. Therefore, with dense marker genotypes, the phase can be established by comparing an individual to close relatives. An edited pseudo-code of the algorithm is provided in Supporting Information, File S1 and File S2.

We assumed biallelic loci with a reference allele coded 0 and an alternative allele coded 2. Genotypes were coded 0, 1, and 2, corresponding to 00, 02, and 22, respectively. Missing alleles and genotypes are assigned 5. At the start of ChromoPhase, all alleles on the paternal and maternal gametes for all individuals are set to 5. We assume that the individuals in the data can be divided into two groups, set D containing all the individuals with dense genotyping, and set S containing all the individuals that are sparsely genotyped. Our dense simulated data had 15 SNP/cM but, in general, "dense" can be defined as being sufficient for the risk of double crossovers between adjacent loci to be negligible. The complete algorithm has different stages and an overview of the progression through the stages is shown in Figure 1. In stage 1, potential sources of shared chromosome segments are identified using pedigree and genotypic similarity. In stage 2 alleles are assigned using two different approaches: 2A employs rule-based allele assignment based upon genotypes of parents, immediate offspring, and mates and in 2B assignments are based upon phasing using unbroken strings of matching alleles on their respective chromosomes. Stage 2 is iterated (A, B, A ... B) and initially involves only individuals in set D. Once the predefined maximum phasing iteration for set D is reached, stage 3 is carried out involving a further predefined number of iterations of 2A and 2B for all individuals in set S. The imputed paternal and maternal gametes of the densely genotyped individuals (*i.e.*, set D) remain unchanged by the imputation of individuals in S, and if only phasing of densely genotyped individuals is required then stage 3 is unnecessary.

Stage 1: Information sources

Each individual in sets D and S is considered as a proband. Molecular genotyping errors are checked at each locus by identifying where the proband genotype is inconsistent with the paternal (maternal) genotype; *e.g.*, proband genotype is 0 and father's genotype is 2. Inconsistent genotypes are set to missing in the proband if it is the only progeny conflicting with the parent. A parent's genotype is set to missing if it conflicts with most offspring. Three sets of densely genotyped relatives are then defined for each proband by tracing through the full pedigree. Hence, completely ungenotyped individuals are used to connect genotyped individuals. The first set of relatives consists of all descendants of the proband and these are collected starting with the youngest

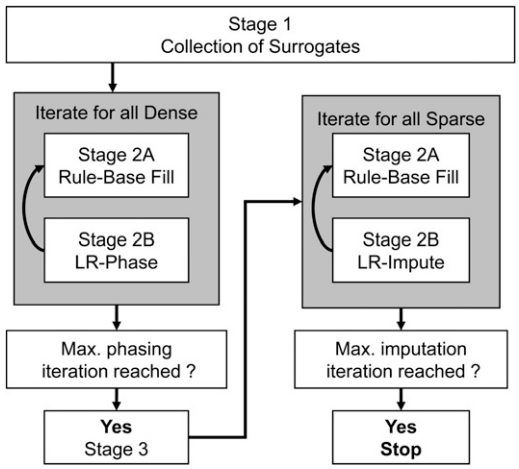

Figure 1 Representation of the three stages and workflow in the long-range (LR) phasing and imputation algorithm, where Dense and Sparse refer to densely and sparsely genotyped individuals.

individual. The second set, starting with the oldest individual, is called surrogate fathers and consists of individuals related to the proband through his or her father. If the father is densely genotyped, then he is the only surrogate father, because more distant relatives do not add further information on the paternal side of the proband. In a proband with an ungenotyped father, the set of surrogate fathers of the proband consists of the father's surrogate fathers and mothers as well as their offspring. Analogous rules are applied to define the surrogate mothers. Only relatives up to 3 degrees removed from the proband are used in the sets of surrogates.

Stage 2A: Single locus, rule-based allele assignment

ChromoPhase applies rule-based allele assignment to the paternal or maternal gamete if they can be unambiguously resolved on the basis of an individual's own known genotype, the parental alleles, and offspring alleles (*e.g.*, Pong-Wong *et al.* 2001; Baruch *et al.* 2006). In all invocations the following rules are applied starting at the top of the pedigree (*i.e.*, the oldest individual) working through to the bottom of the pedigree. If the proband genotype is homozygous, then both its paternal and maternal alleles are equal to the homozygous allele. If the father's genotype is homozygous then the proband paternal allele is equal to the father's homozygous allele. If a proband genotype is missing and only the proband's paternal allele is known, then if an offspring has only one allele known and it is opposite to the proband's paternal allele, the proband's maternal allele must be the same as the offspring's maternal allele.

Stage 2B: Phasing of densely genotyped individuals

At the start of each invocation of 2B, genomic surrogates are identified for each proband, by comparing its genotypes with those for individuals in set D. A genomic surrogate must have nonconflicting homozygotes (for example, no instance of 0 genotype in proband and 2 genotype in the potential genomic surrogate) with the proband for at least 100 consecutive loci if applied to an individual in set D, but 400 consecutive loci if applied to an individual in set S. This is necessary because a proband in set S may have a lot of missing information and the stretch of matches must include a sufficient number of sparsely genotyped loci to effectively identify genomic surrogates. Missing genotypes are not included in the 100 loci in set D, but are included in the 400 loci in set S. Genomic surrogates are chosen to limit the population-based search for shared segments to individuals that "have something to offer" to the proband, thereby reducing the total number of comparisons. In all invocations the following procedure is then applied, starting with the oldest individual working through to the youngest. Each proband is compared to each of its relatives, contained within the sets of surrogate fathers and mothers as well as offspring, to allocate alleles by identifying shared segments for three iterations. In the last two phasing iterations, comparisons to relatives continue but now the proband is also compared to its genomic surrogates to identify shared chromosome segments and fill in missing alleles in the proband if the allele can be found in a shared segment. It is possible that even densely genotyped individuals may have some missing genotypes and these are imputed in the course of 2B.

Stage 3: Phasing and imputation of sparsely genotyped individuals

In stage 3, the iterations of stage 2 are now applied to individuals in S, where rule-based filling is applied (stage 2A) and the proband is compared to its relatives and genomic surrogates for the remaining eight iterations (stage 2B).

We now describe the comparisons of probands with relatives and genomic surrogates as well as the criteria used to accept segments as being shared in stage 2B. These routines are the same for phasing or imputation iterations, except for some shared segment acceptance criteria, detailed below. Probands are separated into two main groups, those that have at least one genotyped parent (nonfounders) and those that do not (founders). Founders may have surrogate fathers or mothers if connected to genotyped relatives through pedigree. In nonfounders, phasing and imputation is simpler because rule-based methods can be used (stage 2A). In founders, there is a need to identify erroneous crossovers within shared segments with surrogates so that proband gametes can be flipped after the crossover, if necessary, to minimize the number of crossovers and, in turn, phase the proband.

Criteria for acceptance of shared chromosome segments

The longer the stretch of matching loci between two individuals, the higher the probability that the segment traces to a common ancestor (Kong *et al.* 2008). We required a minimum length of 100 consecutive matching loci for all individuals to accept a segment as shared. In the last

phasing and imputation iteration we relaxed this to 50 loci to allow the filling in of more alleles. These criteria for defining the minimum length of a shared segment can be adapted to suit a data set and will depend on the number of markers per Morgan. Note that the minimum segment length is different from the length of nonopposing homozygotes required to select genomic surrogates. When selecting genomic surrogates, the main concern is identifying a subset of D large enough to provide enough power to phase or impute, but small enough to reduce the number of redundant comparisons and, in turn, reduce the computational burden. In contrast, here the focus is on accepting specific chromosome segments as shared, and we compare gametes instead of genotypes. Also note that while there is a theoretical basis for the length of a genome segment required to be accepted as shared, the remainder of the algorithm's thresholds and parameters have been chosen heuristically.

A segment starts when the alleles match and ends when the alleles do not match anymore, as demonstrated in Figure 2. Missing genotypes or alleles do not end a segment, although the proportion of missing alleles is tracked. When there is a moderate amount of missing information, we may be reasonably confident that the segment is shared if it is long, but it is difficult to define the ends. Hence, it seems useful to still make use of these segments but penalize their length. Therefore, the algorithm removes a short stretch of loci at the beginning and end of an identified segment, termed "offset" in Figure 2. Testing has shown that having an offset of 20 loci results in fewer errors while not impairing phasing or imputation performance significantly. Furthermore, in both phasing and imputation iterations the offset is multiplied by four (chosen through empirical testing) if the proportion of missing information exceeds half the proportion of loci to be imputed. Short segments with a lot of missing information will be discarded by this quadrupled offset. In the last phasing and imputation iteration the offset is reduced to zero to fill in as many loci as possible. When there is a large amount of missing information within a segment, it will be accepted as shared if it meets the following thresholds: (i) segments in nonfounders in set D require $<$20% missing loci, but no threshold is imposed for set S and (ii) segments of founders in either set (D or S) require less than 50% missing loci. In general, the sparser the genotypes in set S, the more relaxed the missing allele thresholds need to be. At the same time, relaxing the thresholds will increase errors.

Comparisons to surrogates

Nonfounders are compared with their surrogates one gamete at a time and Figure 3 shows the individuals steps in this process. A proband's paternal gamete is compared to both of its surrogate father's gametes consecutively (Figure 2). Similarly, a proband's maternal gamete is compared to its surrogate mother's gametes and both proband gametes are compared to both gametes of descendants, and then genomic surrogates, one at a time. When a segment meets the

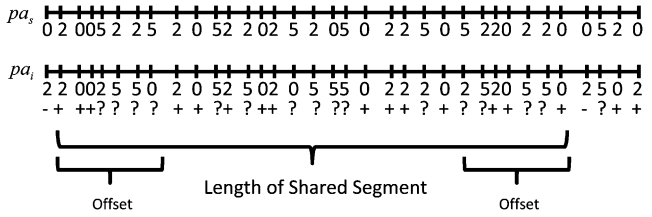

Figure 2 Comparison of proband with a genotyped parent (nonfounder) to its surrogate father. Alleles on the proband's paternal gamete (pa_i) are compared to alleles on surrogate's paternal gamete (pa_s) to identify a shared chromosome segment, where ? is a possible match, + is a match, and – is a nonmatch. The shared segment starts with the first match and ends at the last match. Offset signifies the beginning and end of a matching segment that is removed to guard against errors due to missing alleles (5) before using the segment to fill in information.

acceptance criteria, we collect allele information within the segment as a count of allele 2 and as a count of the total nonmissing alleles in all surrogates that share a segment with the proband at a particular locus. Once comparisons to surrogates have concluded for a particular proband, its missing alleles are filled in on the basis of the collective information from all surrogates. For any given locus, allele 2 is assigned to the proband gamete if the ratio of allele 2 counts over the total number of counts from all contributing surrogates exceeds 0.7, assigned the 0 allele if less than 0.3, and remains as 5 otherwise. The values of 0.7/0.3 can be changed to suit a particular data set. Filling in on the basis of majority information is expected to reduce errors as the information from all surrogates is used and not just the first surrogate that matches. Nevertheless, in the last imputation iteration a slightly modified version of this routine is run where filling in is not based on majority information but is filled in as soon as a segment is accepted as being potentially shared.

Phasing of founders, which is shown in Figure 4, is difficult because rule-based methods based on genotyped parents cannot be used to distinguish paternal or maternal gametes. We define founders as individuals without genotyped parents. Our algorithm makes use of surrogate fathers and mothers to partially distinguish the parental gametes of the founder and then minimizes the number of crossovers for phasing. Phasing and imputation of founders is accomplished by simultaneously comparing both their gametes to one gamete of a surrogate. If the surrogate is a surrogate father then they are compared to the proband's paternal gamete and to the maternal gamete if comparing to a surrogate mother. If the surrogate is a descendent, a descendent of a surrogate father or mother, or a genomic surrogate, then both proband gametes are compared to both surrogate gametes consecutively. This comparison is shown in Figure 5 and the process is the same as that for nonfounders with regard to finding shared segments. Strings of consecutive and matching loci are sought, while keeping track of which proband gamete matched the surrogate. Once the current proband gamete ceases to match, either because of a switch

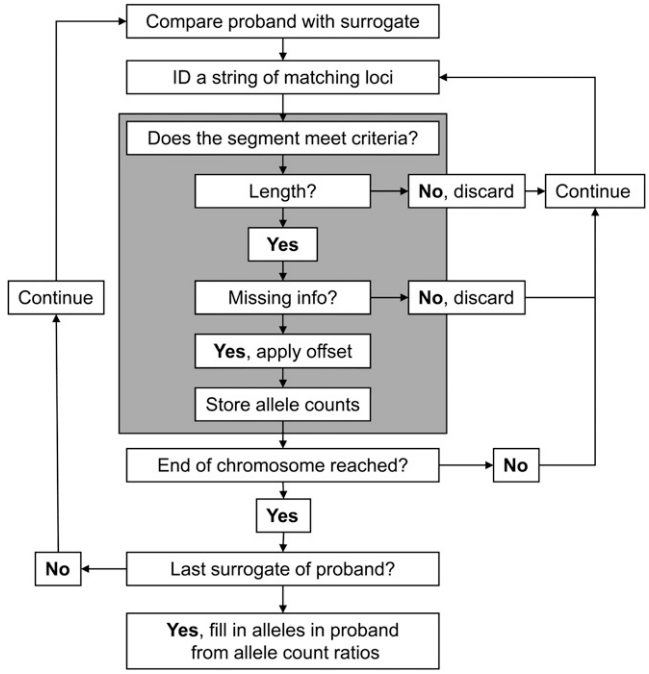

Figure 3 Representation of comparing nonfounders (individuals with at least one genotyped parent) to their surrogates (densely genotyped relative or genomic surrogate).

to matching with the other proband gamete or because neither proband gamete matches the surrogate, then the segment is subject to the acceptance criteria above and, if accepted, missing alleles in the proband gamete are filled with allele information from the surrogate gamete.

The main rationale behind comparing both gametes of a proband founder simultaneously to the surrogate gamete is that it allows the algorithm to collect information on what appear to be crossovers. As an example, in Figure 5 the surrogate gamete first matches the proband's paternal gamete; it then switches to matching the proband's maternal gamete. The locus where the matching switches from paternal to maternal proband gamete either could be regarded as a crossover in the surrogate's gamete or the proband's paternal and maternal gametes need to be flipped from that locus onward (*i.e.*, erroneous crossover in proband). All such crossover points are stored during the assessment of each proband. Before moving to the next proband, in loci where the number of crossovers among surrogates outnumbers the number of noncrossovers, the paternal and maternal gametes of the proband founder are flipped to remove the crossover. This process of minimizing crossovers is required to be repeated as flipping one segment in the proband can reveal new crossover points (Figure 4). In most cases, 5 "minimize crossover" iterations are sufficient for the number of crossovers to converge, but we allow up to 20. While minimizing crossovers for founders is repeated in each invocation of 2B, only few adjustments are required in later invocations as new information becomes available.

Simulated populations for testing

Populations in mutation drift equilibrium were simulated by randomly mating individuals for 1000 generations with crossover and mutation. Effective population size (N_e) was 200 and the number of male and female parents was equal across generations. Previous work established that with this N_e mutation, drift equilibrium was achieved with 1000 generations. One male and one female offspring were produced per mating. Pedigree and genotype information was retained for individuals in the last five generations. In

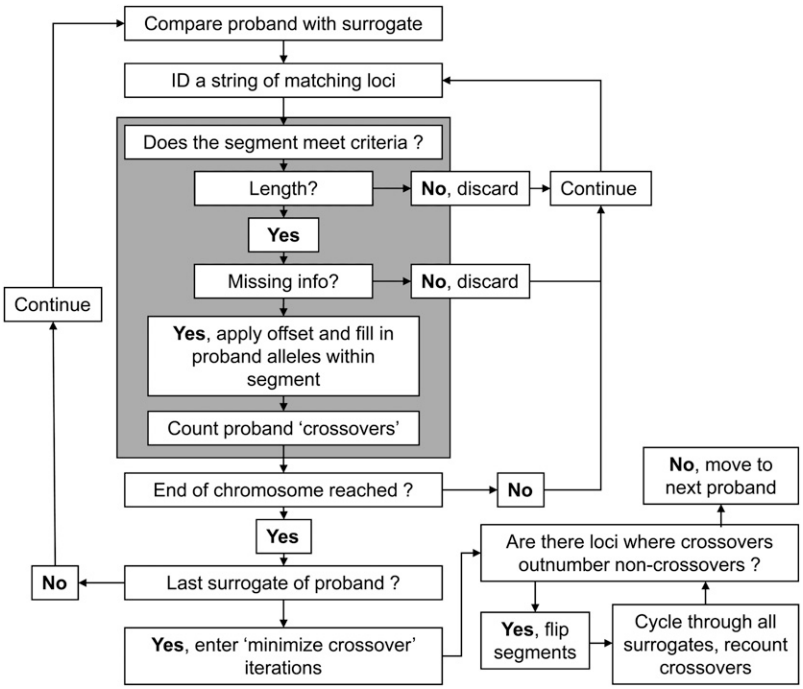

Figure 4 Representation of comparing founders (individuals with no genotyped parents) to their surrogates (densely genotyped relative or genomic surrogate), including iterations in which the number of crossovers are minimized in the proband.

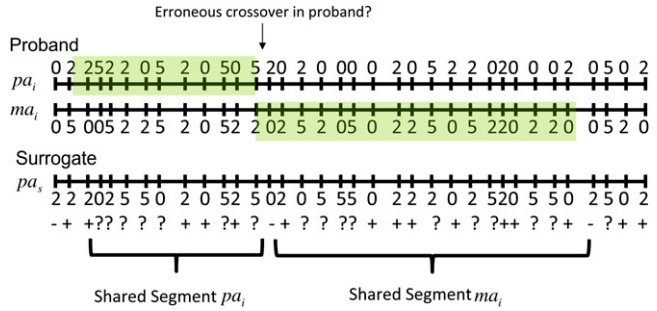

Figure 5 Comparison of proband without genotyped parents to one of its surrogates to illustrate crossover minimization. Alleles on both gametes of proband (pa_i and ma_i) are compared to alleles on surrogate paternal gamete (pa_s) to identify a shared chromosome segment, where ? is a possible match, + is a match, and – is a nonmatch. The proband shares pa_i with the surrogate at the start of the chromosome and then this switches to ma_i. The "switch" loci could be a crossover in the surrogate or, more likely, the paternal and maternal gametes need to be "flipped" in the proband. Proband gametes are flipped only if, after cycling through all surrogates and storing information on all crossovers, there is more evidence for crossovers than no crossovers. Offsets are applied but not shown in the diagram for simplicity.

generations 997 through 999, 100 individuals were simulated. Finally generation 1000 consisted of 200 individuals. for a total of 500 individuals considered for phasing and imputation. The number of individuals in the last generation was achieved by doubling the number of offspring per mating.

The genome simulated consisted of one chromosome which measured 1 M. In generation zero, all individuals were completely homozygous for the same allele in all 40,000 potential loci per Morgan and mutations were applied at a rate of 2.5×10^{-5}/locus/meiosis in the following generations. Mutations switched allele 0 to 2 and vice versa. The number of mutations and crossovers per chromosome were sampled from a Poisson distribution. The mean for the number of mutations corresponded to the product of the number of loci per chromosome (both monomorphic and polymorphic) and the mutation rate, and the mean for crossovers was one per Morgan. The sampled mutations and crossovers were then randomly placed on the chromosome. A more detailed account of the simulations can be found in Daetwyler *et al.* (2010).

Approximately 1500 segregating biallelic loci exceeded a minor allele frequency (MAF) of 0.02 at generation 1000, which is equivalent to a marker density of $7.5N_e$/M. Linkage disequilibrium (LD, r^2) statistics (Hill and Robertson 1968) were 0.169, 0.066, 0.022, 0.008, and 0.005 for neighboring SNP and SNP that were 1, 5, 20, or 50 cM apart, respectively (SE were low). Allele frequency was found to follow a U-shaped distribution as expected.

Testing in simulated data

The utility of ChromoPhase was evaluated in 25 replicates of the data simulated as described above (genome summary = 1 chromosome, 1 M, 1500 loci, 100 QTL). Phasing utility was checked in each replicate consisting of the last four, three, and two generations in the data set. The pedigree used by the program was restricted to the generations being tested. Hence, no additional information was available on ancestors beyond the animals in the genotyped data set. Inferred alleles were compared to true alleles and this yielded the following test parameters for both paternal and maternal alleles, (i) percentage correct, (ii) percentage missing, and (iii) percentage wrong. Statistics on phasing were compiled only for nonfounders.

Imputation of missing genotypes was evaluated in 25 population replicates. Three sparse SNP densities were simulated: 14, 34, and 100/M, which correspond to 420, 1020, and 3000 SNP for a 30-M genome. Markers in the sparse subset were chosen to have higher than average MAF and were evenly distributed across the genome. If an animal was to be imputed, genotypes were set to missing in loci not chosen as part of the sparse set at the beginning of the algorithm. This was done in the last four, three, and two generations to investigate how ChromoPhase copes with varying depths of pedigree and resulted in nine scenarios (*i.e.*, three sparse densities and three pedigree depths). For example, if there are four generations in the data set, three parental generations had dense genotypes (1500 SNP/M) and a proportion of animals in the last generation had sparse genotypes (14–100 SNP/M) and need imputation. Test parameters chosen for imputation of genotypes were, (i) percentage correct, (ii) percentage missing, and (iii) percentage wrong when compared to true genotypes from simulation.

Effect of imperfect imputation of genotypes was evaluated by applying genomic evaluation to the data set. One hundred QTL per Morgan were randomly sampled from the segregating loci. This assured that the number of QTL was larger than the number of independent chromosome segments (Daetwyler *et al.* 2010). Additive allele substitution effects (β) were sampled from N(0,1). True breeding values were calculated for each QTL as $2(1-m_j)\beta_j$ (where m_j is the major allele frequency at locus j), $-2m_j\beta_j$, and $((1-m_j)-m_j)\beta_j$ for the major and minor homozygote and heterozygote genotype, respectively (Falconer and Mackay 1996). Phenotypes were generated by adding random environmental deviations to genotypic values that were also drawn from $N(0,1)$ and scaled to achieve a heritability of 0.3. While the imputation data sets may have included more than two generations (as noted in Table 5), only the last two generations (300 individuals) were used to estimate genomic breeding values to keep sample size constant. The genomic evaluation method fitted a realized relationship matrix (G) based on marker information (NejatiJavaremi *et al.* 1997; Hayes *et al.* 2009b). In each sparse marker density (14, 34, 100 per Morgan), three different scenarios were considered: (i) all 300 individuals were genotyped at high density (Dense, 1500 SNP per Morgan), (ii) all individuals were genotyped at sparse density (Sparse), (iii) individuals in the last generation were imputed to high density

Table 1 Phasing performance in percent in paternal alleles of nonfounder individuals when the data set consisted of 4, 3, or 2 generations (Gen.) of genotyped animals (maximum SE $<$ 0.01)

Gen.	Correct	Missing	Wrong
4	99.97	0.01	0.02
3	99.96	0.01	0.02
2	99.91	0.03	0.07

from the respective sparse densities (Imp). The G matrices were then fitted with phenotypes in ASReml to produce genomic breeding values (Gilmour *et al.* 2000). The QTL were not masked; that is, they were part of the dense genotype data. This was expected to increase the difference in genomic evaluation accuracy between the Dense and Imp scenarios. The accuracy of the genomic evaluation was calculated as the statistical correlation between genomic and true breeding values in the training set.

Comparison with other programs

Our algorithm was compared to the fastPHASE program (Scheet and Stephens 2006). The algorithm described by Burdick *et al.* (2006) was also used to impute missing genotypes. Default settings were used in all cases, as preliminary investigations showed that altering the settings changed the accuracy of imputation $<$1%.

Testing in real Holstein data

Imputation accuracy was tested in a data set of 1183 Holstein bulls, which were densely genotyped with the Illumina Bovine 50K array. Quality control reduced the number of SNP to 39048 SNP and is described in more detail in Hayes *et al.* (2009a). A pedigree consisting of 3674 animals was used to gather the genotyped relatives for each proband. We tested the imputation accuracy on chromosome 1, which had 2529 SNP available. Two scenarios were tested, one in which missing genotypes were imputed from the SNPs on the Illumina Bovine 3K to 50K density and one in which we imputed only 5% of SNP in the 50K SNP chip. The second case was designed to mimic imputing upward from the 50K chip density, because if our algorithm can impute this scenario with a certain accuracy then we should be able to impute from 50K to say 800K or even to full sequence with at least the same accuracy. The Illumina 3K chip had 182 SNP on chromosome 1. Operationally, instead of regenotyping with the 3K chip, we blanked out and imputed the nonsparse SNP in the 50K chip. We tested the imputation of 5% of SNP by random masking. Imputed genotypes were then compared to real 50K genotypes once the algorithm had completed to assess imputation accuracy.

Results

Phasing

Phasing was evaluated in all nonfounders and results can be found in Table 1. The percentage of alleles phased correctly when compared to true alleles was high, ranging from 99.97%, when all generations were available, to 99.91% when only 2 generations were included. Errors decreased slightly as the number of generations increased.

Table 2 Imputation performance (%) when imputing missing genotypes in all individuals in the last generation for three different sparse densities per Morgan and 4, 3, or 2 generations (Gen.) included in the data set (maximum SE $<$ 1.72%)

		Genotypes		
Gen.	Sparse density	Correct	Missing	Wrong
4	14	87.79	0.87	11.34
4	34	94.13	0.29	5.58
4	100	98.67	0.06	1.27
3	14	87.80	1.00	11.20
3	34	94.07	0.31	5.62
3	100	98.57	0.08	1.36
2	14	81.36	1.48	17.2
2	34	83.62	1.49	14.89
2	100	87.07	1.93	11.00

Imputation

Similar trends to phasing were observed for the accuracy of imputation. Table 2 shows means of 25 replicates of imputing missing genotypes at three different sparse densities. The percentage of correctly imputed genotypes was greatest when four generations of data (three generations of dense genotypes and last generation sparse) were available. However, performance was only slightly reduced when three generations were in the data set. The number of nonimputed genotypes increased slightly as the number of generations decreased. The proportion of correctly imputed loci increased and the proportion of wrongly imputed genotypes decreased as the density of the sparse genotypes increased. In Table 3, one can observe that as soon as some individuals (20%) in the last generation have dense genotypes, more information flows to the sparsely genotyped individuals and the proportion of correctly imputed genotypes increases by ~5%. This demonstrates that collateral relatives can be used to improve imputation of missing genotypes when the number of ancestral dense genotypes is limited. When using two or three generations of dense data to impute 80% of sparsely genotyped individuals in the last generation, very similar results to imputing all individuals were obtained. This shows that with sufficient ancestral information, dense genotyping of collateral relatives is of limited value (results not shown).

ChromoPhase correctly imputed approximately 12–16% more missing genotypes than fastPHASE for the scenarios tested (Table 4). Our algorithm was also tested against that of Burdick *et al.* (2006) and was found to impute at least 27% more loci. This may be because the Burdick *et al.* (2006) approach was not designed to handle very sparse marker densities. Computation time in CPU time for ChromoPhase, fastPHASE, and Burdick *et al.* (2006) were approximately 30 CPU seconds, 3 hr 19 min, and 20 sec, respectively.

The accuracy of genomic evaluation (correlation of predicted and true breeding values) using dense genotypes

Table 3 Imputation performance (%) when imputing missing genotypes in 80% of individuals in the last generation for three different sparse densities per Morgan and 2 generations (Gen.) included in the data set (maximum SE < 1.5%)

		Genotypes		
Gen.	Sparse density	Correct	Missing	Wrong
2	14	84.89	1.37	13.75
2	34	89.39	0.86	9.76
2	100	94.09	0.61	5.30

Table 4 Imputation performance of ChromoPhase, fastPHASE and Burdick *et al.* (2006) in replicate 1, shown as the percentage of correctly imputed missing genotypes in all individuals in the last generation for three different sparse densities per Morgan and 3 generations (Gen.) included in the data set

Gen.	Sparse density	ChromoPhase	fastPHASE	Burdick *et al.* (2006)
3	14	87.8	75.3	60.5
3	34	94.1	78.6	60.8
3	100	98.6	85.0	61.6

to calculate the genomic relationship matrix (G) was 75.5% (Table 5). While the depth of the data set was varied between two and four generations, only the last two generations (300 animals) were used to estimate genomic breeding values to keep the sample size constant. The accuracies of the sparse scenarios were all very similar at 56.1%. At these low densities, increasing the sparse loci set slightly seemed not to increase accuracy. As expected, accuracy increased when more missing genotypes were imputed. However, it was apparent that although imputation was imperfect, the accuracy of genomic estimated breeding values achieved for the sparse subset of markers as a percentage of that achieved for the dense accuracy (*i.e.*, accuracy with 1500 loci) was in all cases greater than the proportion of correctly imputed genotypes (Table 5). Consider the scenario with three generations in the data set and a sparse genotype of 14 loci/M; here 87.8% of missing genotypes were correctly imputed but 95% of the dense genomic evaluation accuracy was achieved. Thus, it seems that genomic evaluation is able to cope well with a percentage of loci missing or even wrongly imputed, which has been confirmed independently by Weigel *et al.* (2010) using a Bayesian method to estimate SNP effects for genomic predictions.

Real data present additional challenges such as ungenotyped parents, genotyping errors, and map location errors. Thus the imputation accuracy in real Holstein cattle data was reduced when compared to results in simulated data (Table 6). Nevertheless, the imputation accuracy in real Holstein cattle data using the sparse 3K chip was 92.5% in nonfounders and 72.8% in founders. When 5% of 50K loci were randomly masked, the imputation accuracy was 97.2% in nonfounders and 90.2% in founders. The lower performance in real data when compared to simulated data is due in part to imperfections in the real data, such as map and genotyping errors. In addition, there were many more genotyped founders in the simulated data set and our Holstein data set contained no genotyped dams. It is expected that in the future more genotypes would be available (including more dams), which would allow imputation accuracy in real data to move toward results achieved in simulated data.

Discussion

We have described a long-range phasing and imputation algorithm that seeks out and phases long chromosomal segments that are shared between close or distant relatives. The algorithm takes advantage of family data, in addition to population data, to improve performance over methods that phase per locus or consider only a few loci at a time and use population data only. The results demonstrate that ChromoPhase is accurate for both pure phasing of genotyped loci and for imputation of missing genotypes. The identification of shared chromosome segments is important to phase and impute genotypes, as any missing information within a segment can potentially be filled in the proband with information from its surrogate. The key aspect of identifying a shared segment is the recognition that the probability that a long haplotype coalesces to a common ancestor becomes high if two animals match at a high proportion of alleles within a segment and there are no nonmatches. Kong *et al.* (2008) made use of this concept in individuals of unknown relationship by searching for a sufficiently long stretch of loci with no incompatible genotypes that can therefore be assumed to have originated in a common ancestor. All potential surrogates of a proband for a genome segment of predefined length were identified and stored at the beginning of the Kong algorithm. They then phased a proband by cycling through its surrogates to identify a homozygote at a particular locus. Our approach is similar, but operationally different, as we also make use of pedigree information and thus we are able to compare alleles within family relationships and to compare at the level of the allele instead of genotypes. Our algorithm compares relatives in each iteration and genomic surrogates in later iterations to make use of new information as it becomes available and we do not specify a maximum length for shared segments. Thus, a shared segment may potentially span the full chromosome and allows us to use all available information. Consequently ChromoPhase uses most of the information used by Kong *et al.* (2008), but uses some additional information, such as pedigree and simple rule-based filling.

The use of both family and population genomic information makes our approach feasible in species that have incomplete pedigree information or have only few parents with dense genotypes. Genotyped parents allow for rule-based filling of alleles, which is important in early iterations as little information is available to distinguish gametes. Our results indicate that having at least one densely genotyped parent is crucial to achieving high-accuracy imputation. This is partly because rule-based filling is difficult in individuals

Table 5 Accuracy of estimated breeding values in percent when using dense (Dense), sparse (Sparse), and imputed (Imp) genotypes to calculate realized relationship matrices (maximum SE < 1.0%)

Gen.	Density	Scenario	Accuracy of genomic evaluation	% of all dense accuracy
	1500	Dense	75.5	100.0
		Sparse	56.1	74.3
4	14	Imp	71.7	95.0
4	34	Imp	74.1	98.1
4	100	Imp	75.5	100.0
3	14	Imp	71.7	95.0
3	34	Imp	74.0	98.0
3	100	Imp	75.5	100.0
2	14	Imp	68.8	91.1
2	34	Imp	71.2	94.3
2	100	Imp	72.7	96.3

Gen. refers to the number of generations in data set used for imputation. Sample size 300 (*i.e.*, last two generations) for genomic evaluation in all scenarios.

Table 6 Imputation performance of ChromoPhase in percent on chromosome one in real Holstein cattle data tabulated as the percentage correctly imputed missing genotypes in the youngest founders (F) and nonfounders (NF) for two sparse densities

Sparse density	Type	No. of animals imputed	Genotypes		
			Correct	Missing	Wrong
182	NF	112	92.5	1.1	6.3
182	F	212	72.8	0.6	26.5
2400	NF	112	97.2	1.3	1.5
2400	F	278	90.2	0.7	9.1

without genotyped parents and therefore a larger number of phasing or imputation errors are likely to occur in these founders. Imputation is especially difficult in younger animals that have no genotyped parents or genotyped offspring available (Table 6). Our algorithm may need further development to be able to capitalize on population linkage disequilibrium more effectively and improve imputation rates in founders. However, comparing to relatives, if available, results in significant computer time savings, as animals do not have to be compared to all other animals. Furthermore, restricting comparisons to relatives in the first iterations reduced error rates.

The comparison of haplotypes in our algorithm also results in computational efficiency because the same process is used for phasing and imputation. The main objective of ChromoPhase is to complete as much information as possible in a proband haplotype by using information from shared segments with relatives. It is therefore irrelevant from the method's point of view whether this is for phasing or imputation, although the algorithm benefits when genotypes are available at a locus.

Haplotype libraries have been proposed as a way to phase and impute genotypes (Hickey *et al.* 2011; Vanraden *et al.* 2011). Conceptually, these approaches are not very different from ChromoPhase. While they explicitly build a library of haplotypes, our algorithm contains the haplotypes in each individual with dense genotypes. A library may provide computational efficiencies when there are only few main haplotypes segregating in a population so that the proband needs only to be compared to few haplotypes. This may be the case in domestic animal populations with low N_e. A drawback of the library approach is that the haplotypes are restricted to a specific length. As an example, consider a SNP that has been mapped to the wrong location on the genome or has been incorrectly genotyped in the lab. First, this would inflate the number of haplotypes stored in the library and, second, if an individual has a genotyping error, a match cannot be identified in the library for that whole segment, resulting in reduced imputation. In contrast, our approach compares animals up to the map or genotyping error and, if a nonmatch occurs, will stop the shared segment. However, the next shared segment may start right after the nonmatch increasing imputation when compared to a library approach.

The application of our method in real data sets may require further development to address several challenges, such as completely ungenotyped animals in the data, incomplete pedigrees, genotyping errors, rare alleles, and SNP mapped to wrong genome locations. Currently, completely ungenotyped individuals are not attempted to be imputed. However, doing so should be feasible if an ungenotyped individual has both a genotyped parent and genotyped progeny. It is not expected that imputing completely ungenotyped individuals would increase imputation rates, as they offer no additional information. In fact, it was observed that more errors occurred when ungenotyped animals were imputed and used to impute sparsely genotyped animals, because proportionally more errors occur in completely ungenotyped animals, which then transfer to sparsely imputed animals. It is also important that correct and as complete as possible pedigree information is available for determining surrogates. Most genotyping errors can be detected by comparing trios, although if they are not detected then they may result in erroneous haplotype assignments. Rare alleles can be imputed only if they are observed in the dense genotypes; hence the dense sample needs to be sufficiently large. Map errors may cause a wrongly mapped locus to appear shared between relatives where it may be a match only by chance (*i.e.*, it is only IBS) causing phasing errors. Every effort should be made to correct map errors on SNP chips.

Currently the algorithm applies to autosomes and further modification to sex chromosomes may be necessary. Crossover occurs freely between X chromosomes; hence phasing involving females is expected to be the same as autosomes. Simplification may be possible in males because X-specific markers in the nonparalogous region are already phased. Pseudoautosomal crossover between X and Y is believed to be restricted to the relatively short regions at either end of the X chromosome in human representing approximately 2% of all bases in total (Charlesworth 1991; Ross *et al.* 2005). It has been reported that the ratio of genetic *vs.* physical distance is inflated in the pseudoautosomal region,

which may indicate high haplotype diversity (*e.g.*, Arias *et al.* 2009; Dumont *et al.* 2011).

The application highlighted here is to impute haplotypes and missing genotypes from sparse to medium density. The current study confirms that it is feasible to impute a large number of missing genotypes to a higher density, although the performance is dependent on the sparse chip density and the number of generations in the data set. It will be feasible to use our method to impute 50K genotypes to even higher density once information from such denser SNP chips becomes available in their relatives. So there is no upper limit to how dense the genotypes can be for successful imputation, and even imputing full genome sequence data will eventually be possible once sufficient ancestors have been sequenced.

An interesting result is that when we applied genomic evaluation to the dense, sparse and imputed data sets, the reduction in genomic evaluation accuracy is small with imperfectly imputed genotype data. It was apparent that while imputation is imperfect, the proportion of dense accuracy observed was in all cases larger than the proportion of correctly imputed genotypes. This may be due to more than one SNP being in linkage disequilibrium with each of the QTL across the genome, so that the consequence of a single incorrectly imputed SNP on predicting the effect on a QTL is reduced.

The potential for ChromoPhase to increase the number of genotyped individuals while simultaneously reducing genotyping costs is very large. Key benefits will be increased sample sizes to achieve higher accuracies in genomic selection and to increase the power of QTL studies. Reducing genotyping costs through strategic genotyping of ancestors and upgrading to denser genotyping from sparser SNP chips in the current generation with programs such as ChromoPhase will allow for the application of genomic selection in species for which currently this technology is not economically feasible.

Acknowledgments

The authors thank Phil Bowman for providing a Holstein pedigree and two anonymous reviewers for their helpful comments. H.D.D. was in part supported by the SABRETRAIN Project, which is funded by the Marie Curie Host Fellowships for Early Stage Research Training, as part of the 6th Framework Program of the European Commission. The software is available by request from the corresponding author.

Literature Cited

Arias, J., M. Keehan, P. Fisher, W. Coppieters, and R. Spelman, 2009 A high density linkage map of the bovine genome. BMC Genet. 10: 18.

Baruch, E., J. I. Weller, M. Cohen-Zinder, M. Ron, and E. Seroussi, 2006 Efficient inference of haplotypes from genotypes on a large animal pedigree. Genetics 172: 1757–1765.

Browning, B. L., and S. R. Browning, 2009 A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. Am. J. Hum. Genet. 84: 210–223.

Burdick, J. T., W. M. Chen, G. R. Abecasis, and V. G. Cheung, 2006 In silico method for inferring genotypes in pedigrees. Nat. Genet. 38: 1002–1004.

Calus, M. P. L., T. H. E. Meuwissen, A. P. W. de Roos, and R. F. Veerkamp, 2008 Accuracy of genomic selection using different methods to define haplotypes. Genetics 178: 553–561.

Charlesworth, B., 1991 The evolution of sex-chromosomes. Science 251: 1030–1033.

Clark, A. G., 1990 Inference of haplotypes from pcr-amplified samples of diploid populations. Mol. Biol. Evol. 7: 111–122.

Daetwyler, H. D., R. Pong-Wong, B. Villanueva, and J. A. Woolliams, 2010 The impact of genetic architecture on genome-wide evaluation methods. Genetics 185: 1021–1031.

Dumont, B. L., M. A. White, B. Steffy, T. Wiltshire, and B. A. Payseur, 2011 Extensive recombination rate variation in the house mouse species complex inferred from genetic linkage maps. Genome Res. 21: 114–125.

Elston, R. C., and J. Stewart, 1971 General model for genetic analysis of pedigree data. Hum. Hered. 21: 523.

Falconer, D. S., and T. F. C. Mackay, 1996 *Introduction to Quantitative Genetics*. Longman, Harlow, UK.

Gilmour, A. R., R. Thompson, B. R. Cullis, and S. J. Wellham, 2000 *ASReml Reference Manual*. NSW Department for Primary Industries, New South Wales.

Goddard, M. E., 2008 The use of high density genotyping in animal health. Anim. Genomics Anim. Health 132: 383–389.

Hayes, B. J., P. J. Bowman, A. J. Chamberlain, K. L. Verbyla, and M. E. Goddard, 2009a Accuracy of genomic breeding values in multi-breed dairy cattle populations. Genet. Sel. Evol. 41: 51.

Hayes, B. J., P. M. Visscher, and M. E. Goddard, 2009b Increased accuracy of artificial selection by using the realized relationship matrix. Genet. Res. 91: 47–60.

Hickey, J., B. Kinghorn, B. Tier, J. Wilson, N. Dunstan *et al.*, 2011 A combined long-range phasing and long haplotype imputation method to impute phase for SNP genotypes. Genet. Sel. Evol. 43: 12.

Hill, W. G., and A. Robertson, 1968 Linkage disequilibrium in finite populations. Theor. Appl. Genet. 38: 226–231.

Kong, A., G. Masson, M. L. Frigge, A. Gylfason, P. Zusmanovich, *et al.*, 2008 Detection of sharing by descent, long-range phasing and haplotype imputation. Nat. Genet. 40: 1068–1075.

Lander, E. S., and P. Green, 1987 Construction of multilocus genetic-linkage maps in humans. Proc. Natl. Acad. Sci. USA 84: 2363–2367.

Meuwissen, T. H. E., and M. E. Goddard, 2000 Fine mapping of quantitative trait loci using linkage disequilibria with closely linked marker loci. Genetics 155: 421–430.

Meuwissen, T. H. E., B. J. Hayes, and M. E. Goddard, 2001 Prediction of total genetic value using genome-wide dense marker maps. Genetics 157: 1819–1829.

Nejati-Javaremi, A., C. Smith, and J. P. Gibson, 1997 Effect of total allelic relationship on accuracy of evaluation and response to selection. J. Anim. Sci. 75: 1738–1745.

Pong-Wong, R., A. W. George, J. A. Woolliams, and C. S. Haley, 2001 A simple and rapid method for calculating identity-by-descent matrices using multiple markers. Genet. Sel. Evol. 33: 453–471.

Reik, W., and J. Walter, 2001 Genomic imprinting: parental influence on the genome. Nat. Rev. Genet. 2: 21–32.

Ross, M. T., D. V. Grafham, A. J. Coffey, S. Scherer, K. McLay *et al.*, 2005 The DNA sequence of the human X chromosome. Nature 434: 325–337.

Scheet, P., and M. Stephens, 2006 A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. Am. J. Hum. Genet. 78: 629–644.

Tier, B., 2006 Haplotyping for linkage disequilibrium. Proceedings of the 8th World Congr. Genet. Appl. Livest. Prod., Belo Horizonte, Brazil, August pp. 21–01.

VanRaden, P. M., J. R. O'Connell, G. R. Wiggans, and K. A. Weigel, 2011 Genomic evaluations with many more genotypes. Genet. Sel. Evol. 43: 10.

Villumsen, T. M., and L. Janss, 2009 Bayesian genomic selection: the effect of haplotype length and priors. *BMC Proc.* 3(suppl. 1): S11.

Weeks, D. E., E. Sobel, J. R. Oconnell, and K. Lange, 1995 Computer-programs for multilocus haplotyping of general pedigrees. Am. J. Hum. Genet. 56: 1506–1507.

Weigel, K. A., G. de los Campos, A. I. Vazquez, G. J. M. Rosa, D. Gianola *et al.*, 2010 Accuracy of direct genomic values derived from imputed single nucleotide polymorphism genotypes in Jersey cattle. J. Dairy Sci. 93: 5423–5435.

Windig, J. J., and T. H. Meuwissen, 2004 Rapid haplotype reconstruction in pedigrees with dense marker maps. J. Anim. Breed. Genet. 121: 26–39.

Wood, A. J., and R. J. Oakey, 2006 Genomic imprinting in mammals: emerging themes and established theories. PLoS Genet. 2: e147.

Communicating editor: I. Hoeschele

GENETICS

Supporting Information
http://www.genetics.org/content/suppl/2011/06/24/genetics.111.128082.DC1

Imputation of Missing Genotypes From Sparse to High Density Using Long-Range Phasing

Hans D. Daetwyler, George R. Wiggans, Ben J. Hayes, John A. Woolliams, and Mike E. Goddard

DOI: 10.1534/genetics.111.128082

File S1

Supporting Data

File S1 is available as a compressed folder (.zip) at http://www.genetics.org/content/suppl/2011/06/24/genetics.111.128082.DC1.

This folder contains:

- Genotype data from Illumina Bovine 50K array, alligned to Btau 4.0
- Data for a sample of Holstein Friesian animals on chromosome 1
- Data for Replicate 1 and Replicate 2

File S2
ChromoPhase Pseudo Code

Supporting Material for "Imputation of Missing Genotypes from Sparse to High Density using Long-Range Phasing", Daetwyler et al. 2011, Genetics.

- Read sorted pedigree
- Read genotypes
- Collect relatives from youngest to oldest
    - If sire(proband) known
        - Add proband to offspring of sire
        - If sire genotyped
            - Add sire to paternal relatives of proband
        - End if
        - If proband is genotyped
            - Add proband to list of descendents of sire
            - Store degree removed of proband to sire
        - Else
            - Store genotyped descendents of proband as descendents of sire
            - Increase degree removed by 1
        - End if
    - Do same process for dam(proband)
- Collect surrogate fathers and mothers, from oldest to youngest
    - If sire(proband) known
        - If sire(proband) not genotyped
            - Add all paternal relatives of sire to paternal relatives of proband
            - Increase degree removed of each new paternal relative by 1
            - Do same for maternal relatives of sire
            - If sib(proband) genotyped
                - Add to paternal relatives of proband
            - Else
                - Add all sibs descendents to paternal relatives of proband
            - End if
        - End if
    - End if
    - Do same for dam of proband
- Collect all relatives in one array
    - Loop through paternal, maternal relatives and descendents
        - Add to list of relatives
        - Store degree removed
        - Store whether paternal (1), maternal (2) or descendent (3)
    - End loop
- Set desired SNP to missing to test imputation in genotyped animals
    - Store full genotypes in array for later checking
    - Option 1, read SNP to be set to missing from file
        - Set logical variable for SNP set to missing 'impute-snp to true (initialised as false)
        - Randomly choose desired proportion of probands as animals to be imputed and set logical variable 'impute-animal' to true (initialised as false)
        - Loop through all probands
            - Where impute-animal and impute-snp is true, set genotype to 5 (missing)
        - End loop
- Option 2, set random SNP to missing instead of reading in list, choose impute-animal as above

- Set genotyping errors to missing and detect pedigree errors
    - If proband and sire(proband) are genotyped
        - Loop through SNP
            - If homozygote genotypes oppose
                - Loop through other offspring of sire
                - If no other offspring conflict with sire
                - set only original proband genotype to 5
                - Else if sire is in conflict with more than 20% of offspring
                - set sire genotype to 5
                - Else
                - Set this genotype to 5 for sire and all offspring
                - End if
                - End loop
            - End if
        - End loop
    - Do same for dams
- Initialise haplotypes
    - If proband genotype is known and homozygous
        - Fill in both haplotypes
    - Else
        - Set both haplotypes to 5
    - End if

Start iterations through all genotyped probands

- If iteration = maximum phasing iteration
    - Cut offset (number of loci cut of end of shared segments) to 0
    - Set minimum length of shared segments to 50 loci
- Else if iteration > phasing iteration
    - Reset offset and run length to original values, 20 and 100, respectively
- End if
- If iteration = maximum imputation iteration
    - Cut offset (number of loci cut of end of shared segments) to 0
    - Set minimum length of shared segments to 50 loci
- End if
- If iteration = 1, call COLL_GENOMIC_SURROGATE(FALSE)
- If iteration > max. phasing iteration +1, call COLL_GENOMIC_SURROGATE(TRUE)
- Loop through all genotyped probands
    - If iteration ≤ max. phasing iteration
        - only loop through animals with full genotypes
    - else
        - only loop through imputed animals
    - end if
    - In first three phasing or imputation iterations, re-initialise haplotypes
    - If sire(proband) OR dam(proband) are genotyped (i.e. proband is a non-founder)
        - Apply rule-based filling of haplotypes
        - Loop through relatives with dense genotypes < 4 degrees removed
            - Call ID_SHARED_RUN_NON_FOUNDER
            - In last imputation iteration call FIND_RUN_NON_FOUNDER
        - End loop
        - If iteration > phasing iteration -2
            - Loop through genomic surrogates
                - Call ID_SHARED_RUN_NON_FOUNDER
            - End loop
        - End if
        - Call VOTE

- Else (i.e. proband has no genotyped parents, Founder)
  - Loop through relatives with dense genotypes < 4 degrees removed
    - Call FIND_RUN_FOUNDERS(TRUE)
  - End loop
  - If iteration = max phasing iteration OR iteration > max. phasing + 2
    - Loop through genomic surrogates
      - Call FIND_RUN_FOUNDERS(TRUE)
    - End loop
  - End if
  - Check for apparent crossovers
    - Loop 20 times
      - If no evidence for erroneous crossovers ➔ EXIT loop
      - Loop through relatives
        - Call FIND_RUN_FOUNDERS(FALSE)
      - End loop
      - If iteration > phasing iteration -2
        - Loop through genomic surrogates
          - Call FIND_RUN_FOUNDERS(FALSE)
        - End loop
      - End if
      - Call FLIP_CROSSOVERS
    - End loop
- End loop for non-founders and founders
- End loop through probands

End iteration

- Evaluate imputation by comparing imputed genotypes to original dense genotypes
- In simulated data, evaluate phasing

SUBROUTINES:

COLL_GENOMIC_SURROGATE (logical)

- Loop through genotyped animals
  - If logical=FALSE
    - If animal to be imputed, CYCLE
    - Set min-genomic-surrogate-segment = minimum length of shared run
  - Else if logical=TRUE
    - If densely genotyped animal, CYCLE
    - Set min-genomic-surrogate-segment = (minimum length of shared run)*4
  - End if
  - Loop through densely genotyped animals
    - Loop through SNP
      - If segment count > min-genomic-surrogate-segment, EXIT
      - If logical=FALSE
        - If abs(proband genotype – surrogate genotype) ≠ 2
          - Increase segment count by 1
        - Else
          - Segment count = 0
        - End if
      - Else if logical = TRUE
        - If proband genotype = surrogate genotype
          - Increase segment count by 1
        - Else if proband genotype ≠ surrogate genotype
          - If proband genotype homozygous
            - Segment count = 0

                - End if
            - End if
        - End if
    - End loop
    - If segment count > min-genomic-surrogate-segment
        - Add densely genotyped animal to proband's list of genomic surrogates
    - End if
    - Segment count = 0
- End loop
- End loop

ID_SHARED_RUN_NON_FOUNDER

- If densely genotyped proband, threshold for missing info = 0.20
- If iteration = max. phasing iteration, threshold for missing info = 1.00 (i.e. no threshold)
- Loop through SNP
    - Check if proband haplotype matches surrogate's
    - If it matches
        - Increase loci count by 1
        - If proband haplotype = 5, then increase missing loci count by 1
    - Else
        - If ratio-missing > (1- proportion-SNP-to be imputed/2), multiply offset by 4
        - If run long enough and ratio of missing info < threshold
            - Apply offset to length of run
            - Loop through remaining run of loci
                - If surrogate haplotype = 5
                    - Number total counts +1
                - End if
                - If surrogate haplotype = 2
                    - Number 2counts +1
                - End if
            - End loop
        - End if
    - End if
- End loop

VOTE

- Loop through SNP
    - Loop through proband chromosomes
        - If proband haplotype = 5
            - If no information found in surrogates (i.e. number total counts = 0)
                - Allele ratio = 0.5
            - Else
                - Allele ratio = number 2counts / number total counts
            - End if
        - End if
        - If proband haplotype = 5
            - If allele ratio > 0.7
                - Proband haplotype = 2
            - Else if allele ratio < 0.3
                - Proband haplotype = 0
            - Else
                - Proband haplotype = 5

- End id
- End if
- End loop
- End loop

FIND_RUN_NON_FOUNDER

- Loop through proband haplotypes
  - Loop through SNP
    - Check if proband haplotype matches surrogate's haplotype
      - If it matches
        - Loci count + 1
        - If either haplotype missing, missing loci count + 1
      - Else if loci count > minimum run length AND proportion missing loci < 0.5
        - If ratio-missing > (1- proportion-SNP-to be imputed/2), multiply offset by 4
        - Apply offset to length of run
        - Loop through remaining run of loci
          - If proband haplotype = 5
            - Fill with allele of surrogate if known
          - End if
        - End loop
        - Reset loci and missing loci count
      - End if
  - End loop
- End loop

FIND_RUN_FOUNDERS(logical)

- Loop through SNP
  - Check if either proband haplotype matches surrogate haplotype
  - If one or both match and matching haplotype has not switched from previous locus
    - Store which haplotype matched
    - If no previous match, set SNP as first in run
    - Increase loci count by 1
    - If any of the haplotypes are 5, increase missing loci count by 1
  - End if
  - If haplotypes do not match
    - If loci count > minimum run length AND proportion missing loci < 0.5
      - Define last SNP in run as previous SNP
      - If (logical)=FALSE
        - Count crossovers and non-crossovers at each locus in run
      - Else
        - If ratio-missing > (1- proportion-SNP-to be imputed/2), multiply offset by 4
        - Apply offset to length of run
        - Loop through remaining run of loci
          - If proband haplotype = 5
            - Fill with allele of surrogate if known
          - End if
        - End loop
      - End if
    - End if
    - Store

            - Reset loci count and missing loci count to zero
        - End if
        - If matching switched from one proband haplotype to the other
            - Set first SNP in new run
            - Store which haplotype matched
        - End if
- End loop

FLIP_CROSSOVERS

- Set maximum crossover ratio to 0.99
- Loop through SNP
    - If ratio of (crossovers / (no-crossovers +1)) > maximum crossover ratio
        - maximum crossover ratio = crossovers / (no-crossovers+1)
        - RSNP = locus
        - Flag=1
    - else
        - if (Flag=1) then
            - crossover-point (i) = RSNP
            - crossover-ratio(i)=maxrec
            - Flag=0
            - maxrec = 0.99
            - i = i + 1
        - end if
    - end if
- end loop

- if i > 1
    - loop through i's in pairs
        - if crossover-ratio ≥ 1.0
            - flip all loci between i's
        - end if
    - end loop
- end if