Published in final edited form as:

Genet Epidemiol. 2014 September ; 38(6): 531–541. doi:10.1002/gepi.21838.

PedBLIMP: Extending Linear Predictors to Impute Genotypes in Pedigrees

Wenan Chen and Daniel J. Schaid

Department of Health Sciences Research, Division of Biomedical Statistics and Informatics, Mayo Clinic, Rochester, Minnesota

Abstract

Recently, Wen and Stephens [Wen and Stephens 2010] proposed a linear predictor, called BLIMP, that uses conditional multivariate normal moments to impute genotypes with accuracy similar to current state-of-the-art methods. One novelty is that it regularized the estimated covariance matrix based on a model from population genetics. We extended multivariate moments to impute genotypes in pedigrees. Our proposed method, PedBLIMP, utilizes both the linkage disequilibrium (LD) information estimated from external panel data and the pedigree structure or identity by descent (IBD) information. The proposed method was evaluated on a pedigree design where some individuals were genotyped with dense markers and the rest with sparse markers. We found that incorporating the pedigree/IBD information can improve imputation accuracy compared to BLIMP. Because rare variants usually have low LD with other single nucleotide polymorphisms (SNPs), incorporating pedigree/IBD information largely improved imputation accuracy for rare variants. We also compared PedBLIMP with IMPUTE2 and GIGI. Results show that when sparse markers are in a certain density range, our method can outperform both IMPUTE2 and GIGI.

Keywords

linear predictor; genotype imputation; linkage disequilibrium; identity by descent

Introduction

Genotype imputation is a technique to infer missing genotypes based on observed genotype data [Li, et al. 2009; Marchini and Howie 2010]. It can be used to merge data sets where genotypes are produced from different platforms. It can also improve statistical power by increasing the number of SNPs to be tested because, usually, the genotyped SNPs are simply tag SNPs of the causal variants. In general, imputation techniques can be grouped into two categories based on the relatedness of genotyped individuals. The first category deals with genotypes from unrelated individuals. In this case, the inference of missing genotypes relies on correlation among densely genotyped markers due to linkage disequilibrium (LD). Several popular methods have been developed, for example, PHASE [Stephens and Scheet 2005], fastPHASE [Scheet and Stephens 2006], IMPUTE2 [Howie, et al. 2009; Marchini, et

al. 2007], BEAGLE [Browning and Browning 2009; Browning and Yu 2009; Browning and Browning 2007], MACH [Li, et al. 2009; Li, et al. 2010], or a combination of a phasing method SHAPIT2 [Delaneau, et al. 2012; Delaneau, et al. 2013] and IMPUTE2. These methods usually use a Hidden Markov Model (HMM) to model the construction of the haplotypes. When individuals are related from a pedigree, the relatedness can be exploited to some degree, e.g., haplotype estimation can be improved by jointly modeling all related individuals [Howie, et al. 2009]; BEAGLE and SHAPEIT2 can include duos or trios of related subjects. However, more general pedigree structures cannot be incorporated explicitly.

The second category of imputation deals with genotypes from related individuals from pedigrees, such as GHOST [Burdick, et al. 2006; Chen and Abecasis 2007], MERLIN [Abecasis, et al. 2002] and GIGI [Cheung, et al. 2013]. The inference of missing genotypes relies on the allele sharing or IBD among related individuals in a pedigree. GHOST uses the Elston-Stewart algorithm, whose complexity increases exponentially with the number of markers being analyzed [Chen and Abecasis 2007; Elston and Stewart 1971]. MERLIN uses the Lander-Green algorithm whose complexity increases exponentially with the pedigree size [Abecasis, et al. 2002; Lander and Green 1987]. To overcome the computing burden for large pedigrees with a large number of markers, GIGI samples inheritance vectors (IVs) from a Markov Chain Monte Carlo (MCMC) sampler using a set of sparse framework markers. It then infers missing genotypes probabilistically based on sampled IVs and available dense genotypes. There are also long-range phasing and imputation methods [Kong, et al. 2008] [Daetwyler, et al. 2011]. All these pedigree-based methods rely solely on related individuals to impute missing genotypes through estimated IBD among them.

A limit of current pedigree-based methods is that they cannot use LD information from a reference panel. For example, when the probability of IBD is 0, the missing alleles can only be imputed using the population allele frequencies. This may be a problem for some designs where only the last generation is available for genotyping. Figure 1 illustrates this problem. The pedigree represents a common study design for older onset disease: only genotypes on the third generation are available. For each pair of the two siblings, one (solid red) might be genotyped with high density markers and the other (gradient blue) with low density markers. The combination of both low density and high density markers could be the result that all individuals were genotyped with low density markers in the past, e.g., in linkage analysis, and only some individuals are selected for high density genotyping. These could be a tradeoff between genotype density and available budget. This design is the focus for imputation in this paper. We can see that the regions represented by C and D in individuals 2 and 4 are not in IBD with any region from individuals densely genotyped. Incorporating LD may improve imputation accuracy for these regions.

A natural idea is to use both pedigree/IBD information and LD information when imputing genotypes in pedigrees. The method we proposed here is motivated by two papers. The first is the BLIMP method for imputing unrelated individuals proposed by Wen and Stephens [Wen and Stephens 2010]. BLIMP uses a linear predictor to impute genotypes based on the first two moments, i.e., the mean and covariance, of genotypes. It also developed a regularized/shrunken estimation of the mean and covariance based on a model from

population genetics. The second is the derived mean and covariance structure of genotypes from individuals in a pedigree by Schaid et al. [Schaid, et al. 2013]. In this paper, we combined these two components together and proposed a method called PedBLIMP extending BLIMP to impute genotypes in pedigrees. The proposed method utilizes both linkage disequilibrium (LD) information estimated from external panel data and the pedigree/IBD information. In the simulation studies, we compare PedBLIMP with BLIMP, IMPUTE2, SHAPEIT2 for pre-phasing plus IMPUTE2 (SHAPEIT2 + IMPUTE2), and GIGI.

Methods

BLIMP

BLIMP [Wen and Stephens 2010] uses a linear predictor to estimate the dosage vector of untyped SNPs, $\mathbf{Y}_u$, based on the dosage vector of typed SNPs, $\mathbf{Y}_t$, from the same individual. It utilizes the correlation between SNPs close to each other due to LD. Here we cast the method in the context of a linear predictor that minimizes the mean squared error

$E \left(\left\| \hat{\mathbf{Y}}_u - \mathbf{Y}_u \right\|^2 \right)$, so that the assumption of normal distribution is not necessary. In general, the minimum mean squared error (MMSE) estimator can be expressed as $\hat{\mathbf{Y}}_u = E[\mathbf{Y}_u | \mathbf{Y}_t]$. With further assumption of a linear form, i.e., $E[\mathbf{Y}_u | \mathbf{Y}_t] = \mathbf{A}\mathbf{Y}_t + \mathbf{b}$, where $\mathbf{A}$ is a matrix and $\mathbf{b}$ is a vector, and solving $\mathbf{A}$ and $\mathbf{b}$ using MMSE criterion, the best linear predictor $\hat{\mathbf{Y}}_u$ and $Cov(\hat{\mathbf{Y}}_u)$ can be calculated using only the first and second moments as follows:

$$\hat{\mathbf{Y}}_u = \boldsymbol{\mu}_u + \boldsymbol{\Sigma}_{ut} \boldsymbol{\Sigma}_{tt}^{-1} (\mathbf{Y}_t - \boldsymbol{\mu}_t), \quad (1)$$

$$Cov(\hat{\mathbf{Y}}_u) = \boldsymbol{\Sigma}_{uu} - \boldsymbol{\Sigma}_{ut} \boldsymbol{\Sigma}_{tt}^{-1} \boldsymbol{\Sigma}_{tu}, \quad (2)$$

where $\boldsymbol{\mu}_u$ and $\boldsymbol{\mu}_t$ are the means of $\mathbf{Y}_u$ and $\mathbf{Y}_t$ respectively, $\boldsymbol{\Sigma}_{uu}$ and $\boldsymbol{\Sigma}_{tt}$ are the covariance matrices of $\mathbf{Y}_u$ and $\mathbf{Y}_t$ respectively, $\boldsymbol{\Sigma}_{ut}$ and its transpose $\boldsymbol{\Sigma}_{tu}$ are the covariance matrices between $\mathbf{Y}_u$ and $\mathbf{Y}_t$. The mean and covariance matrix of the genotype vector can be estimated from panel data $\mathbf{M}$, which usually contains independent haplotypes. Equations (1) and (2) are the general forms of linear predictors not specific to BLIMP, which we refer to later when discussing PedBLIMP.

The problem with the ordinary sample covariance matrix estimated from $\mathbf{M}$ is that the number of SNPs is usually larger than the number of haplotypes, therefore the estimated covariance matrix is singular. In order to take the inverse of the estimated covariance matrix, which is necessary in the linear predictor, BLIMP gives an elegant solution based on a model from population genetics. It assumes that the haplotypes are generated following Li and Stephens's model [Li and Stephens 2003]. Then the derived conditional mean $\boldsymbol{\mu}_h = E(\mathbf{h} | \mathbf{M})$ and covariance $\boldsymbol{\Sigma}_h = Cov(\mathbf{h} | \mathbf{M})$ for the randomly generated haplotype vector $\mathbf{h}$ can be estimated as follows:

$$\begin{aligned}\hat{\boldsymbol{\mu}}_h &= (1 - \theta) \mathbf{f}^{panel} + \frac{\theta}{2} \mathbf{1}, \\ \hat{\boldsymbol{\Sigma}}_h &= (1 - \theta)^2 \mathbf{S} + \frac{\theta}{2} \left(1 - \frac{\theta}{2}\right) \mathbf{I},\end{aligned}$$

where θ is a parameter relating to mutation, $\mathbf{f}^{panel}$ is the maximum likelihood estimate of the mean, $\mathbf{1}$ is a vector of ones, and $\mathbf{I}$ is the identity matrix. $\mathbf{S}$ is a shrunk version of the maximum likelihood estimate of the covariance matrix $\mathbf{S}^{panel}$ as follows:

$S_{ij} = e^{-\rho_{ij}/(2m)} \Sigma_{ij}^{panel}$, where m is the number of haplotypes in the panel, $\rho_{ij} = 4N_e d_{ij}$ is the population scaled recombination rate between SNPs i and j , and N_e is the effective population size. Note that d_{ij} can be obtained from existing estimated genetic maps, e.g., it can be downloaded with the panel data from the IMPUTE2 website. The parameter θ uses the value suggested by Li and Stephens [Li and Stephens 2003]:

$$\theta = \frac{\left(\sum_{i=1}^{2m-1} \frac{1}{i}\right)^{-1}}{2m + \left(\sum_{i=1}^{2m-1} \frac{1}{i}\right)^{-1}}.$$

Assuming random mating and the genotype vector coded as the sum of two haplotype vectors, the mean $\boldsymbol{\mu}_g$ and covariance $\boldsymbol{\Sigma}_g$ of the genotype vector are twice of $\boldsymbol{\mu}_h$ and $\boldsymbol{\Sigma}_h$, respectively. Therefore, the subvectors and submatrices $\boldsymbol{\mu}_u$, $\boldsymbol{\mu}_t$, $\boldsymbol{\Sigma}_{tt}$ and $\boldsymbol{\Sigma}_{uu}$ can all be estimated with the external panel $\mathbf{M}$. This illustrates that BLIMP incorporates LD information in the covariance matrices which are estimated from a reference panel. If we further assume the genotype vector follows a multivariate normal distribution approximately, then the best linear predictor is the MMSE estimator and has a normal distribution.

PedBLIMP

Because genotypes among nonfounders in pedigrees are correlated, genotypes of an individual's relatives can provide useful information when imputing the genotypes of the individual. To illustrate with an extreme case: if we know both parents' genotypes are the same and homozygous for a locus, then their children's genotypes must be the same homozygotes, assuming no de novo mutations. Besides LD information, PedBLIMP incorporates the correlation among related individuals into the linear predictor. Instead of considering the genotype vector of a single individual, now we consider the genotype vector including all individuals in a pedigree. Schaid et al.'s derivation of the mean and covariance structure of genotypes in a pedigree is the foundation of extending BLIMP into PedBLIMP [Schaid, et al. 2013]. Suppose we have a randomly sampled pedigree of n individuals. The expected genetic relatedness for a randomly picked locus among all individuals can be characterized by a matrix $\boldsymbol{\omega}$. Each element ω_{ij} in the matrix is the relatedness between individuals i and j , which is twice the kinship coefficient. For outbred pedigrees, it is also the linear correlation coefficient between the genotypes of two individuals at a locus when the genotypes are coded additively. The expected relatedness can be calculated from

available software packages, e.g., the R package kinship2 [Therneau, et al. 2013]. Denote by $\mathbf{G}$ the genotype matrix with additive coding 0, 1 and 2 for the number of the reference allele. Element g_{ik} denotes the genotype of individual i at SNP k . We only consider autosomes here. Assuming random mating and no recombination for a small region, the first and second moments of elements in $\mathbf{G}$ are:

$$\begin{aligned} E(g_{ik}) &= 2p_k \\ \text{Cov}(g_{ik}, g_{jl}) &= \text{Cov}(g_{ik}, g_{il}) \Omega_{ij}, \end{aligned}$$

where p_k is the reference allele frequency in the population, $\text{Cov}(g_{ik}, g_{il})$ is the covariance between SNPs k and l in individual i . Because it is the same for all individuals, we can write it as $\text{Cov}(g_{\cdot k}, g_{\cdot l})$. The above equation for the covariance matrix illustrates how the LD information and the pedigree information are combined together. We employ the same shrinkage method as in BLIMP to estimate p_k and $\text{Cov}(g_{\cdot k}, g_{\cdot l})$ from panel data. To have a compact representation, we stack each column of $\mathbf{G}$ on top of each other to form a vector $\mathbf{G}_{vec}$. Therefore we have estimates as

$$\begin{aligned} \hat{E}(\mathbf{G}_{vec}) &= 2 \times \left[\underbrace{\hat{\mu}_1, \dots, \hat{\mu}_1}_n, \underbrace{\hat{\mu}_2, \dots, \hat{\mu}_2}_n, \dots \right]', \\ \hat{\text{Cov}}(\mathbf{G}_{vec}) &= 2 \times \hat{\Sigma}_h \otimes \Omega, \end{aligned}$$

where $\hat{\mu}_i$ is the i th element in $\hat{\mu}_h$, and $\otimes$ denotes the Kronecker product between two matrices. With these estimated first and second moments, the best linear predictor can be applied for imputing missing genotypes in pedigrees. If we assume a normal distribution of $\mathbf{G}_{vec}$, then matrix $\mathbf{G}$ has a matrix normal distribution. The relatedness matrix with Kronecker product is also used in Price et al.'s work on cross-tissue heritability [Price, et al. 2011].

Even though we focus on imputing genotypes where individuals are genotyped at different sets of markers, i.e., dense and sparse set of markers, in theory PedBLIMP can also be applied to impute genotypes where all individuals are genotyped at the same set of markers. In the latter case, however, we can prove that PedBLIMP and BLIMP are exactly the same, as shown in the Appendix.

Replacing the Expected Relatedness Matrix with the Locus Specific Relatedness Matrix

In PedBLIMP, Ω_{ij} is the expected relatedness for a randomly picked locus, representing a global average value across the whole genome conditional on the pedigree information only. For a specific locus knowing the exact IBD configuration of the four alleles between two individuals can result in a more accurate local relatedness measurement for the locus. Similar to the global expected relatedness, the locus specific relatedness can be defined as twice the fraction of allele pairs that are shared IBD among all four allele pairs between two individuals. Figure 1 shows the difference between the global expected relatedness matrix and the local locus/region specific relatedness matrix. In addition to pedigree structure, the measured genotypes can provide information to gain more accurate estimation of the relatedness for each locus or each small genomic region. To estimate the IBD configurations

for each locus/region, we followed the method used in GIGI [Cheung, et al. 2013]. We first sampled the inheritance vectors (IVs) conditional on the observed genotypes. An inheritance vector of a locus is a vector with the length equal to twice the number of non-founders in a pedigree. Each non-founder has two elements: one for the maternal DNA and the other for the paternal DNA. For each element, it is 0 if the DNA is inherited from the parent's maternal DNA, otherwise it is 1. The program *gl_auto* from the MORGAN package was used for this step [Tong and Thompson 2008] [Thompson 2011]. Then, for each IV sample, we calculated the relatedness for each marker based on the IBD configurations. Denote by b_{kij} the number of allele pairs that are shared IBD for SNP k between individuals i and j . Then the estimated locus specific relatedness based on the IV sample is $\Omega_{kij}^* = 2 \times b_{kij} / 4$. In the following, we refer to the estimated locus specific relatedness as local relatedness. Because sampling IVs requires markers in linkage equilibrium (LE), we needed to select a subset of SNPs as the framework markers. Follow the suggestion for GIGI, we selected a subset from the common SNP set among all individuals in the pedigree and made sure that the distances between SNPs was at least 0.4 cM and the minor allele frequencies (MAF) was at least 0.1. To form a Monte Carlo estimate, once we obtained the local relatedness matrix for each IV sample, we would have needed to perform PedBLIMP with each local relatedness matrix to calculate the predicted dosage scores, and then average all predicted dosage scores. However, in order to save computing time, we first averaged the relatedness matrix corresponding to each IV sample to obtain a single estimate of the relatedness matrix for each SNP, then we ran PedBLIMP only once with the averaged relatedness matrix. We may have sacrificed some accuracy in doing this, but it saved a lot of time and seems to work well in our simulations.

In our implementation, PedBLIMP imputes a small block of SNPs each time, e.g., usually less than 1 Mb. A final tweak is that instead of using different relatedness matrices from one framework marker to another in a small region, we averaged the local relatedness matrices across all framework markers in a small region. The underlying reasoning is that, when the pedigree size is not large, the chance of recombination in a small region will be low; therefore, averaging across all framework markers in a region can improve the stability and accuracy of the local relatedness matrix.

Comparison between GIGI and PedBLIMP without LD

GIGI does not use the LD information between markers. The corresponding scenario in PedBLIMP is that the estimated covariance matrix $\hat{\Sigma}_h$ is a diagonal matrix. In this case we can impute each marker independently as in GIGI. The imputed dosage vector on marker i of all untyped individuals in a pedigree becomes:

$$\hat{g}_u = \pi_i + \Omega_{ut} \Omega_{tt}^{-1} (g_t - \pi_i),$$

where g_t is the genotypes of typed individuals on marker i , π_i is the estimated mean dosage of marker i from the panel, Ω_{ut} is the corresponding submatrix in the relatedness matrix Ω between untyped individuals and typed individuals and Ω_{tt} the submatrix among typed individuals. So PedBLIMP reduces to a very simple linear predictor to impute the missing

genotypes relying on the correlation among related individuals. All the information provided by the inheritance vectors are summarized into the relatedness matrix. On the other hand, GIGI utilizes the inheritance vectors through a full probabilistic model to derive the probabilities of each genotype. When it is safe to ignore LD, GIGI may be a better choice because of the full utilization of the inheritance vectors. In the following simulations, we also compare the results of PedBLIMP with GIGI when the markers are very sparse, which is close to the no LD scenario.

Uncertainty Measure of Imputation

The standard error of each imputed dosage can be calculated from Equation (2) and used as a measure of imputation uncertainty. Besides, in order to output genotype probabilities, such as used in IMPUTE2 and GIGI, we treat the posterior distribution as Gaussian for each imputed dosage with mean and standard error calculated from Equations (1) and (2). The probabilities are then calculated as the areas covering each genotype segment. Specifically, suppose the genotype dosages are coded as 0, 0.5 and 1, the imputed dosage is d with standard error s . Then probabilities of genotype 0, 0.5 and 1 are $\Phi(0.25/s) - \Phi(-0.25/s)$, $\Phi(0.75/s) - \Phi(0.25/s)$ and $1 - \Phi(0.75/s)$, respectively, where Φ is the cumulative function of the normal distribution $N(d, s)$. We note that the above measure is not a calibrated probability measure. Based on simulations, we plotted the estimated probability vs. the actual accuracy similar as in [Browning and Browning 2009]. We found that in general the estimated probabilities tend to be conservative, i.e., the estimated probability is slightly lower than the actual accuracy. In practice, we think it will be a helpful option for users to decide the quality of the final imputation besides the imputed dosage.

Implementation Issues

If $\sum_{ut} \sum_{tt}^{-1}$ in Equation (1) is fixed for all individuals to be imputed, then computing the inverse and storing the resulting matrix will save computing time. For example, in BLIMP, this can be applied when all individuals are genotyped on the same platform so the pattern of untyped and typed SNPs is the same for all individuals. For PedBLIMP, different pedigree structures result in different relatedness matrices. Because the covariance matrix used in the linear predictor is the Kronecker product of the panel covariance matrix and the relatedness matrix, we need to calculate the inverse each time when imputing genotypes on a new pedigree structure. In our implementation, instead of calculating the inverse, we solved the equation $\mathbf{u}\mathbf{x} = (\mathbf{Y}_t - \boldsymbol{\mu}_t)$ and calculated $\hat{\mathbf{Y}}_u = \boldsymbol{\mu}_u + \sum_{ut} \mathbf{x}$. Because $\mathbf{u}\mathbf{u}$ is a symmetric positive definite matrix, Cholesky decomposition can be used to solve the linear system of equations.

Similar to other imputation methods, PedBLIMP imputes SNPs in a chunk each time. For each chunk, we add a buffer region on both sides to avoid an edge effect. However, only the SNPs in the middle chunk are imputed. Because the computing time of PedBLIMP depends on the number of SNPs in $\mathbf{u}\mathbf{u}$ and $\mathbf{u}\mathbf{t}$, we need to set a proper chunk size and buffer size. A too large chunk size will make the computing very slow, and there is little gain from SNPs far away from the SNPs to be imputed. If the chunk size is too small, we might miss useful information from the SNPs not included. We use two quantities to limit the size of the

chunk: the maximal number of SNPs and the maximal length of the chunk, in bases. The size of the chunk should not exceed either of them. Another set of maximal number of SNPs and maximal length is used for the buffer region. Furthermore, we set a threshold on the panel covariance matrix when including SNPs in the buffer region. The shrinkage coefficient $e^{-\rho_{ij}^2/(2m)}$ should not be smaller than the threshold. Sometimes the relatedness matrix $\mathbf{\Omega}$ may be singular. For example, this happens when there are twins in the pedigree. Even without twins, for locus specific relatedness, there may be regions that two individuals are in IBD for both haplotypes. In this case we add a small diagonal matrix $\varepsilon \mathbf{I}$ to $\mathbf{\Omega}$, e.g., in our implementation, we set $\varepsilon=0.001$.

Because the shrunken mean and covariance of the panel are the same for all imputations, once the panel data is decided, we first precompute and store them for use in later imputation. This is also computed by chunks with buffer regions. Therefore, it avoids computing and saving the elements far away from the main diagonal of the full covariance matrix of the panel. We set the panel chunk size larger than the imputation chunk so that the mean and covariance from a panel chunk can be loaded into memory to use for several chunks in imputation.

Data Simulation

The pedigree structure used for simulations is illustrated in Figure 1. There were 10 individuals in the pedigree and genotypes were available only for the third generation. Figure 1 also shows the inheritance patterns represented by the letters inside shapes and the relatedness matrix in our simulation. It illustrates that for a specific region, the global relatedness matrix is usually different from the local relatedness matrix. We assumed no recombination so that it is simpler to compare the performance of PedBLIMP when it used the global, the estimated local, or the true local relatedness matrix.

For the genotype data simulations, we downloaded the 1000 Genome Phase1 haplotypes from IMPUTE2 website (http://mathgen.stats.ox.ac.uk/impute/impute_v2.html#reference). The haplotypes of the 4 founders in the pedigree were randomly sampled from 178 haplotypes of GBR population (British in England and Scotland) and the 170 haplotypes of CEU populations were used as the panel. This design is similar to that of Wen and Stephens where the HapMap CEU population is used as the panel to impute the WTCCC British Cohort [Wen and Stephens 2010]. All evaluations were done using chromosome 22. The SNPs from HapMap release 22 were used as high density SNPs and only a very small set of the SNPs not in the 1000 Genome data was excluded. In total, there were 33,199 SNPs on chromosome 22. This was also the SNP set used in the CEU panel. For individuals that are densely genotyped all SNPs in the reference panel are observed. Because there is at least one densely genotyped individual, all SNPs are partially genotyped in the pedigree. For low density/sparse markers, we chose several marker sets with varying densities. The first set of low density markers was from Affymetrix 5.0 array (downloaded from UCSC Genome Browser website). After excluding SNPs not in 1000 Genome data, it contained 4,504 SNPs. Because the genetic length of chromosome 22 is about 75.12 cM, the average SNP distance, defined as the total chromosome length divided by the number of SNPs, was about 0.0167 cM per SNP. For the rest of the marker sets, we randomly sampled a subset of SNPs from

the first low density marker set by setting the average SNP distance to 0.03, 0.05, 0.07, 0.09, 0.1, 0.2, 0.3, 0.4, and 0.5 cM, in total 10 different sparse levels of low density markers. The average SNP distance and the corresponding SNP numbers are listed in Table 1. For each data set, we imputed all the SNPs in the panel that were not observed in pedigree data, i.e., the missing genotypes in the two individuals with low density markers. To evaluate the performance of different methods, we simulated 20 pedigrees and aggregated the comparison results.

Compared Methods of Imputation

We compared PedBLIMP with BLIMP, IMPUTE2, SHAPEIT2 plus IMPUTE2 and GIGI. The following describes the settings for each method.

For PedBLIMP, the maximal number of SNPs in a chunk to be imputed was 100, and maximal length was 0.5 Mb. The maximal number of SNPs in the buffer region on both sides was also 100 with a maximal length of 0.5 Mb. The threshold of the shrinkage coefficient on the panel covariance matrix for buffer SNPs was 10^{-4} , which was usually satisfied given the chunk and buffer constraints. Therefore, the maximal number of SNPs involved in an imputation was 300 for a chunk. We pre-computed the mean and covariance matrix from the panel data using a chunk size of 300 SNPs and stored them in files before imputation. The haplotypes from CEU population was used as the panel. The effective population size was set to 11,418. The genetic map is linearly interpolated for every SNP in the panel so that the genetic distance can be calculated for any pair of SNPs, which is required for calculating the shrunken covariance matrix. We sampled the inheritance vectors using *gl_auto*. 6,000 MCMC iterations were used and a sampled inheritance vector was output for every 30 iterations. Three versions of PedBLIMP are compared: PedBLIMP with the global relatedness derived from the pedigree structure, denoted by PedBLIMP_{global}, PedBLIMP with the local relatedness from estimated IBD, denoted by PedBLIMP_{local} and PedBLIMP with the true local relatedness from true IBD states, denoted by PedBLIMP_{true}.

We also implemented our own version of BLIMP for comparison. We set the maximal length of an imputed chunk to 0.5 Mb. For the buffer size, we let the threshold of the shrinkage coefficient on the panel covariance matrix to control the number of SNPs involved in the computation. We tried both 10^{-4} and 10^{-8} . The results were very similar, so we used 10^{-4} in our comparison.

For IMPUTE2 (v2.3.0), instead of supplying the genotypes of all four individuals in the pedigree to the program, we put the two individuals with dense SNPs (no missing SNPs) into a second unphased genotype panel. We found that this could often achieve a slightly better result. The same panel data was used as in PedBLIMP. We set the number of haplotypes to use as templates when phasing observed genotypes (the option *-k*) to 100 and the buffer size to 1,000 kb (the option *-buffer*). The effective population size was set to 11,418. Other options used default values. The chunk size for imputation was 5 Mb, except the last one, which was less than 5 Mb.

SHAPEIT2 (v2.r727) assumes similar densities among individuals. Otherwise it can slow down computations and is likely to be error-prone. Therefore the combination of both high

and low density SNPs among pedigree members considered in this study is not appropriate for SHAPEIT2. However, we still tried to use SHAPEIT2 for pre-phasing and then use IMPUTE2 for imputing the pre-phased haplotypes. We only fed the low density SNPs from all individuals to SHAPEIT2. Then we fed the most likely pairs of haplotypes from individuals with low density SNPs to IMPUTE2. Here obviously we did not use the information provided by the individuals with dense markers, which might result in suboptimal results. We used the same panel data as in PedBLIMP. The effective population size was set to 11,418 and all other options used default values. The chunks size for IMPUTE2 was the same as that used for IMPUTE2 above. Note that pedigree information cannot be used by SHAPEIT2 because of no genotypes for parents.

GIGI (v1.04.1) used the same sampled inheritance vectors as in PedBLIMP. The allele frequencies were estimated from the CEU panel. The genetic map of SNPs was also the same as used in PedBLIMP.

Evaluation Criteria

We used the fraction of errors (or error rate) and mean squared error (MSE) to measure the performance of imputation for all imputed SNPs. The fraction of errors is the fraction of incorrectly imputed genotypes. Calculating the fraction of errors requires the imputed genotypes instead of dosage values. Therefore, for BLIMP and PedBLIMP, we rounded the imputed dosages to the nearest genotypes as previously suggested for BLIMP. For GIGI and

IMPUTE2, we used the posterior mode. The MSE is defined as $MSE = \frac{1}{N} \sum_{k=1}^N (g_k - \hat{g}_k)^2$, where g_k is the true genotype dosage value and $\hat{g}_k$ is the imputed dosage value, N is the total number of imputed SNPs.

We also compared the performance on rare variants. We defined rare variants as those SNPs with minor allele frequencies (MAF) less than 0.05. MAF was estimated from all European populations in the 1,000 Genome Phase 1 data. There were SNPs with MAF = 0 for the European population in the data because these SNPs may have nonzero MAFs for other populations. Because imputing SNPs with MAF = 0 is straightforward, we excluded these SNPs in evaluation. First, we used the fraction of errors to evaluate the performance on all rare variants. For rare variants, almost all genotypes are either heterozygous or homozygous with two common alleles. The proportion of homozygous common genotypes is much larger than the heterozygous genotypes. Because of this imbalance, the total fraction of errors is largely influenced by the homozygous common genotypes. In order to gain more insight on the performance, we assessed another two separate fraction of errors for heterozygous and homozygous common genotypes, similar to those used in [Howie, et al. 2009]. The first is the fraction of errors in predicting heterozygous genotypes as homozygous common genotypes, and the second is the fraction of errors in predicting homozygous common genotypes as heterozygous genotypes.

Results

Performance on all SNPs

Figure 2 shows the overall fraction of errors on all imputed SNPs. Figure 3 shows the overall MSE on all imputed SNPs. The patterns of these two error measurements are similar. Both figures show that methods using LD information when the density of markers is relatively high have better results than GIGI. The fraction of errors and MSE of all methods, except GIGI, increased as the average SNP distance increased. This is reasonable, because all methods, except GIGI, incorporate LD information, and the larger the average SNP distance, the weaker the LD and the less information the observed SNPs can provide in predicting the missing SNPs. All three versions of PedBLIMP had much better performance than BLIMP due to incorporating extra information from markers of related individuals. PedBLIMP_{local} had further improved performance than PedBLIMP_{global}. The results of PedBLIMP_{local} were very close to PedBLIMP_{true}. This shows the importance of estimating the local relatedness matrix in PedBLIMP.

In the following, we refer to PedBLIMP_{local} as PedBLIMP if no specific subscript is used. When the average distance between SNPs was small, e.g., less than 0.1 cM, IMPUTE2 had slightly lower error rate than PedBLIMP, the difference in error rate was about 0.01. However, when the average distance was larger than 0.1 cM, PedBLIMP had less error than IMPUTE2; the difference in error rate was over 0.04. There may be two reasons why IMPUTE2 has a slightly better result than PedBLIMP when the density of markers is high. First, IMPUTE2 has a more complex model than PedBLIMP's linear predictor. Second, the phasing update step in IMPUTE2 uses both the reference panel and all study samples in a joint model, which implicitly accounts for the relatedness among individuals. However, when the density of markers became lower, especially when LD became very weak, explicit modeling of the relatedness by PedBLIMP showed its strength, even with the simple linear predictors. Compared to GIGI, PedBLIMP had much lower error, when the distance between markers was less than 0.4 cM. As the markers became sparser, the performance of PedBLIMP became closer to that of GIGI. This is reasonable, because there was little LD information to be used and the main information was from correlated individuals. SHAPEIT2 + IMPUTE2 had similar or slightly lower error than BLIMP, but higher than PedBLIMP and IMPUTE2. This is because SHAPEIT2 + IMPUTE2 does not fully use the relatedness information as explained before.

Performance on Rare Variants

Figure 4 shows the total fraction of errors on rare variants ($0 < \text{MAF} < 0.05$). Again, PedBLIMP showed much better performance than BLIMP. When the average SNP distance was small, PedBLIMP and IMPUTE2 had similar error rates, both much less than GIGI. When the average SNP distance was larger than 0.05 cM, PedBLIMP showed lower error rates than IMPUTE2. As the markers became sparser, the error rate of PedBLIMP became closer to GIGI. IMPUTE2's error rate surpassed GIGI when the average SNP distance was about 0.1 cM. It also shows that SHAPEIT2+IMPUTE2 had similar or slightly lower error than BLIMP, but higher than PedBLIMP and IMPUTE2.

Figure 5 shows the fraction of errors on rare variants when heterozygous genotypes were predicted as homozygous common. It shows that PedBLIMP had much better performance than BLIMP. When the average SNP distance was at least 0.07 cM, PedBLIMP achieved lower error rate than IMPUTE2. When the distance was over 0.4 cM, the error rate of IMPUTE2 could be over 90%, while the error rates of PedBLIMP and GIGI were around 50%. PedBLIMP had much better performance than GIGI when the average distance between SNPs was less than 0.4 cM. As the markers became even sparser, the performance of PedBLIMP became closer to GIGI.

Figure 6 shows the fraction of errors on rare variants when homozygous common genotypes were predicted as heterozygous. All methods except PedBLIMP_{global} had very low error rates (< 0.005). GIGI achieved the lowest error rate. PedBLIMP had slightly lower error rate than IMPUTE2, except the last point when the average SNP distance was 0.5 cM. For the last point, we also see a slight difference between PedBLIMP_{true} and PedBLIMP_{local}, in both Figure 4 and Figure 6. This is because at very sparse levels, the selected framework markers used to sample inheritance vectors become so sparse (about 60 SNPs on ~75 cM region) that the sampled IVs are not good enough, and substantially reduce the quality of the local relatedness matrix for SNPs in each chunk.

In summary, from our simulation results we found that PedBLIMP had much better performance than BLIMP. This is because PedBLIMP uses extra information from related individuals while BLIMP assumes individuals are unrelated. When the average SNP distance was in the range of 0.09 – 0.3 cM, PedBLIMP had the most advantage over both IMPUTE2 and GIGI for imputing rare variants. Over all HapMap SNPs, PedBLIMP showed better results in the range of 0.1 – 0.5 cM. The advantage of PedBLIMP over IMPUTE2 is more obvious when imputing rare variants. This is because LD is generally weak for rare variants and PedBLIMP explicitly incorporates relatedness information. When the average distance between SNPs is over 0.4 cM, the performance of PedBLIMP becomes closer to GIGI. Under this scenario, GIGI may be a better option, because incorporating LD information brings little gain and GIGI is fast once the IVs are sampled.

Time and memory usage

We first show the theoretic time and space complexity and then report the actual time and memory usage. Suppose there are n_h individuals with high density markers and n_l individuals with low density markers. The number of SNPs in a chunk is c and the buffer size is also c , so in total there are $3c$ SNPs involved. Let r be the missing ratio, i.e., 1 minus the ratio between the number of low density markers and the number of high density markers. Then the number of untyped markers is $n_u = cn_l r$ and the number of typed markers $n_t = 3c(n_h + n_l(1 - r))$. In Equation (1), the main time-consuming part is solving the linear

equations, which is $O\left(\frac{1}{3}n_t^3\right)$ using Cholesky decomposition. Therefore, the time complexity for a chunk is $O(9c^3(n_h + n_l(1 - r))^3)$. Suppose the total number of SNPs to impute is L . Then the time complexity is $O(9c^2L(n_h + n_l(1 - r))^3)$. The complexity of

calculating the standard error is $O(n_t^3 n_u)$ for a chunk and $O(9c^2L(n_h + n_l(1 - r))^2 n_l r)$ for L SNPs. The main memory usage is due to the covariance matrix, $\mathbf{u}_t$, between the untyped

and typed markers and the covariance matrix, $\mathbf{M}$, of the typed markers. Specifically, the total number of elements is $n_u n_t + n_t^2 = 9c^2 (n_h + n_l (1 - r)) \left(n_h + n_l \left(1 - \frac{2}{3}r \right) \right)$. The time cost is roughly proportional to the cubic of the pedigree size, and the memory roughly proportional to the square of the pedigree size. Both time and memory cost are proportional to the square of the number of SNPs in the imputation chunk. Even though the complexity is polynomial with respect to the pedigree size, PedBLIMP may be more suitable for small to moderate size pedigrees.

Table 2 shows the actual time and memory usage of PedBLIMP, IMPUTE2 and GIGI in our simulations. For the current implementation of PedBLIMP in R, we assume that we have calculated and stored the mean and covariance matrix of the reference panel for each chunk, because once calculated, these can be used for all pedigrees in imputations. The time of sampling the inheritance vectors is not included, but listed separately in the last row because PedBLIMP can optionally avoid this step and use the global relatedness matrix with sacrificed accuracy. PedBLIMP had similar running time as IMPUTE2 on simulated data. GIGI was faster than PedBLIMP, SHAPEIT2 + IMPUTE2 was the fastest. The time and memory usage of PedBLIMP and GIGI were stable for different average SNP distances. The time of IMPUTE2 and SHAPEIT2 decreased as the number of observed SNPs decreased. The speed of PedBLIMP could potentially be improved when implemented in languages, such as C++.

Discussion

For genome-wide association studies, individuals are usually genotyped with similar marker densities. In contrast, for pedigrees, it can be cost efficient to mix marker densities, where some individuals are genotyped with high density markers and other relatives are genotyped with low density markers, relying on imputation to fill in the untyped markers. We proposed an algorithm, PedBLIMP, an extension of BLIMP, to impute genotypes in pedigrees for the mixed density design. From our simulations, we found that when the average SNP distance for sparse markers was in the range of 0.1 to 0.4 cM, PedBLIMP generally showed better performance than compared methods, such as IMPUTE2 and GIGI, on both common and rare SNPs. For rare variants, even when the marker density is relatively high, LD can still be weak and PedBLIMP may achieve better performance than IMPUTE2. In general, the improvement of PedBLIMP compared to IMPUTE2 depends on the fraction of markers that can be estimated using IBD but not LD. When the fraction is high, more information can be used by explicitly incorporating the IBD information. Even though IMPUTE2 implicitly incorporates relatedness information, when the marker density decreases, it is difficult for IMPUTE2 to fully exploit this information.

Because the extra information comes from related individuals in the pedigree, the imputation results will also depend on the design of genotyping: which individuals are genotyped with the high density markers and which with the low density markers. The more individuals with high density markers, the lower the error in imputation. This is true for PedBLIMP, IMPUTE2 and GIGI. The final design may be a tradeoff between the budget and imputation accuracy. A recent method proposed a good solution especially for rare variants [Cheung, et

al. 2014]. Another consideration is the downstream analysis. Chi et al. [Chi, et al. 2013] showed that even though the imputation accuracy of different methods are substantially different, when applied to association studies, the loss in power of the inferior methods are still tolerable. Therefore, influence of imputation accuracy on the downstream analysis may also be a factor to consider in genotyping design.

If the missing genotype pattern is the same for all individuals in a pedigree, PedBLIMP and BLIMP are identical (see Appendix). This appears counter intuitive. However, we can interpret it in the following way. For the missing SNP set of an individual, other individuals also have complete missing values on the same set. Given the observed SNPs in an individual, the other individuals do not contribute any extra information.

We used a very simple method to automatically select framework markers from a SNP set, and it seemed to work well. Table 1 lists the number of selected framework markers to sample inheritance vectors from one data set. We found that the number of selected markers to sample IVs does not matter until it is too low. For example, when the average SNP distance is 0.5 cM, the difference between PedBLIMP_{true} and PedBLIMP_{local} becomes obvious in Figure 4 and Figure 6. This may not be a problem in real applications because in our simulation the low density SNPs, from which the framework markers were selected, were randomly sampled. In real data, the low density markers are usually carefully selected, e.g., with relatively high MAF, so that they can provide more information in estimating inheritance vectors. In our simulation, we chose 6,000 MCMC iterations and output an IV sample every 30 iterations. This seems enough for our simulated pedigree. However, when the pedigree size becomes large, we expect the number of MCMC iterations will need to increase due to potentially higher uncertainty. For example, Cheung et al. [Cheung, et al. 2013] used 30,000 MCMC iterations for a large pedigree. The time cost of sampling IVs will increase accordingly.

For each imputation chunk, when estimating the local relatedness matrix, we chose to average the relatedness matrix across framework markers in the chunk. This may be reasonable if the pedigree size is small. When the pedigree size is large, recombination may not be negligible. Therefore, using a distinct relatedness matrix for each framework marker may be a better option. In this case, the relatedness matrix for other SNPs can be chosen from the closest framework marker. Furthermore, to improve stability and accuracy of the local relatedness matrix, more MCMC iterations may be needed.

As can be seen from Equation (1), each imputed dosage value, i.e., each element in $\hat{\mathbf{Y}}_u$ only depends on the observed genotypes and the covariance matrix. There is no explicit constraint among elements in the predicted vector $\hat{\mathbf{Y}}_u$. There may be potential violation of Mendelian rules in the imputed dosages from PedBLIMP. This is also true for BLIMP and IMPUTE2 used on pedigree data. However, the portion of imputed SNPs from PedBLIMP with violations is small based on our simulations. Because GIGI uses the inheritance vector directly to estimate genotypes, no Mendelian errors were found in the simulations.

Using information from LD and using information from pedigree relatedness may result in inconsistency. For example, IMPUTE2 and GIGI may give inconsistent imputed genotypes.

It seems difficult to completely separate the two sources of information mathematically in the linear predictor of PedBLIMP. However, we can think of ways to control the influence of these two sources through adjusting the corresponding correlation coefficients. In the extreme case, if we set the correlation coefficient from LD to 0, then we are using only the IBD information. If we set the correlation coefficient from IBD to 0, then we are using only the LD information as in BLIMP. The quality of the reference panel and samples may influence the adjustment. This can be a direction for future research.

In our genotyping design, SHAPEIT2+IMPUE2 did not perform as well as IMPUTE2. This is different from the performance on GWAS data where the set of measured genotypes are almost the same for individuals [Delaneau, et al. 2013]. How the model used in SHAPEIT2 or IMPUTE2 can be adapted in the mixed density design is an interesting question. When genotypes are available for all individuals in a pedigree, a new version of SHAPEIT2 [O'Connell, et al. 2014] can make use of the pedigree information by splitting a pedigree into duos and trios to improve phasing accuracy.

There may be other ways to combine both the pedigree/IBD information and LD information when imputing genotypes in pedigrees. One possible way is to fuse the methods for related individuals and methods for unrelated individuals. For example, one could use GIGI to utilize the pedigree information to impute genotypes first and accept those with very high confidence. For those genotypes that are still missing, IMPUTE2 could be used for further imputation leveraging LD information from an external panel. Furthermore, using LD to improve estimation of IBD [Han and Abney 2011] might prove beneficial.

Distant or cryptic relatedness may happen among individuals in GWAS samples. The distant relatedness information can be taken into account in PedBLIMP by calculating the local relatedness among individuals, provided that a good pairwise IBD estimation for each region from the genotype data without explicit pedigrees is available. Some existing software, such as fastIBD [Browning and Browning 2011], may provide this information. The gain of doing this may be small because the fraction of genome IBD is usually very small and the markers genotyped are usually the same for everyone. Currently PedBLIMP assumes the same LD structure among pedigree members and it may not work very well for a pedigree consisting of different subpopulations or admixed samples. In this case different LD structures due to different ancestral populations need to be considered.

Acknowledgments

This research was supported by the U.S. Public Health Service, National Institutes of Health, contract grant number GM065450.

Appendix: Proof of equivalence between BLIMP and PedBLIMP when all individuals have the same missing pattern

Denote by n the number of individuals in the pedigree. For individual $i = 1, \dots, n$, BLIMP estimates the dosage vector $\mathbf{Y}_u^i$ as

$$\hat{Y}_u^i = \mu_u^1 + \Sigma_{ut} \Sigma_{tt}^{-1} (Y_t^i - \mu_t^1), \quad (\text{A.1})$$

$$\text{Cov}(\hat{Y}_u^i) = \Sigma_{uu} - \Sigma_{ut} \Sigma_{tt}^{-1} \Sigma_{tu}, \quad (\text{A.2})$$

where μ_u^1 is the mean for untyped vector Y_u^i , μ_t^1 is the mean for the typed vector Y_t^i . The superscript **1** implies that the mean is the same for all individuals. Σ_{ut} and Σ_{tu} are the covariance matrices between Y_u^i and Y_t^i , Σ_{uu} and Σ_{tt} are the covariance matrices of Y_u^i and Y_t^i .

In PedBLIMP, we put all untyped and typed markers into vectors with the following order:

$$Y_u = [Y_u^1, \dots, Y_u^n]', Y_t = [Y_t^1, \dots, Y_t^n]', \mu_u = [\mu_u^1, \dots, \mu_u^n]', \mu_t = [\mu_t^1, \dots, \mu_t^n]',$$

The corresponding covariance matrices between Y_u and Y_t are $\Gamma_{ut} = \Omega \otimes \Sigma_{ut}$, and $\Gamma_{tu} = \Omega \otimes \Sigma_{tu}$. The covariance matrix of Y_t is $\Gamma_{tt} = \Omega \otimes \Sigma_{tt}$, and the covariance matrix of Y_u is $\Gamma_{uu} = \Omega \otimes \Sigma_{uu}$.

PedBLIMP estimates Y_u as:

$$\hat{Y}_u = \mu_u + \Gamma_{ut} \Gamma_{tt}^{-1} (Y_t - \mu_t) = \mu_u + (\Omega \otimes \Sigma_{ut}) (\Omega \otimes \Sigma_{tt})^{-1} (Y_t - \mu_t)$$

Because $(\Omega \otimes \Sigma_{tt})^{-1} = \Omega^{-1} \otimes \Sigma_{tt}^{-1}$, we have

$$\hat{Y}_u = \mu_u + (\Omega \otimes \Sigma_{ut}) (\Omega^{-1} \otimes \Sigma_{tt}^{-1}) (Y_t - \mu_t) = \mu_u + ((\Omega \Omega^{-1}) \otimes (\Sigma_{ut} \Sigma_{tt}^{-1})) (Y_t - \mu_t),$$

Because the relatedness matrix times its inverse is the identity matrix,

$$\hat{Y}_u = \mu_u + (I_n \otimes (\Sigma_{ut} \Sigma_{tt}^{-1})) (Y_t - \mu_t), \quad (\text{A.3})$$

Expanding the Kronecker product in A.3, it can be seen that the rows corresponding to each block of one individual in A.3 is equivalent to A.1.

For the covariance matrix of the estimation, we have

$$\text{Cov}(\hat{Y}_u) = \Gamma_{uu} - \Gamma_{ut} \Gamma_{tt}^{-1} \Gamma_{tu} = \Omega \otimes \Sigma_{uu} - (I_n \otimes (\Sigma_{ut} \Sigma_{tt}^{-1})) (\Omega \otimes \Sigma_{tu}),$$

This can be simplified to

$$\text{Cov}(\hat{Y}_u) = \Omega \otimes \Sigma_{uu} - \left(\Omega \otimes \left(\Sigma_{ut} \Sigma_{tt}^{-1} \Sigma_{tu} \right) \right) = \Omega \otimes \left(\Sigma_{uu} - \Sigma_{ut} \Sigma_{tt}^{-1} \Sigma_{tu} \right). \quad (\text{A.4})$$

Because the diagonal of Ω is 1 assuming an outbred pedigree, the covariance matrix for estimated dosages within each individual in A.4 is the same as in A.2.

References

- Abecasis GR, Cherny SS, Cookson WO, Cardon LR. Merlin--rapid analysis of dense genetic maps using sparse gene flow trees. *Nat Genet.* 2002; 30(1):97–101. [PubMed: 11731797]
- Browning BL, Browning SR. A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *Am J Hum Genet.* 2009; 84(2):210–23. [PubMed: 19200528]
- Browning BL, Browning SR. A fast, powerful method for detecting identity by descent. *Am J Hum Genet.* 2011; 88(2):173–82. [PubMed: 21310274]
- Browning BL, Yu Z. Simultaneous genotype calling and haplotype phasing improves genotype accuracy and reduces false-positive associations for genome-wide association studies. *Am J Hum Genet.* 2009; 85(6):847–61. [PubMed: 19931040]
- Browning SR, Browning BL. Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *Am J Hum Genet.* 2007; 81(5):1084–97. [PubMed: 17924348]
- Burdick JT, Chen WM, Abecasis GR, Cheung VG. In silico method for inferring genotypes in pedigrees. *Nat Genet.* 2006; 38(9):1002–4. [PubMed: 16921375]
- Chen WM, Abecasis GR. Family-based association tests for genomewide association scans. *Am J Hum Genet.* 2007; 81(5):913–26. [PubMed: 17924335]
- Cheung CY, Marchani Blue E, Wijsman EM. A statistical framework to guide sequencing choices in pedigrees. *Am J Hum Genet.* 2014; 94(2):257–67. [PubMed: 24507777]
- Cheung CY, Thompson EA, Wijsman EM. GIGI: an approach to effective imputation of dense genotypes on large pedigrees. *Am J Hum Genet.* 2013; 92(4):504–16. [PubMed: 23561844]
- Chi EC, Zhou H, Chen GK, Del Vecchio DO, Lange K. Genotype imputation via matrix completion. *Genome Res.* 2013; 23(3):509–18. [PubMed: 23233546]
- Daetwyler HD, Wiggans GR, Hayes BJ, Woolliams JA, Goddard ME. Imputation of missing genotypes from sparse to high density using long-range phasing. *Genetics.* 2011; 189(1):317–27. [PubMed: 21705746]
- Delaneau O, Marchini J, Zagury JF. A linear complexity phasing method for thousands of genomes. *Nat Methods.* 2012; 9(2):179–81. [PubMed: 22138821]
- Delaneau O, Zagury JF, Marchini J. Improved whole-chromosome phasing for disease and population genetic studies. *Nat Methods.* 2013; 10(1):5–6. [PubMed: 23269371]
- Elston RC, Stewart J. A general model for the genetic analysis of pedigree data. *Hum Hered.* 1971; 21(6):523–42. [PubMed: 5149961]
- Han L, Abney M. Identity by descent estimation with dense genome-wide genotype data. *Genet Epidemiol.* 2011; 35(6):557–67. [PubMed: 21769932]
- Howie BN, Donnelly P, Marchini J. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS Genet.* 2009; 5(6):e1000529. [PubMed: 19543373]
- Kong A, Masson G, Frigge ML, Gylfason A, Zusmanovich P, Thorleifsson G, Olason PI, Ingason A, Steinberg S, Rafnar T. Detection of sharing by descent, long-range phasing and haplotype imputation. *Nat Genet.* 2008; 40(9):1068–75. others. [PubMed: 19165921]
- Lander ES, Green P. Construction of multilocus genetic linkage maps in humans. *Proc Natl Acad Sci U S A.* 1987; 84(8):2363–7. [PubMed: 3470801]
- Li N, Stephens M. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics.* 2003; 165(4):2213–33. [PubMed: 14704198]

- Li Y, Willer C, Sanna S, Abecasis G. Genotype imputation. *Annu Rev Genomics Hum Genet.* 2009; 10:387–406. [PubMed: 19715440]
- Li Y, Willer CJ, Ding J, Scheet P, Abecasis GR. MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genet Epidemiol.* 2010; 34(8):816–34. [PubMed: 21058334]
- Marchini J, Howie B. Genotype imputation for genome-wide association studies. *Nat Rev Genet.* 2010; 11(7):499–511. [PubMed: 20517342]
- Marchini J, Howie B, Myers S, McVean G, Donnelly P. A new multipoint method for genome-wide association studies by imputation of genotypes. *Nat Genet.* 2007; 39(7):906–13. [PubMed: 17572673]
- O’Connell J, Gurdasani D, Delaneau O, Pirastu N, Ulivi S, Cocca M, Traglia M, Huang J, Huffman JE, Rudan I. A general approach for haplotype phasing across the full spectrum of relatedness. *PLoS Genet.* 2014; 10(4):e1004234. others. [PubMed: 24743097]
- Price AL, Helgason A, Thorleifsson G, McCarroll SA, Kong A, Stefansson K. Single-tissue and cross-tissue heritability of gene expression via identity-by-descent in related or unrelated individuals. *PLoS Genet.* 2011; 7(2):e1001317. [PubMed: 21383966]
- Schaid DJ, McDonnell SK, Sinnwell JP, Thibodeau SN. Multiple genetic variant association testing by collapsing and kernel methods with pedigree or population structured data. *Genet Epidemiol.* 2013; 37(5):409–18. [PubMed: 23650101]
- Scheet P, Stephens M. A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *Am J Hum Genet.* 2006; 78(4):629–44. [PubMed: 16532393]
- Stephens M, Scheet P. Accounting for decay of linkage disequilibrium in haplotype inference and missing-data imputation. *Am J Hum Genet.* 2005; 76(3):449–62. [PubMed: 15700229]
- Therneau, T.; Atkinson, E.; Sinnwell, J.; Schaid, D.; McDonnell, S. kinship2: Pedigree functions. R package version 1.5.4. 2013.
- Thompson E. The structure of genetic linkage data: from LIPED to 1M SNPs. *Hum Hered.* 2011; 71(2):86–96. [PubMed: 21734399]
- Tong L, Thompson E. Multilocus lod scores in large pedigrees: combination of exact and approximate calculations. *Hum Hered.* 2008; 65(3):142–53. [PubMed: 17934317]
- Wen X, Stephens M. Using Linear Predictors to Impute Allele Frequencies from Summary or Pooled Genotype Data. *Ann Appl Stat.* 2010; 4(3):1158–1182. [PubMed: 21479081]

Figure 1.

(a) Simulated pedigrees. The solid red shape indicates individuals with high density SNPs; the blue shape with gradient fill indicates individuals with low density SNPs; empty shape indicates individuals without genotypes. The letters in shapes represent the flow of a specific chromosome region with the paternal chromosome on the left and the maternal chromosome on the right. (b) Global relatedness for the whole genome (twice the kinship matrix) and the local relatedness for a specific region. Rows and columns of the relatedness matrix correspond to the four numbered individuals in the third generation. The red font shows the differences between the two matrices.

Figure 2.

Fraction of errors on all imputed SNPs. BLIMP and PedBLIMP round the imputed dosages to the nearest genotypes. GIGI and IMPUTE2 use the posterior mode. The average SNP distance is the chromosome length divided by the number of SNPs.

Figure 3.

Mean squared error (MSE) on all imputed SNPs. BLIMP and PedBLIMP use the imputed dosages. GIGI and IMPUTE2 use the posterior mean. The average SNP distance is the chromosome length divided by the number of SNPs.

Figure 4.

Fraction of errors on rare variants ($0 < \text{MAF} < 0.05$). This error includes both predicting heterozygous genotypes as homozygous common and predicting homozygous common as heterozygous genotypes. BLIMP and PedBLIMP round the imputed dosages to the nearest genotypes. GIGI and IMPUTE2 use the posterior mode. The average SNP distance is the chromosome length divided by the number of SNPs.

Figure 5.

Fraction of errors in predicting heterozygous genotypes ($0 < \text{MAF} < 0.05$) as homozygous common genotypes. BLIMP and PedBLIMP round the imputed dosages to the nearest genotypes. GIGI and IMPUTE2 use the posterior mode. The average SNP distance is the chromosome length divided by the number of SNPs.

Figure 6.

Fraction of errors in predicting homozygous common genotypes ($0 < \text{MAF} < 0.05$) as heterozygous genotypes. BLIMP and PedBLIMP round the imputed dosages to the nearest genotypes. GIGI and IMPUTE2 use the posterior mode. The average SNP distance is the chromosome length divided by the number of SNPs.

Table 1

Average SNP distances and the number of SNPs.

Distance (cM)	0.0167	0.03	0.05	0.07	0.09	0.1	0.2	0.3	0.4	0.5
#SNP	4504	2503	1502	1073	834	751	375	250	187	150
#IV marker ^a	124	123	117	114	110	108	89	75	61	59

^aThe number of markers selected to infer inheritance vectors (IVs) from one pedigree data set. The total length of chromosome 22 is about 75.12 cM.

Table 2

Time and memory usage on the simulated data with different average SNP distances

Methods	0.0167 cM ^a		0.5 cM	
	Time ^b	Memory ^c	Time	Memory
PedBLIMP (estimated relatedness)	~ 205 s	~ 291 M	~ 190 s	~ 291 M
IMPUTE2 (two panels)	~ 236 s	~ 331 M	~ 176 s	~ 314 M
GIGI	~ 104 s	~ 2.3 M	~ 104 s	~ 2.3 M
SHAPEIT2 + IMPUTE2	~ 17 s	< 35 M	~ 8 s	< 35 M
Inheritance vector sampling ^d	~ 13 s	< 20 M	~ 6 s	< 20 M

^a Average SNP distances calculated as the total genetic distance divided by the number of observed SNPs.

^b Time measured in seconds.

^c Peak memory measured in megabytes.

^d Inheritance vector sampling required for both GIGI and PedBLIMP with estimated relatedness matrix; 6,000 MCMC iterations were used and a sampled inheritance vector was output for every 30 iterations. The evaluation was done on a laptop with a 2.4 GHz Intel Core i5 with 16 gigabytes memory and only one core was used in executing programs.