University of Alberta

MISSING SNP GENOTYPE IMPUTATION

by

Yining Wang

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

Fall 2011
Edmonton, Alberta

Abstract

High-throughput single nucleotide polymorphism (SNP) genotyping technologies conveniently produce large SNP genotype datasets for genome-wide linkage and association studies. Various factors, from array design and hybridization, can give rise to a certain percentage of missing calls, and the problem becomes severe when the target organisms such as cattle do not have a high resolution genomic sequence available. Missing calls in SNP genotype datasets would undermine downstream data analysis. Therefore, effective methodologies for dealing with missing genotypes are in urgent need. In this dissertation, we start with a brief introduction to the concepts in genetics, then present a collection of imputation methods, with focus on machine learning algorithms, to tackle the missing SNP genotype problem. We demonstrate that these imputation approaches can achieve satisfactory accuracies, tested on the real population SNP genotype datasets, and highlight the places where our new methods find useful. We conclude with some possible future directions for the genome-wide SNP genotype imputation problem.

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Guohui Lin, for his support, encouragement, and guidance throughout my master studies.

I would also like to thank my examining committee members, Dr. Russ Greiner and Dr. Changxi Li, for reading my thesis, and providing insightful suggestions and valuable feedback.

Finally, my great thanks go to my parents for their constant love and support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 High-throughput Genotyping Technology

A single nucleotide polymorphism (SNP) represents the most common form of genetic variations in the genome between individuals of the same species. Such a genetic variation involves alterations of one nucleotide at a physical location, and is considered to be a SNP only if it occurs in at least 1% of the population. Because of their abundance, heredity stability, and availability of high-throughput genotyping technologies [34], SNPs have been used as genetic markers to facilitate the new-generation genetic tool for constructing the high density genetic map [10] and carrying out the *genome-wide association studies* (GWAS), which aim at identifying genetic associations with traits from these common genetic variations. Thus far, SNPs have been recognized to be the etiology of many complex diseases such as prostate cancer, bipolar disorder, and obesity [5, 11, 24].

In general, GWAS, either case-control, categorical or quantitative, are based on the *"common disease/trait-common variant"* (CDCV) hypothesis [1], and require many samples along with large and dense SNP markers. The tools for scanning millions of SNPs for each sample to detect the polymorphisms are DNA microarrays. For diploid species such as human and cattle, the high throughput genotyping technologies utilize DNA microarrays together with the information on the distribution of SNPs along the genomes to generate unphased genotype for each SNP marker. With the completion of the International HapMap Project (Phase I) [10], a haplotype map for understanding genetic variants and the haplotype structures of humans has become available; as a result, a set of tag SNPs has been identified and can be used as the reference to the distribution of SNPs throughout human genomes.

Currently, two competing high-throughput genotyping platforms from the Affymetrix GeneChip and the Illumina BeadChip respectively are the two popular choices for whole-genome genotyping. Both platforms are single-channel microarray systems and contain a selection of variant probes along the genome [33, 24]. The major difference between the two platforms lies in their SNP-selection strategies in term of genome coverage [24]. Microscopic SNP probes on the Illumina array are selected almost entirely from the tag SNPs with optimal minor allele frequency derived from the international HapMap project; on the other hand, Affymetrix array includes about half of those tag SNPs and the rest are mainly from an unbiased selection of SNPs [33, 24].

1.2 SNP Genotype Missing Value Problem

High-density SNP microarray chips can produce unphased genotype values for each SNP marker. However, due to the current design of high-throughput genotyping technology, certain amount of bias, known as genotyping errors, could be introduced to the process of selection and amplification [24]. In addition, the genotyping assays are prone to contain missing calls. Missing calls can be attributed to the poor quality of DNA samples and the ambiguity of fluorescence signals [17]. Poor quality of DNA samples can result in the failure of amplification and reduction of intensity of subsequent fluorescence signals on the background. The ambiguity in the reads of fluorescence signals can lead to "no-call" procedure that any of the clusters of genotype cannot be assigned to those signals [17]. For humans, the current general-purpose high-density SNP chips are estimated to contain a portion of missing genotypes and genotyping errors in the range $[0.05\%, 1\%]$, due to the completion of the human genome project [9]; for other species such as cattle, a high-resolution genetic map of their whole genomes has not yet been available, and consequently their slightly lower density SNP chips could contain more missing data and errors, which is similar to the earlier human DNA microarrays whose missing calls could range from 5% up to 20% [13, 31].

The unphased genotype data obtained from the high-throughput genotyping approaches are considered to be the major issue that complicates GWAS, since most existing tools for GWAS could not handle data with missing values. The missing genotypes present in the dataset, particularly when the percentage is high, also challenge the current association study methods. In practice, when markers with missing genotypes are recognized to be extremely suspected, one can choose to repeat the genotyping or modify the GWAS tools to accommodate the missing data. However, both approaches are expensive in terms of labor and cost. Another common strategy to tackle the missing values in the SNP data is to discard those SNP markers and/or samples that contain missing genotypes above some threshold [11, 14]. However, this may significantly reduce the mapping resolution in sacrifice of good data — see also the Results chapter on our dataset preprocessing, detection power of GWAS tools, and undermine the inference of gene-trait association [11]. Lastly, one can try to computationally infer and substitute the missing genotypes with predicted values, also known as *imputation*. Although imputation tends to be a low-cost approach, we should be cautious that a poor imputation may introduce biases or errors to the SNP datasets [25].

The so-called "unphased genotypes" also challenge the missing genotype imputation. For diploid organisms such as human, chromosomes come in pairs. The Mendelian law of inheritance states that, for each individual, one of a pair of homologous autosomes is inherited from her father and the other from her mother. However, SNP genotype data do not specify which chromosome comes from the paternal or the maternal, and are usually referred to as unphased data. As a result, missing genotype imputation is usually coupled with the haplotype inference. Some approaches to genotype imputation involve haplotype inference at a preprocessing stage in order to recover the inheritance information. Roberts *et al.* [25] mentioned that haplotype inference with missing data is known to be computationally intractable. Therefore, either near optimal approximation algorithms are designed to facilitate haplotype inference with missing data, or machine learning techniques are adopted to find the best choice for the reconstruction of haplotypes based on domain knowledge.

1.3 Motivation of Missing SNP Genotype Imputation

With the help of missing genotype imputation, current GWAS tools that usually do not tolerate missing values can continue to be used without modification. Moreover, missing genotype imputation can greatly improve the detection power of GWAS without reducing the resolutions of SNP data in that success of GWAS is governed by statistical power. By not reducing SNPs or samples into the study, the statistical power becomes stronger. Although identifying strong gene-trait associations may require relatively few samples, large numbers of samples can get rid of lower-penetrance effects. In genetics, the proportion of individuals carrying a particular variation of a gene that would only sometimes express an associated trait is known as lower-penetrance effects.

The gene-trait association study can extend our knowledge of diseases and help design customized drugs. SNPs are known to affect drug metabolism and clearance of drugs. For instance, GWAS on SNPs can help predict the likelihood that someone will develop a particular illness and answer questions such as why individuals differ in their side effects when absorbing the same therapeutic. In the future, physicians and pharmacists can resort to individual SNP sequencing and design customized drug therapy for any particular patient.

1.4 Our Contributions to Missing SNP Genotype Imputation

Our main contribution in this work is to develop a framework that uses efficient, effective, and biologically meaningful machine learning approaches that work with a genetic map to infer the missing genotypes within a SNP dataset. The genetic distance shown in the genetic map serves as a parameter threshold for finding out the haplotype blocks of SNPs that tend to stay together during inheritance and is simple enough to model the recombination events and cluster the closely linked SNPs within a short region. We design a novel nearest neighbor algorithm and a weighted version to facilitate the fast imputation of missing genotypes.

1.5 Thesis Outline

The rest of the dissertation is organized as follows: In Chapter 2, we briefly provide an introduction to the concepts of genetics in Section 2.1. In Section 2.2 we formally define the missing SNP genotype imputation problem, followed by the evaluation measures of the imputation problem in Section

2.3. In Chapter 3, the important missing SNP genotype value and haplotype allele imputation methods proposed in recent years are reviewed in detail, including fastPHASE and NPUTE. In Chapter 4, the machine learning algorithms we employed are introduced, including the nearest neighbor algorithm, its weighted variants, neural network, support vector machines, and first-order Markov chain. Chapter 5 presents the experimental results, and discusses a number of factors that are important to the imputation. We conclude the dissertation in Chapter 6 to point out some possible future work directions.

Chapter 2

Background

2.1 Concepts in Genetics

Chromosomes are organized structures of the double-stranded DNA sequences, which carry genetic information of an organism. Geneticists identify those positions where SNPs reside on a chromosome, called SNP *loci*. In this dissertation, we consider only *biallelic* SNPs of diploid organisms. That is, at a SNP locus, there are only two possible distinct alleles, denoted by 0 and 1 respectively. For humans, SNPs made up most of the genetic variations [8, 6] and there are millions of them. SNPs occur once in every 300 basepairs on average, and there are estimated about 10 million SNPs in the human genome. In a high density SNP genotype dataset, SNP loci are physically close to each other, and alleles at these loci tend to stay together over a small distance, and thus called genetically *tightly-linked*. For this reason, sometimes SNPs are referred to as tightly-linked markers.

Diploid organisms such as human are species that have paired chromosomes. For each individual, a *genotype* at a SNP locus of a pair of homologous chromosomes consists of two alleles. Since genotype does not provide information of which one of the two chromosomes each allele comes from, genotype at a locus can be denoted as an unordered pair of alleles, and the genotype of a homologous chromosome is a sequence of unordered alleles of SNPs. On the other hand, a haplotype at a SNP locus consists of two alleles and specifies which chromosome each allele comes from; a haplotype of a chromosome consists of all the alleles, one for each SNP locus, on the chromosome. Figure 2.1 illustrates concepts mentioned above.

Although, according to the Mendelian law of inheritance mentioned previously, a child inherits parental genetic information for each locus, in general, she does not inherit the exact copies from the paternal and maternal chromosomes respectively due to the existence of mutations and recombination events. That is, during the meiosis process, the two parental chromosomes get duplicated and shuffled and four chromatids are generated; one chromatid is passed on to the child. Nevertheless, it is observed that the recombination event is rare [18]. Between two consecutive SNP loci along the chromosome, the recombination rate is described by the genetic distance between them. Such information can be obtained from a genetic map.

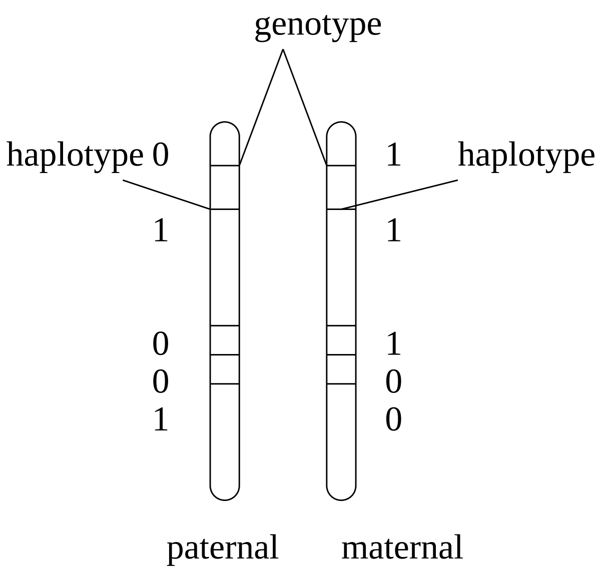

Figure 2.1: An illustration of a structure of chromosomes, genotypes, and haplotypes.

Due to the Mendelian law of inheritance and the fact that mutations and recombination events are rare, for each haplotype allele at a small block of chromosome, it is likely that many individuals share the same haplotype allele due to identical-by-descent (IBD). In other words, unrelated individuals in population data tend to have common alleles from a common ancestor in a short chromosomal region. This is also known as the coalescent theory in genetics. Such regions are usually referred to as high *linkage disequilibrium* (LD) regions. In fact, the coalescent theory underlies most of the *haplotyping-based* imputation methods using a variety of techniques. The extent of the haplotype block shared among individuals can be different from locus to locus and is limited by the existence of mutations and recombination events. Therefore, any well designed genotype imputation methods should be able to make a balance between the coalescent theory and recombination events. A common way is to consider the similarities among haplotype alleles within a viably sized window [25] or a high LD region [26].

All the imputation methods discussed in this dissertation are claimed to be able to work on high-density SNP population data. By population SNP data, we mean that the samples are unrelated. Moreover, the SNPs in the population data are tightly linked and correlated because the millions of SNPs are collected and the average distance between two consecutive SNP markers is small. The missing SNP genotype imputation on large-scale population data can be formulated as follows. Given SNP genotype population data with missing calls, our goal is to efficiently and effectively, in terms of speed and accuracy respectively, impute missing SNP data based on the coalescent theory and recombination. We would like the possible bias and potential errors to be as small as possible.

2.2 Missing SNP Genotype Problem Formalism

To date, all genotype imputation approaches fall into the following four categories: (1) direct genotype imputation with the use of a haplotype reference panel [22]; (2) an integration of haplotype inference and imputation; (3) post-haplotyping imputation method that deals with missing haplotype alleles; (4) direct genotype imputation without the use of a haplotype reference panel and without haplotype inference. Most missing genotyping imputation approaches try to infer missing genotypes from a commercial SNP array by utilizing a reference panel composed of haplotypes from Phase II of the International HapMap Project [10]. NPUTE [25] by Roberts *et al.* is a fast nearest neighbor algorithm, which is a post-haplotyping imputation method. In this dissertation, we focus on addressing missing SNP value imputation in the latter two scenarios.

For post-haplotyping imputation scenario, we assume that the SNP data either have identical alleles at each locus for each genotype or have been preprocessed so that haplotypes are obtained for each individual at each SNP locus. Therefore, post-haplotyping imputation can be considered as a binary classification problem. That is, we are given a SNP haplotype dataset $H = \{h_1, h_2, \ldots, h_n\}$ with n haplotypes at M SNP loci drawn from a population, where $h_i = \{h_{i1}, h_{i2}, \ldots, h_{iM}\}$ and $h_{im} \in \{0, 1, ?\}$. The task is to infer those missing alleles denoted by "?" within the dataset. The possible values for each missing allele are $\{0, 1\}$. For our implemented approaches, an additional genetic map is provided, which specifies genetic distance between every two SNP loci. For organisms whose genetic map is not available, we approximate the genetic distance to be the difference between two physical positions divided by one million, since one centi-Morgan (cM) corresponds to about one million basepairs on average along human chromosomes.

For direct genotype imputation without the reference haplotype and without haplotype inference, the problem can be formalized as follows. Suppose that we have a biallelic SNP genotype dataset comprised of n diploid individuals over M SNP loci drawn from a population and we use 0 and 1 to denote the two distinct alleles at each SNP locus. The possible genotypes are $\{00, 01, 11\}$. Let $G = \{g_1, g_2, \ldots, g_n\}$ denote the genotypes for n individuals, each g_i comprised of genotype data at M markers. The data set can be represented as an $n \times M$ matrix, in which each unphased genotype g_{ij} can be represented as an unordered pair of alleles with one of the four values: $\{00, 01, 11, ??\}$, where 00 and 11 are called *homozygous*, 01 is called *heterozygous*, and $??$ denotes a missing genotype. Similarly, aside from the input matrix G, a genetic map is provided to keep track the genetic distance for each locus. Thus, we define the genotype imputation as a *multi-classification* problem: given a genotype data set G, we try to assign one of the three classes $\{00, 01, 11\}$ to each of the missing SNP genotypes.

2.3 Performance Evaluation

For the missing SNP value imputation problem, the classification accuracy (or imputation accuracy) is a standard measurement for evaluating the performance of any approach, which is defined as the proportion of correctly imputed values. Note that we only take into account those missing SNP values that have at least four neighboring SNP loci within a chosen genetic distance threshold, because we need information from neighboring loci to construct features for SVM and neural networks. We note that in the literature of genotype imputation, imputation has also been regarded as a regression problem [19], for which the imputation accuracy is defined to be the percentage of correctly imputed *minor alleles* over total number of minor alleles in the target missing SNP values. Given that each genotype consists of two alleles, we think that such a definition overestimates the performance of any approach. For example, if the correct genotype was 00 at a SNP locus for a particular individual and a heterozygous genotype 01 was imputed, they viewed this scenario as having produced one of two correct alleles; however, we consider this to be an incorrectly imputed genotype.

Chapter 3

Related Work

This chapter briefly surveys previous work that employed machine learning algorithms to tackle the missing SNP value imputation problem. The machine learning approaches can be categorized into four fields as mentioned in Chapter 2. We also survey the related machine learning algorithms applied for haplotyping inference, which are closely linked to the missing value imputation. All computational and statistical approaches for missing value imputation and haplotype inference are based on the observation of nonrandom patterns of alleles over short regions of tightly linked loci. Niu *et al.* [23] in 2002 introduced the idea of “partition ligation” to divide SNP loci along the genome into segments containing a small number (about 8) of order-preserved consecutive loci [23], and applied Gibbs sampler for haplotype inference. Qin *et al.* [15], Stephens and Donnelly [28] and Lin *et al.* [20] employed this idea in subsequent haplotype inference and missing value imputation studies. We refer to the imputation after the haplotype inference as a *post-haplotyping imputation* and the direct genotype imputation as *genotype-based imputation*.

The most common machine learning imputation approach is to predict missing genotype from haplotype frequencies of population samples using either Bayesian methods [30, 21, 23] or expectation maximization (EM) [15]. Haplotype frequencies are obtained after performing the haplotype inference at an early stage of the imputation. More recent approaches incorporate models of recombination by partitioning markers into haplotype blocks based on entropy measures [31] or by inferring a mosaic of haplotype clusters [26]. Tree-based imputation methods have also been developed, which infer missing genotype on the basis of perfect phylogeny rather than haplotype structure [12, 11]. Essentially, all these methods impute missing genotypes to satisfy the haplotyping needs, and thus their accuracies highly depend on haplotype inferences.

Haplotype inferences are highly associated with the problem of SNP genotype imputation, and is considered to be the first step to genotype imputation for most existing genotype imputation. To get started, we will first present two machine learning approaches to the haplotype inference.

3.1 Bayesian Approaches for Haplotype Inference

Niu *et al.* [23], in 2002, proposed a Bayesian approach to haplotype inference. Stephens *et al.* [30] adopted the Bayesian idea with consideration of linkage disequilibrium (LD) regions and implemented a software called PHASE (V2.0) to handle haplotype inference.

In the Bayesian approach, parameters are random variables and the goal is to estimate posterior distribution given observed data. In the context of haplotype inference, we compute the posterior distribution of haplotype frequencies given observed genotypes G using Bayes' rule:

$$P(f_H|G) = \frac{P(G|f_H)P(f_H)}{P(G)},$$

where $P(f_H)$ is the prior distribution of haplotype frequencies and f_H is the haplotype frequencies. $P(f_H)$ is assumed to be known. Markov chain - Monte Carlo (MCMC) algorithm is used to calculate $P(G|f_H)$ and $P(G)$, mainly because the state spaces for evaluating $P(G|f_H)$ is exponentially too huge to enumerate.

The two Bayesian approaches proposed by Niu *et al.* [23] and Stephens *et al.* [29] respectively both adopt the Gibbs sampling algorithm [4] to estimate the posterior distribution of haplotype frequencies. However, they differ in the prior distributions they assume. Based on the Dirichlet prior, Niu's approach starts with an assignment of haplotype frequencies. At each iteration, for each individual, a pair of haplotypes is sampled and the haplotype frequencies are updated based on the pair of haplotypes. On the other hand, based on a prior approximating the coalescent model, Stephens *et al.*'s [29] approach starts with an arbitrary haplotype sampling of the given genotypes and at iteration updates a randomly selected individual.

3.2 Maximum Likelihood Haplotype Inference

The maximum likelihood approach [16] tries to estimate haplotype frequencies that maximize the probability of the observed genotype data, where haplotype frequencies are unknown parameters that need to be inferred. The likelihood of the population dataset is the product of the probability of each individual because all individuals are independent of each other. Moreover, the probability of an individual with an observed genotype is just the summation of the product of two haplotype frequencies for all haplotype pairs that are consistent with the genotype.

The Expectation-Maximization (EM) algorithm is a widely used algorithm for maximum likelihood estimates (MLEs). Initially, the EM assigns arbitrary haplotype frequencies. At the i-th iteration, the expected occurrences of a haplotype allele is calculated using haplotype frequencies (corresponding to the Expectation-step). Next, the haplotype frequencies are updated based on the expected occurrences of a haplotype (corresponding to the Maximization-step). The EM terminates with the haplotype frequencies returned when it converges. It should be noted that the EM algorithm may converge to local maxima, which one can detect by starting from different initial conditions to examine whether they converge to the same solution [16].

3.3 Quantity Measurement for Haplotype Allele Imputation

In 2005, Su *et al.* [31] proposed a new approach for the missing SNP imputation problem based on the information quantity measure "entropy" since LD measurements according to their paper are usually too noisy for haplotype block constructions. Low diversity is a notable feature for haplotype blocks and low entropy indicates low diversity. Missing SNP haplotype alleles can be inferred by considering haplotype frequencies within haplotype blocks. SNP haplotype alleles within haplotype blocks tend to stay together and keep unchanged during the recombination events. The haplotype block structure is measured by entropy satisfying the following conditions:

- the total entropy within a block should be minimized,
- the total entropy between every adjacent blocks should be maximized, and
- the total mutual information of adjacent blocks should be minimized,

where the mutual information of adjacent blocks is defined as the difference of the sum of block entropies and the block entropy's of all combinations of haplotypes across any two blocks. Dynamic programming (DP) is applied to partition the large-scale SNP haplotypes into blocks that satisfy the above three conditions. Next, if there are m missing SNPs within a haplotype block, the goal is to minimize the block entropy $E(\cdot)$:

$$X = \arg\min_{X} E(X),$$

where $X = (x_1, x_2, \ldots, x_m)$ is the random variable of these m missing SNPs. An EM-like iterative process is developed to find a value for each x_i. To impute the i-th missing SNP at the i-th run, in the first step of the iterative process, estimate the frequency of haplotype containing x_i denoted by $h(x_i)$ for $x_i = 0$, given non-missing SNPs in the block (denoted by D) and set of haplotypes excluding $h(x_i)$ (denoted by H):

$$f_0 = P(h(x_i = 0)|H, D),$$

and the conditional probability of frequency of haplotype h for $x_i = 1$,

$$f_1 = P(h(x_i = 1)|H, D).$$

Here $x_i = 0$ if $\frac{f_0}{f_0+f_1} \geq 0.5$; otherwise, $x_i = 1$. In this manner, all the missing SNPs are imputed for each block.

3.4 NPUTE

Roberts *et al.* in 2007 proposed a new post-haplotyping imputation approach called NPUTE using fast K-nearest neighbor (KNN) searches with the following three key elements [25]:

- data structures support fast K-NN searches over arbitrary window sizes in constant time,
- the advantage of fast speed enables exhaustive searches over all reasonable window sizes, and
- the method does not rely on sampling, and hence enables estimation of imputation accuracy by inferring every missing SNP.

The main idea behind this method is that a sliding window centering at the missing entry is set up to find the closest samples to the target sample and the alleles from these closest neighbors are used to fill the missing entry. Let n be the number of samples and M be the number of SNPs in the population dataset. Their approach is illustrated and explained in haplotype-based scheme, meaning that NPUTE bases their imputation on biallelic haplotype alleles denoted by "0" and "1" respectively, not genotypes.

$$v_{ij} = \begin{cases} 0.5, & \text{if either sample } s_i \text{ or } s_j \text{ is missing;} \\ 0, & \text{if } s_j = s_i; \\ 1, & \text{if } s_i \neq s_j. \end{cases}$$

As each SNP pairwise mismatch vector is computed, a new data structure called a *mismatch accumulator array* (MAA) is built:

- Initialize MAA of width $n(n-1)/2$ and height $M+1$. The first row is set to zero;
- Loop through SNPs in their sequence order, each row vector MAA_{i+1} is updated by adding the mismatch vector to the previous row vector.

The cost of constructing such an array (MAA) is linear in the number of SNPs. Hence, the mismatch value within a window size L extending a SNP i above and below can be obtained later by subtracting the MAA vector with index $\max(i-L,0)$ from the vector with index $\min(i+L+1,M)$. The sample with the minimum mismatch value is then selected and the value at the SNP locus is used to fill the missing data in the target sample. If multiple haplotypes are tied for the minimum mismatch value, then they allow a vote for the call. A tie in the vote is broken by taking the next minimum mismatch values into account and so on.

Due to their fast speed, given a population SNP dataset, their approach first tries to find the optimal sliding window size by scanning over a large range of window sizes and estimating the imputation accuracy for each non-missing allele. In this manner, a good estimate of the performance is also obtained. That is, values of non-missing SNP alleles are used to validate the inferred values generated by NPUTE. The accuracy for non-missing unknown alleles is obtained for each sliding window size. The optimal window size is defined as the one that yields the highest imputation accuracy for those non-missing SNPs. After the optimal size has been determined in such a *training phase*, the real imputation starts up, for which the procedure is exactly the same except that this time the program imputes the missing SNPs.

3.5 fastPHASE

A robust and improved missing SNP genotype imputation based on local clustering and hidden Markov model (HMM) was introduced by Scheet and Stephens [26] in 2006. The implementation of their method is called fastPHASE. They adopted a local clustering idea to capture the observation that over a small number of loci haplotype alleles tend to be clustered into similar patterns. The nearby alleles within small regions tend to arise from the same cluster due to the coalescent theory. Therefore, their assumption is that given a SNP genotype dataset, each haplotype allele at a given SNP originates from one of the K clusters; moreover, the cluster membership can be altered along the genome. The cluster membership of any observed genotype at a SNP locus is modeled as the latent variable in HMM. The other improvement, compared to its predecessor PHASE, is its speed [26]. fastPHASE can be outlined in two steps.

Parameter Estimation The parameters $\nu = (\theta, \alpha, r)$ represent the haplotype frequencies within each cluster, relative frequencies of clusters, and the recombination rate, respectively. They are estimated by applying the EM algorithm T times from T different starting points since EM would typically result in different set of estimates $\hat{\nu}_t$. They combined the obtained T estimates to make predictions. The number of clusters, denoted by K, is an input. It is suggested in the paper that it would be more fruitful to try out different K values and to combine the results, than just selecting a single K value.

Missing Genotype Imputation Suppose that the genotype of individual i at the SNP locus m, denoted by g_{im}, is missing. Since $\hat{\nu}_t$ is obtained in the previous step by applying the EM algorithm, the probability that $g_{im} = x$ where $x \in \{00, 01, 11\}$, given all observed genotypes g and estimated parameter values $\hat{\nu}_t$, is computed by the EM algorithm with respect to g_{im}. To infer the value of g_{im}, they used the estimate that yields the best probability value for the missing genotype:

$$\hat{g}_{im} = \arg\max_{x \in \{00,01,11\}} \frac{1}{T} \sum_{t=1}^{T} p(g_{im} = x | g, \hat{\nu}_t).$$

Chapter 4

Methods

Recall that the genotype imputation scenario we are interested in is formalized as a multi-classification problem: given a missing SNP genotype dataset G, we want to assign one of the three classes (values) $\{00, 01, 11\}$ to each missing SNP genotype. Additionally, a genetic map for the corresponding dataset is provided, in which the genetic distance between every two SNP loci can be looked up.

To address both the IBD and the recombination events, we set up a genetic distance threshold as a parameter to generate a block of SNP loci extending the target missing value above and below. We refer to the upper bound and lower bound of the block (the number of SNP loci below and above the target missing SNP locus) as upper window size L and lower window size R respectively. The underlying assumption is that haplotype alleles within this block tend to be IBD. To impute the i-th SNP marker of individual j, denoted by g_{ij}, we construct the training dataset to include sequences with known genotypes ($g_{im} \neq ??$) at SNP locus i, denoted by $\mathcal{T} = \{x_i, y_i\}_{i=1}^{k}$, where x_i is the input feature vector that is constructed based on the neighboring SNPs below and above and y_i denotes the target variable that is derived from $g_{im} \neq ??$. It should be noted that when the missing rate is high, it would be unavoidable for a training dataset to contain missing calls at SNP loci other than locus i. Moreover, we assume that the genotypes are missing at uniformly random (MAR). That is, the occurrences of missing SNP genotypes do not depend on any data.

4.1 Baseline Approach

A naïve method for genotype imputation is to impute missing SNP values with the non-missing genotype that occur most often in the population at the SNP locus, which is also applicable to the post-haplotyping imputation. The naïve approach does not use information from neighboring loci to impute and usually gives poor accuracy. The expected imputation accuracy is equal to p, where p is the frequency of genotype (haplotype) with the most frequent occurrences, but in the worst case it could be as low as 0 for both directly genotype-based imputation and the haplotype-based imputation if we assume genotypes are missing at random. For this reason, in practice, imputation based on the majority allele frequency is often out of interests. We included it as the BaseLine approach in our experiment.

4.2 Nearest Neighbor (NN) and Its Variants

We first extend the NPUTE method [25] to address both the direct missing SNP genotype imputation problem without using reference haplotype panels and the post-haplotyping imputation. The basic idea behind NPUTE is that it uses instances of observations in the training dataset $\mathcal{T}$ in the feature space to infer missing genotypes for testing sequences. As NPUTE does not generalize a learning model for classification tasks but rather bases its learning directly on the *stored* training instances, it is also sometimes referred to as a memory-based learning method. One key factor in all k-nearest neighbor methods is to define a distance function that implies "closeness" between samples with a voting scheme.

4.2.1 Distance Function

Given a genotype dataset G, the scoring scheme δ for two genotypes at a SNP locus is shown in Table (4.1). This scoring scheme assumes the hamming-like code for genotypes 00, 01, 11, and $??$ represented as three-dimensional vectors $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$, and $(0, 0, 0)$ respectively. For instance, $\delta(00, 11) = |1 - 0| + |0 - 0| + |0 - 1| = 2$, $\delta(??, 00) = |0 - 1| + |0 - 0| + |0 - 0| = 1$, $\delta(??, ??) = 0$ and so on.

Table 4.1: The scoring scheme $\delta(\cdot, \cdot)$ between two genotype alleles.

$\delta(\cdot,\cdot)$	00	01	11	$??$
00	0	2	2	1
01		0	2	1
11			0	1
$??$				0

For post-haplotyping imputation, we split each genotype sequence into two haplotype sequences. The task becomes to impute missing alleles $?$ from $\{0, 1\}$. Therefore, the score function is the same as the one used in NPUTE, shown in Eq. 4.1:

$$\delta(a, b) = \begin{cases} 0, & \text{if } a = b \neq ? \\ 1, & \text{else if } a = ? \text{ or } b = ? \\ 2, & \text{otherwise,} \end{cases} \tag{4.1}$$

where a and b are two haplotype alleles.

Subsequently, the distance between each sample k and the target sample j at the target SNP locus i is defined as

$$dist_i(j, k) = \sum_{i-L \leq m \leq i+R, m \neq i} \delta(g_{mj}, g_{mk}). \tag{4.2}$$

We also introduce two *weighted* k-nearest distance functions based on the fact that linkage disequilibrium (LD) decreases as we moves away. The first approach, denoted by "WeightedNN" in our experiment, is to base weights on the window size L and R as shown in Eq. (4.3). That is, we put more weights to SNPs that are closer to the missing SNP at locus i.

$$dist_i(j,k) = \sum_{i-L \leq m < i} (L - i + m + 1)\delta(G_{mj}, G_{mk}) + \sum_{i < m \leq i+R} (R - m + i + 1)\delta(G_{mj}, G_{mk}). \tag{4.3}$$

Next, we use *pointwise mutual information* (PMI) between the SNP locus with the missing value to be imputed and every other SNP locus within the block as weights. The PMI between loci i and j can be calculated as follows:

$$SI(i,j) = \sum_{x,y \in \{00,01,11\}} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}, \tag{4.4}$$

where $P(x,y)$ is the joint probability of genotype x at SNP locus i and genotype y at SNP locus j, and $P(x)$ is the probability of genotype x at the given SNP locus. In details, we calculate $P(x,y)$ and $P(x)$ in the first order Markov chain. Therefore, the modified distance function is

$$dist_i(j,k) = \sum_{i-L \leq x \leq i+R, x \neq i} SI(i,x) \times \delta(g_{xj}, g_{xk}). \tag{4.5}$$

We refer to the above weighted NN approach based on PMI as “MIKNN” in our experiment.

4.2.2 Voting Scheme

Using the scoring scheme defined in Table 4.1 for direct genotype imputation and Eq. (4.1), the distance between the target sample j and every sample of the training dataset can be calculated, from which the genotype value at locus i of the nearest neighbor(s) can be used for filling the missing genotype g_{ij}. In practice, there can be multiple tied nearest neighbors, and they might have distinct genotypes at locus i. In this dissertation, the value of k in k nearest neighbors refers to k distances rather than k nearest samples. For instance, with $k = 1$, 1NN tries to include all samples of the training dataset that have the equal minimum distance to the target sample. The most straightforward scheme for letting these nearest neighbors to vote on the presence of a testing sequence is the majority vote scheme, which basically treats each neighbor as one vote, and we choose the genotypes with the highest vote to impute the missing value g_{ij} at locus i. If votes for all distinct genotypes at locus i tie, then the majority voting scheme can choose to randomly select a sample from the tied genotypes and use its genotype for imputation. Our preliminary testing exhibited that such a majority voting did not yield satisfactory imputation accuracy. Besides, for classic k nearest neighbor, one needs to test multiple choices of k on the training data to find k that works best on the dataset.

Consequently, we turned to the voting scheme that is employed in NPUTE [25]. In this new voting scheme, we start with the nearest neighbors for voting and choose the genotype of highest votes for imputation. In case of tied votes among the nearest neighbors, it adds more training instances that are the next closest to the testing sample till the tied votes are broken. We adopted this scheme for our implemented nearest neighbor method, denoted as NN.

Alternatively, we took the neighborhood information of nearest neighbor(s) into account to reduce ambiguity. That is, we also included samples in the training dataset whose distances to one of the nearest neighbors are within the nearest distance. Then, we followed the same procedure mentioned above to impute missing genotype g_{ij}. Our method Neighbor1NN is implemented based on this idea, where we selected the nearest neighbors for voting, and in case of tied votes, we used the nearest distance to the testing sequence as a *distance threshold* and for each nearest neighbor to add training instances that are less than or equal to the distance threshold for voting. The genotype which receives the majority vote will be chosen as the imputed value. In case of tied votes, we randomly select one from candidate genotypes of tied highest votes.

4.3 Artificial Neural Network (NeuralNet)

We employed a standard three-layer feed-forward neural network with a gradient descent training algorithm for genotype imputation.

4.3.1 Sequence Encoding and Output Interpretation

For each missing SNP genotype we define a block with upper window size L and lower window size R, and set $W = L + R$. We adopt the orthogonal encoding, where genotypes $\{00, 01, 11\}$ are encoded by orthogonal binary vectors $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$ respectively. The advantage of this orthogonal encoding is that we do not need to introduce algebraic correlations between genotypes. Besides, compared to other complex encodings of SNP genotypes, whether orthogonal or not, our encoding scheme needs not worry about filtering extra information. For example, if one includes too much extra information that might not be strongly correlated to the output, the imputation task can even become harder [3]. In NeuralNet, the missing genotype "??" is predicted by a binary vector (a_{00}, a_{01}, a_{11}), where $\sum_{i\in\{00,01,11\}} a_i = 1$ and a_k denotes the frequency of genotype k at the SNP locus in the training dataset. It should be noted that the encoding scheme has the disadvantage of being wasteful of memory space, because it requires an input layer of size $3 \times W$. Let $\mathbf{x}$ denote the input vector of $3 \times W$ dimension. Assume there are M neurons in the hidden layer and $K = 3$ neurons in the output layer. In our experiment, we set M to $\max(W, 3)$ for genotype imputation and to $\max(W, 2)$ for post-haplotyping imputation respectively. The output of the neural network is a vector $\mathbf{y} = (y_1, y_2, y_3)$, where $y_i \in (0, 1)$ and $\sum_{i=1}^{3} y_i = 1$, which is calculated by the *softmax* function

$$\mathbf{y} = \frac{\exp \mathbf{T}}{\sum_{k=1}^{3} \exp(T_k)},$$

where

$$\mathbf{T} = (T_1, T_2, T_3),$$

$$T_k = \beta_{0k} + \beta_k^\top \mathbf{z},\ k = 1, 2, 3,$$

$$z_m = \sigma(\alpha_{0m} + {\alpha_m}^\top \mathbf{x}),\ m = 1, 2, \ldots, M,$$

and

$$\sigma(\alpha_{0m} + {\alpha_m}^\top \mathbf{x}) = \frac{1}{1 + \exp(-(\alpha_{0m} + {\alpha_m}^\top \mathbf{x}))}.$$

4.3.2 Network Model and Training

Let $\mathbf{w} = \{\alpha_{0m}, \alpha_m; \beta_{0k}, \beta_k\}$ denote the weights to be trained, where $m = 1, 2, \ldots, M$ and $k = 1, 2, 3$. In total, the neural network we constructed has $3M(W + 1) + 3(M + 1)$ parameters to be trained. Further, for each sequence $\mathbf{x}_i$ in the training dataset $\mathcal{T} = \{\mathbf{x}_i\}_{i=1}^n$, there is an associated observed genotype value $\mathbf{t}_i$, where $\mathbf{t}_i \in \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$. In the training phase, we try to minimize the error function

$$E(\mathbf{w}) = \sum_{i=1}^{n} \sum_{k=1}^{3} (-t_{ik} \log(y_{ik})). \tag{4.6}$$

A batch version of the gradient descent training approach is applied for all SNP loci with missing calls. That is, we initialize the weights $\mathbf{w}$ with some random guess. Then, we iteratively update the weights at time step $(t + 1)$ as follows:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + (1 - \mu)\eta \nabla E(\mathbf{w}^{(t)}) + \mu(\mathbf{w}^{(t)} - \mathbf{w}^{(t-1)}), \tag{4.7}$$

where η is the learning rate and μ is the momentum in the range $[0, 1]$. Adding the momentum could smooth out oscillations since $-\nabla E(\mathbf{w})$ in practice may not always point to the global minimum of the error function. We stop our training process after a fixed number of iterations, which has the advantage of saving us out of tuning the regularization ratio, since a large regularization could cause over-fitting.

4.4 Support Vector Machine

SVMs (Support Vector Machines) are a useful algorithm for classification tasks. We use the SVM software LIBSVM [7] for genotype imputation.

4.4.1 Sequence Encoding and Output Interpretation

Similar to neural networks' encoding, we adopt the orthogonal encoding for SNP genotype values, where genotypes $\{00, 01, 11\}$ are represented by orthogonal binary vectors $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$ respectively. Again, the missing genotype "??" is handled by the expected value vector (a_{00}, a_{01}, a_{11}), where a_i represents the frequency of genotype i at the given SNP locus. Therefore, for each input vector, its components lie in the range of $[0, 1]$. For the output, the known genotypes "00", "01" and "11" are represented as 0, 1, and 2 respectively. For post-haplotyping imputation, because it has no heterozygous genotypes 01 in the dataset, we use $(0,1)$ and $(0,1)$ to denote the homozygous alleles 00 and 11 respectively, and missing genotypes "??" are encoded as (a_{00}, a_{11}), where $\sum_{i \in \{00,11\}} a_i = 1$, and each a_i is the frequency of the homozygous genotype i within the training dataset at the SNP locus.

4.4.2 Model Selection

Given the training dataset $\mathcal{T} = \{\mathbf{x}_i\}_{i=1}^n$ and the corresponding genotype value (target) y_i, where x_i follows the input encoding scheme. For posting haploying imputation $y_i \in \{-1, 1\}$ to represent 00 and 11 respectively. For post-haplotyping imputation, the support vector machine is a minimization problem [32]

$$\begin{aligned} \underset{\mathbf{w},b,\xi}{\text{minimize}} \quad & \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \end{aligned} \tag{4.8}$$

where $C > 0$ is the penalty term of the errors. Usually the problem is solved in its dual form:

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & \frac{1}{2}\alpha^\top Q\alpha - \mathbf{e}^\top \alpha \\ \text{subject to} \quad & \mathbf{y}^\top \alpha = 0, \\ & 0 \leq \alpha_i \leq C, i = 1, 2, \ldots, n. \end{aligned} \tag{4.9}$$

where $\mathbf{e} = [1, 1, \ldots, 1]^\top$, $Q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, and $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(x_i)^\top \phi(x_j)$ is the kernel function. For each input $\mathbf{x}_i$, the function ϕ maps it into a higher dimensional space. SVM can be viewed as a minimization problem that tries to find a hyperplane with maximal margin in the higher dimensional space.

For direct genotype imputation, LIBSVM constructs $\frac{k(k-1)}{2} = \frac{3\times 2}{2} = 3$ classifiers between every two different classes. The result from each binary classification for each training sample of $\mathcal{T}$ is viewed as a vote. Similar to nearest neighbor's majority vote scheme, when a testing sequence $\mathbf{x}$ comes in, its class is assigned to be a class (genotype) with maximum number of votes. In case of tied votes, LIBSVM simply select the one with the smallest index according to its implementation.

We adopt the radial basis function (RBF) for genotype imputation, where the RBF kernel is given by $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2)$ and $\gamma > 0$. The advantage of RBF kernel is that it can handle the case where the relation between class labels and attributes is nonlinear.

4.4.3 Cross-Validation and Grid-Search

There are two parameters for an RBF kernel: C and γ. It is not known beforehand which C and γ are best for a given problem; consequently some kind of model selection (parameter search) must be done. The goal is to identify good (C, γ) so that the classifier can accurately predict missing genotypes. The purpose of training is not trying to achieve high training accuracy for the training dataset but rather find a model that is general enough and can work well for those missing genotype sequences. A common strategy is to use cross-validation. In ℓ-fold cross-validation, we first divide the training set into ℓ subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining $(\ell - 1)$ subsets. Thus, each sample of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified. The cross-validation procedure can prevent the over-fitting problem.

LIBSVM implements a so-called "grid search" for choosing an optimal pair (C^*, γ^*) using cross-validation. In practice, LIBSVM tries different pairs of (C, γ) values drawn from a table consisting of two finite exponentially growing sequences of $C = \{2^{-5}, 2^{-3}, \cdots, 2^{15}\}$ and $\gamma = \{2^{-15}, 2^{-13}, \cdots, 2^{3}\}$. The pair (C^*, γ^*) that yields the best cross-validation accuracy is selected as an optimal pair. After finding optimal parameters (C^*, γ^*), LIBSVM trains the entire training dataset again to construct the final classifier for testing samples.

4.5 First-order Markov Chain with Add-One Smoothing

Inspired by its success on biological sequence applications such as DNA sequence analysis [2], we apply the Markov chain (MC) model to the the missing genotype imputation problem on SNP datasets. A first-order Markov chain $\{X_n, n = 0, 1, 2, \ldots\}$, sometimes also called the observed Markov model, is a stochastic process that takes on a finite set of possible values and is defined by an initial probability distribution $P(X_0)$ and transition probabilities between two states $P(X_n|X_{n-1})$. In a first-order Markov chain, the probability of a current state depends only on the previous state, and is independent of other past states:

$$P(X_n|X_0, X_1, \ldots, X_{n-1}) = P(X_n|X_{n-1}).$$

Given an observation sequence, we are interested in computing the joint probability

$$P(X_0, X_1, \ldots, X_n) = P(X_0) \prod_{n=1}^{n} P(X_n|X_{n-1}).$$

To impute a missing SNP value g_{ij}, we construct two first order local Markov chain (MC) using the L up-stream SNPs and R downstream SNPs within the genetic distance threshold respectively. Each genotype at a SNP marker represents a state and takes on genotype values from $00, 01, 11$. We explain in detail how to calculate the initial probability and the subsequent transition probabilities.

4.5.1 State Probabilities in MC

The upstream MC consists of SNP markers in the order $\langle g_{(i-L)j}, g_{(i-L+1)j}, \ldots, g_{(i+1)j}, g_{ij}\rangle$, where g_{ij} is the target missing SNP genotype denoted by ??. Similarly, the downstream MC consists of SNP markers in the order $\langle g_{(i+R)j}, g_{(i+R-1)j}, \ldots, g_{(i-1)j}, g_{ij}\rangle$. We illustrate how to compute the initial probability for the upper-stream MC and in the similar manner, the initial probability for the downstream MC can be computed as well. In the first step, we count at SNP locus $(i - L)$ the frequency of each known genotype values 00, 01 and 11, respectively, occurring in the training dataset $\mathcal{T}$, which yields the probability distribution $P(X_0)$. Let c_{00}, c_{01}, and c_{11} be the frequencies of genotypes 00, 01, and 11 respectively. In case of any zero frequency, we apply add-one smoothing to the counts of each genotype. Therefore, $P(X_0 = i) = \frac{c_i+1}{c_i+V}$, where $i \in \{00, 01, 11\}$ and V refers to the number of distinct types of genotypes at the locus ($V = 3$ for direct genotype based imputation and $V = 2$ for post-haplotyping imputation). If the observed genotype $g_{(i-L)j} \in \{00, 01, 11\}$ is known, we use $P(X_0 = g_{(i-L)j})$ for calculation; in case of another missing value observed in the initial state, we use expected value of $P(X_0)$ instead. That is,

$$P(X_0 = ??) = E[P(X_0)] = \sum_{a \in \{00,01,11\}} P(X_0 = a) \times P(X_0 = a).$$

4.5.2 Transition Probabilities in MC

To compute the transition probability from state X_{n-1} to state X_n at two adjacent SNP loci, we count the frequencies of all combinations of genotypes in the training dataset. Again, add-one smoothing is applied to each count when any frequency is 0. If genotypes at both X_{n-1} and X_n are known, then we simply use $P(X_n|X_{n-1})$ derived from the frequencies. We provide formula for how to calculate the transition probability from state $X_{(n-1)}$ to state X_n in case of occurrences of missing values:

$$P(X_n = a|P(X_{(n-1)}) = ??) = \sum_{b \in \{00,01,11\}} P(X_n = a|X_{(n-1)} = b)P(X_{(n-1)} = b),$$

where $a \in \{00, 01, 11\}$;

$$P(X_n = ??|P(X_{(n-1)}) = b) = \sum_{a \in \{00,01,11\}} P(X_n = a|X_{(n-1)} = b)P(X_n = a),$$

where $b \in \{00, 01, 11\}$;

$$P(X_n = ??|P(X_{(n-1)}) = ??) = \sum_{a,b \in \{00,01,11\}} P(X_n = a|X_{(n-1)} = b)P(X_n = a)P(X_{(n-1)} = b).$$

Chapter 5

Results and Discussion

To assess the performance of the imputation algorithms, we used two real SNP datasets for simulation studies. We examine both missing SNP genotype imputation and missing SNP haplotype imputation.

The first dataset was obtained from the International HapMap project (Phase I) [10], the non-redundant SNP genotype dataset. Population in this genotype dataset have been grouped according to their ancestry, and we chose the sub-population SNP genotype dataset on chromosome 17 of African ancestry in Southwest USA (ASW) for study. We refer to this dataset as the *human* dataset, without specifying more detailed information in the sequel. The dataset consists of 83 individuals genotyped at 40,775 SNP loci along the chromosome, and it contains 0.268% missing calls. The human dataset is used for performance evaluation on missing SNP genotype imputation, and, for that purpose, those SNP loci containing a missing value were removed, with 34,071 (or 83.60%) SNP loci left.

The second dataset was extracted from the NIEHS/Perlegen resequencing project, which provides a high-resolution map of 16 common mouse strains with 11.1% missing calls. We used again the chromosome 17 SNP whole genome dataset, which is made up of 15 inbred mouse strains genotyped at 288,229 SNP loci along the chromosome. Our examination confirmed that all the genotype values are homozygous (*i.e.*, only 00 and 11); This dataset is referred to as the mouse dataset, and it is used for performance evaluation on missing SNP haplotype imputation. Again, those SNP loci containing a missing value were removed, with 144,820 (or 50.24%) SNP loci left for simulation studies.

5.1 Simulated Missing SNP Datasets

For both the human and mouse datasets, we first generated three datasets at three different density levels, to mimic the real high, medium, and low density genotyping arrays. They are density-1, the original dataset, density-0.1 and density-0.01. The density-0.1 datasets were obtained from the original dataset by picking every 10-th SNP, and thus contains 3,408 and 14,482 SNPs, respectively.

Likewise, the density-0.01 datasets were obtained from the original dataset by picking every 100-th SNP, containing 341 and 1,449 SNPs, respectively. So now we have six datasets, three human and three mouse.

Next, taking one of the six datasets, we uniformly randomly mask a portion of genotypes to create a simulated missing SNP dataset. Such portion is called the *missing rate*, and it is one of 0.5%, 1%, 2%, 5%, 10%, and 20%. At each missing rate, a total of 10 simulated datasets were identically and independently generated, to be used in the experiments.

We remark that our simulated missing SNP values are missing completely at random, that is, the missing values are independent of both observable variables and unobservable parameters of interest.

5.2 Experimental Setup

Except fastPHASE and NPUTE, which were run under their instruction, the imputation methods we implemented were run using five different genetic distance thresholds on each simulation dataset to constrain the locality. Table 5.1 summarizes these thresholds for the human and mouse datasets, adjusted by the density level (but not the missing rate). They were set so to guarantee a certain number of, yet not too many, SNP loci inside the covering window for the target missing SNP locus. The genetic distance of the entire human chromosome 17 is 129.4752161384 centi-Morgans (cMs); for mouse chromosome 17, we did not have the precise genetic distance and approximated it by 1cM per million basepairs.

We set up the genetic distance thresholds to respect the Mendelian laws of inheritance, as well as the fact that recombination events are rare so that alleles of nearby SNP loci tend to be inherited together due to IBD. As a side effect, it is possible that some masked SNP loci do not have any neighbor SNP loci within the covering window set by the threshold. We excluded these masked SNP loci from imputation or the subsequent performance evaluation. In fact, we imposed a constraint on the target masked SNP loci to have at least 4 neighbor SNP loci inside the covering window, otherwise not to be imputed by any method.

Table 5.1: Genetic distance thresholds (in centi-Morgan) set for the human and mouse datasets at three density levels.

Dataset	Density	Genetic Threshold (cM)				
human	0.01	1	2	3	4	5
	0.1	0.1	0.2	0.3	0.4	0.5
	1	0.01	0.02	0.03	0.04	0.05
mouse	0.01	0.2	0.4	0.6	0.8	1.0
	0.1	0.02	0.04	0.06	0.08	0.1
	1	0.002	0.004	0.006	0.008	0.01

We preprocess the datasets according to our classification formulation, by randomly assigning 0 for one allele and 1 for the other allele, to convert them into matrices with entries of 00, 01, 11, and $??$. Note that those mouse datasets do not contain 01 entries as no heterozygosities exist, neither would an entry $??$ be imputed as 01. We collected the imputation accuracies for neural network (NeuralNet), nearest neighbor (NN), weighted nearest neighbor (WeightedNN), support vector machine (SVM), Markov chain (MC), neighborhood 1-nearest neighbor (Neighbor1NN), mutual information based weighted nearest neighbor (MIKNN), as well as those of previously the best imputation methods, fastPHASE and NPUTE. Confirmed by the authors of NPUTE, NPUTE was designed to take the advantage of the homozygosities and its code does not work for missing SNP genotype imputation. Therefore, NPUTE was not run on the human datasets. We ran fastPHASE by setting the recommended number of haplotype clusters to 20. To run NPUTE on the mouse datasets, we followed its instruction to first examine a range of window sizes (the number of SNP loci upstream) from 1 up to 50 during the training phase to search for an optimal window size for each dataset; then we chose the window size that yielded the best training imputation accuracy for real missing value imputation.

All our implemented imputation methods were run in the *batch testing* mode, not sequential mode, meaning that no imputed values would be used for imputing other missing values. Though sequential imputation has been reported advantageous [27], our consideration is not to propagate erroneous imputation to the entire dataset.

5.3 Imputation Accuracy Comparison

Recall that each SNP dataset is defined by a combination (species, density, missing rate), and there are $2 \times 3 \times 6 = 36$ such combinations. For each combination there are 10 simulated datasets. We ran all imputation methods on the ten simulated datasets using all five associated different genetic distance thresholds. For fastPHASE, NPUTE, and BaseLine (the majority vote), genetic distance threshold does not have any effects and they were run once only. In the following (and the Appendix), the average imputation accuracies are reported, where the average was taken over various subsets of simulated datasets.

5.3.1 Average Imputation Accuracies

Firstly, for each of the 36 combinations (species, density, missing rate), we calculated the average imputation accuracy for a method by taking the average over all five runs using different genetic distance thresholds on the 10 simulated datasets. That is, it is the average of 50 accuracies (again, for fastPHASE, NPUTE, and BaseLine, it can also be regarded as the average of 10 accuracies, which however might vary slightly due to the changing number of target missing values). Table 5.2 lists these average imputation accuracies for the six combinations of density-0.01 human dataset, where each column corresponds for a missing rate. They are also plotted in Figure 5.1 for easier view of performance difference. As one can see, in these low density datasets, our methods NN and WeightedNN seemingly performed better than previously the best method fastPHASE, which in turn performed better than the other machine learning algorithms.

Table 5.2: Average imputation accuracies at the 6 missing rates on human dataset of density 0.01.

Methods	Missing Rate					
	0.5%	1%	2%	5%	10%	20%
fastPHASE	0.69	0.6782	0.6785	0.6787	0.6756	0.6735
NN	0.7552	0.7521	0.749	0.7471	0.7393	0.7349
WeightedNN	0.7302	0.7161	0.7102	0.7065	0.6936	0.6877
SVM	0.6541	0.6544	0.6488	0.6587	0.6488	0.6516
NeuralNet	0.6478	0.6431	0.6465	0.6528	0.6432	0.6463
Neighbor1NN	0.6579	0.653	0.6497	0.6585	0.6507	0.6541
MC	0.6553	0.6505	0.654	0.6606	0.6513	0.6528
BaseLine	0.6589	0.6535	0.6497	0.6588	0.6511	0.6542
MIKNN	0.606	0.5945	0.6032	0.598	0.5919	0.5902

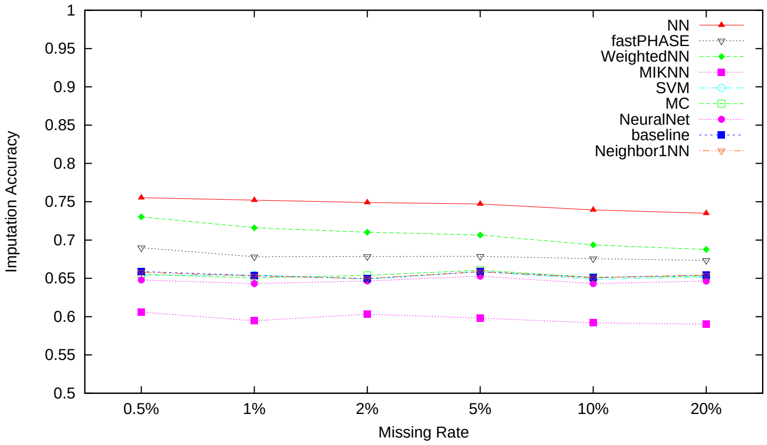

Figure 5.1: Average imputation accuracies at the 6 missing rates on human dataset of density 0.01.

The next five sets of tables and figures list and plot the average imputation accuracies for density-0.1 human dataset (Table 5.3, Figure 5.2), density-1 human dataset (Table 5.4, Figure 5.3), density-0.01 mouse dataset (Table 5.5, Figure 5.4), density-0.1 mouse dataset (Table 5.6, Figure 5.5), density-1 mouse dataset (Table 5.7, Figure 5.6), respectively.

For each of the 36 combinations, among the 50 imputation accuracies, the best one for each imputation method was also recorded, and presented in detail in the Appendix (Section A.2). Continuing the trend, as seen in Figure 5.2 for the medium density human datasets, our method NN still performed better than fastPHASE, which caught up with our WeightedNN. On high density human datasets Figure 5.3, fastPHASE became the winner, beating our NN, WeightedNN, SVM and MIKNN about 3%. This last observation confirms well the claim by fastPHASE that it is designed

Table 5.3: Average imputation accuracies at the 6 missing rates on human dataset of density 0.1.

Methods	Missing Rate					
	0.5%	1%	2%	5%	10%	20%
fastPHASE	0.7782	0.7797	0.7838	0.78	0.7716	0.7566
NN	0.8089	0.8079	0.8056	0.8041	0.7969	0.7848
WeightedNN	0.7875	0.7859	0.7817	0.774	0.7624	0.7454
SVM	0.741	0.7395	0.7385	0.7351	0.7261	0.7104
NeuralNet	0.6301	0.6342	0.6274	0.6732	0.6706	0.6651
Neighbor1NN	0.6532	0.6532	0.6493	0.6553	0.6554	0.6536
MC	0.7046	0.7053	0.7053	0.7065	0.7026	0.6946
BaseLine	0.6554	0.6552	0.6508	0.6567	0.6566	0.6549
MIKNN	0.7198	0.7232	0.7213	0.7156	0.7077	0.6908

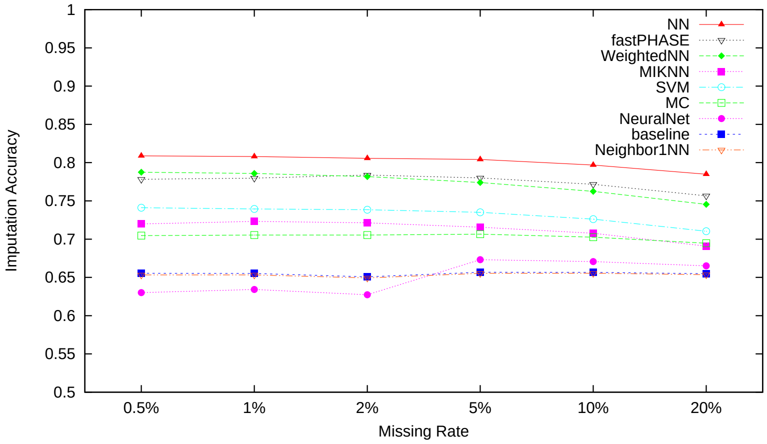

Figure 5.2: Average imputation accuracies at the 6 missing rates on human dataset of density 0.1.

for high density SNP datasets. Yet we may draw the conclusion a bit further that on low to medium density SNP datasets, one should better use our nearest neighbor (NN) imputation method.

Note that on the mouse datasets, NPUTE was run to collect its imputation accuracies. Because of no heterozygosities, imputation on the mouse datasets can also be regarded as missing SNP haplotype imputation, or posting-haplotyping imputation. Different from missing SNP genotype imputation, here one can see from Figures 5.4, 5.5, and 5.6 that regardless of the density, our methods NN, WeightedNN, MIKNN, and fastPHASE performed better than the others including NPUTE. Indeed, these four imputation methods seemingly cluster together, performed better than the machine learning algorithms. Among these four, one can see further from Tables 5.5–5.7 that NN and WeightedNN performed slightly better.

5.3.2 Statistical Significance Testing

To evaluate the imputation accuracy difference between two methods, we performed six statistical right-tailed paired t-tests on the imputation accuracies between the pair at six missing rates 0.5%,

Table 5.4: Average imputation accuracies at the 6 missing rates on human dataset of density 1.

Methods	Missing Rate					
	0.5%	1%	2%	5%	10%	20%
fastPHASE	0.9617	0.9616	0.9611	0.9597	0.9502	0.9378
NN	0.9224	0.9214	0.9189	0.9158	0.91	0.8984
WeightedNN	0.9163	0.9135	0.9091	0.901	0.891	0.8734
SVM	0.9027	0.9021	0.8918	0.8693	0.8715	0.8217
NeuralNet	0.7839	0.7834	0.7804	0.7769	0.769	0.7534
Neighbor1NN	0.6415	0.6419	0.6427	0.6452	0.6455	0.6456
MC	0.7725	0.7718	0.7705	0.768	0.7614	0.7483
BaseLine	0.6504	0.6496	0.6493	0.6502	0.6492	0.6488
MIKNN	0.9123	0.9112	0.9104	0.9062	0.898	0.882

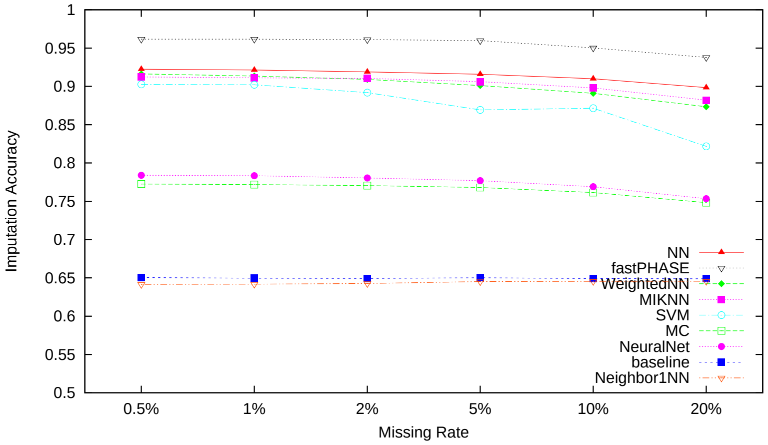

Figure 5.3: Average imputation accuracies at the 6 missing rates on human dataset of density 1.

1%, 2%, 5%, 10%, and 20%, respectively, separated for the human and mouse datasets with different densities. For example, on the density-0.01 human datasets with missing rate 0.5%, there are 50 imputation accuracies collected for every method from 10 simulated datasets each run using five different genetic distance thresholds. The performance of the method is thus represented as a 50-dimensional vector, and the t-test is to evaluate whether one vector is statistically significantly better than another. In more details, in our right-tailed t-test, the hypothesis is

$$H : \mu_1 > \mu_2 \quad \text{(The mean of the first vector is greater than the mean of the second.)}$$

A p-value less than 0.05 indicates that the mean of the first vector is statistically significantly greater than the mean of the second, greater than 0.95 indicates that the mean of the first vector is statistically significantly less than the mean of the second, and close to 0.50 indicates that the two means are statistically no different from each other. Table 5.8 presents the p-values from these pairwise comparisons on the human datasets at 3 density levels, with missing rate 0.5%. These p values were calculated by Octave, a GNU scientific software, and were rounded to have three significant

Table 5.5: Average imputation accuracies at the 6 missing rates on mouse dataset of density 0.01.

Methods	Missing Rate					
	0.5%	1%	2%	5%	10%	20%
fastPHASE	0.9107	0.9072	0.8968	0.8997	0.8917	0.8803
NPUTE	0.8371	0.856	0.8458	0.8445	0.839	0.8281
NN	0.9223	0.9184	0.9101	0.9081	0.899	0.8808
WeightedNN	0.924	0.9197	0.9088	0.9014	0.8887	0.8678
SVM	0.8657	0.854	0.8507	0.8581	0.8495	0.8398
NeuralNet	0.832	0.8271	0.8217	0.8277	0.8164	0.809
Neighbor1NN	0.7846	0.7844	0.7816	0.8068	0.8059	0.8103
MC	0.8388	0.8319	0.8276	0.8384	0.8289	0.8267
BaseLine	0.8213	0.8133	0.8034	0.8179	0.8117	0.8143
MIKNN	0.8995	0.9035	0.8887	0.8902	0.883	0.8666

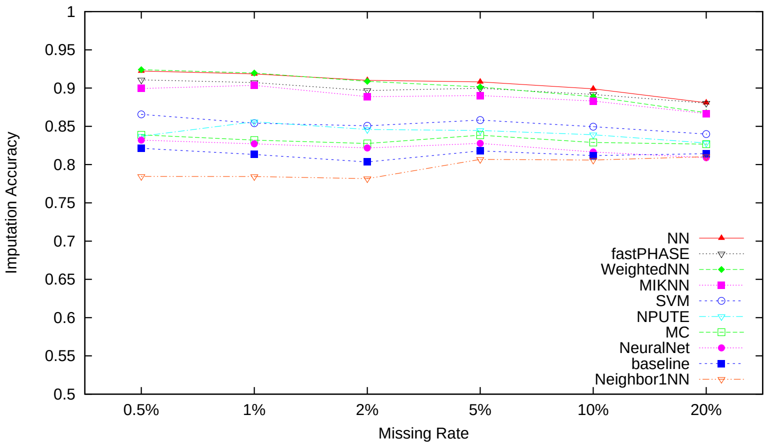

Figure 5.4: Average imputation accuracies at the 6 missing rates on mouse dataset of density 0.01.

digits. Therefore, p-values such as 0.000 or 1.000 should be interpreted as < 0.0001 and > 0.9999, respectively.

From Table 5.8, we can see that our NN and WeightedNN methods performed statistically significantly better than fastPHASE on the human datasets at two density levels 0.01 and 0.1, while at density level 1 fastPHASE performed significantly better than all the other methods. Furthermore, NN outperformed significantly all the other methods at all three density levels, except fastPHASE at density level 1. These strongly suggest that our NN method is useful in practice, given that the imputation time for fastPHASE grows exponentially in the number of SNPs (or equivalently the density) — see Section 5.7 — and it took weeks to months for us to collect the fastPHASE results.

In summary, through the p-value tables (Table 5.8 and five other on the human datasets, six other on the mouse datasets in the Appendix, Section A.1) we are able to draw conclusions on the imputation performances of all the methods. Specifically, on human datasets with missing rate 0.5%, at density 0.01 we have NN $>$ WeightedNN $>$ fastPHASE $>$ $\{$SVM, MC, BaseLine, Neu-

Table 5.6: Average imputation accuracies at the 6 missing rates on mouse dataset of density 0.1.

Methods	Missing Rate					
	0.5%	1%	2%	5%	10%	20%
fastPHASE	0.9334	0.9314	0.9278	0.9271	0.9244	0.9125
NPUTE	0.873	0.8684	0.8661	0.8659	0.8617	0.8531
NN	0.9411	0.9398	0.9355	0.9324	0.927	0.9112
WeightedNN	0.9432	0.9407	0.9343	0.9267	0.9177	0.9009
SVM	0.8861	0.8842	0.8808	0.878	0.875	0.8597
NeuralNet	0.8647	0.8629	0.8579	0.8549	0.8254	0.8114
Neighbor1NN	0.7616	0.768	0.7795	0.798	0.8072	0.808
MC	0.8666	0.8635	0.8568	0.8561	0.8538	0.8429
BaseLine	0.8252	0.8189	0.8155	0.8145	0.8159	0.8134
MIKNN	0.9271	0.9271	0.923	0.9199	0.9152	0.9023

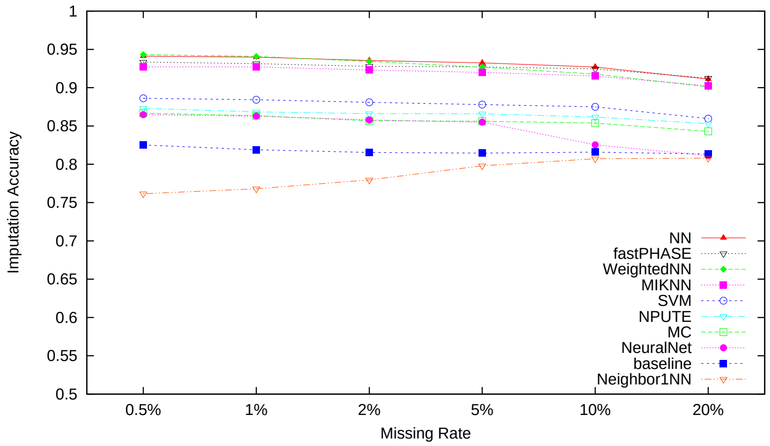

Figure 5.5: Average imputation accuracies at the 6 missing rates on mouse dataset of density 0.1.

ralNet, Neighbor1NN} > MIKNN; at density 0.1 we have NN > WeightedNN > fastPHASE > SVM > MIKNN > MC > {Neighbor1NN, BaseLine} > NeuralNet; and at density 1 we have fastPHASE > NN > WeightedNN > MIKNN > SVM > NeuralNet > MC > BaseLine > Neighbor1NN. Correspondingly on the mouse datasets with missing rate 0.5%, at density 0.01 we have {NN, WeightedNN} > fastPHASE > MIKNN > {NPUTE, SVM} > MC > NeuralNet > BaseLine > Neighbor1NN; at density 0.1 we have {NN, WeightedNN} > fastPHASE > MIKNN > SVM > NPUTE > {MC, NeuralNet} > BaseLine > Neighbor1NN; at density 1 we have NN > WeightedNN > fastPHASE > MIKNN > SVM > NPUTE > MC > NeuralNet > BaseLine > Neighbor1NN.

Table 5.7: Average imputation accuracies at the 6 missing rates on mouse dataset of density 1.

Methods	Missing Rate					
	0.5%	1%	2%	5%	10%	20%
fastPHASE	0.944	0.9452	0.9439	0.9417	0.9383	0.9305
NPUTE	0.8819	0.8821	0.8816	0.8789	0.8758	0.8695
NN	0.9504	0.9511	0.9498	0.9466	0.9408	0.9294
WeightedNN	0.9514	0.9508	0.9468	0.9391	0.9321	0.9213
SVM	0.9035	0.9042	0.8969	0.6902	0.8837	0.7648
NeuralNet	0.8544	0.8524	0.8478	0.8371	0.8245	0.806
Neighbor1NN	0.7503	0.7697	0.7887	0.8044	0.8103	0.8105
MC	0.8733	0.8753	0.8737	0.8705	0.8655	0.8553
BaseLine	0.8179	0.8182	0.8178	0.8168	0.8167	0.8149
MIKNN	0.9414	0.9425	0.9412	0.9386	0.9343	0.9252

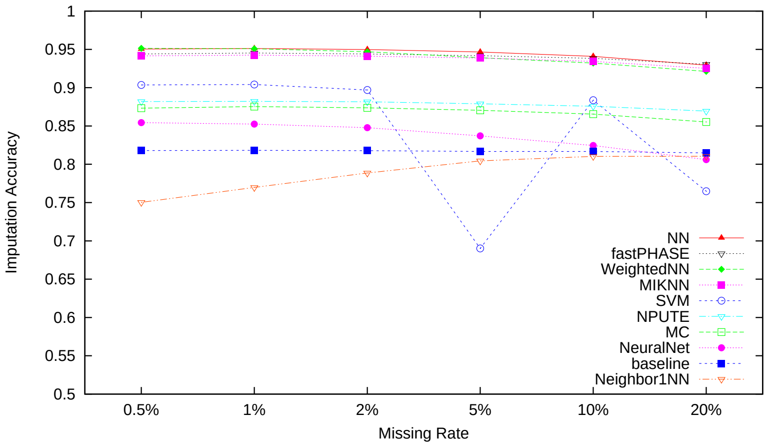

Figure 5.6: Average imputation accuracies at the 6 missing rates on mouse dataset of density 1.

5.4 Effects of Missing Rate

As we see from Figures 5.1–5.6, the missing rate had impact on the imputation accuracy. For example, for NN, WeightedNN, fastPHASE, MIKNN, and NeuralNet, their average imputation accuracies decrease when the missing rate increases, on both human and mouse datasets. The Neighbor1NN, however, performed slightly strangely on the mouse datasets, that its average imputation accuracies increase slightly when the missing rate increases. The SVM showed a sharp drop on the density-1 with missing rate 5%, which is likely dataset specific, since it came back as normal at missing rate 10%.

Except the abnormal behavior of Neighbor1NN and SVM, the general tendency is reasonable. As the missing rate increases, to impute the missing value at a SNP locus, we lost some useful values that were supposed to be used, but chose to use slightly fuzzy information. Moreover, as this information is taken as the expected value over all possible SNP values, it becomes more bias

Table 5.8: The right-tailed t-test p-values for pairwise comparisons on the human datasets at three density levels, with missing rate 0.5%, where the hypothesis is the average imputation accuracy of a row method is greater than the average imputation accuracy of a column method.

	fast PHASE	NN	Weighted NN	SVM	Neural Net	Neighbor 1NN	MC	BaseLine	MIKNN
	Density-0.01 human datasets								
fastPHASE	0.500	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
NN	0.000	0.500	0.001	0.000	0.000	0.000	0.000	0.000	0.000
WeightedNN	0.000	0.999	0.500	0.000	0.000	0.000	0.000	0.000	0.000
SVM	1.000	1.000	1.000	0.500	0.214	0.679	0.557	0.718	0.000
NeuralNet	1.000	1.000	1.000	0.786	0.500	0.903	0.835	0.921	0.000
Neighbor1NN	1.000	1.000	1.000	0.321	0.097	0.500	0.369	0.548	0.000
MC	1.000	1.000	1.000	0.443	0.165	0.631	0.500	0.674	0.000
BaseLine	1.000	1.000	1.000	0.282	0.079	0.452	0.326	0.500	0.000
MIKNN	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.500
	Density-0.1 human datasets								
fastPHASE	0.500	1.000	0.996	0.000	0.000	0.000	0.000	0.000	0.000
NN	0.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
WeightedNN	0.004	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000
SVM	1.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000
NeuralNet	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000	1.000
Neighbor1NN	1.000	1.000	1.000	1.000	0.000	0.500	1.000	0.773	1.000
MC	1.000	1.000	1.000	1.000	0.000	0.000	0.500	0.000	1.000
BaseLine	1.000	1.000	1.000	1.000	0.000	0.227	1.000	0.500	1.000
MIKNN	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000	0.500
	Density-1 human datasets								
fastPHASE	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
NN	1.000	0.500	0.014	0.000	0.000	0.000	0.000	0.000	0.000
WeightedNN	1.000	0.986	0.500	0.000	0.000	0.000	0.000	0.000	0.046
SVM	1.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
NeuralNet	1.000	1.000	1.000	1.000	0.500	0.000	0.000	0.000	1.000
Neighbor1NN	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
MC	1.000	1.000	1.000	1.000	1.000	0.000	0.500	0.000	1.000
BaseLine	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
MIKNN	1.000	1.000	0.954	0.000	0.000	0.000	0.000	0.000	0.500

toward the known values. Even further, for NN and its variants including WeightedNN, MIKNN, and NPUTE, this fuzzy information could lead to more ties, which reduce further their imputation accuracies. On the other hand, for Neighbor1NN, because its voting scheme is different from NN and its variants, it showed a different tendency and became closer to the BaseLine, as expected.

5.5 Effects of Density Level

Putting the second columns of Tables 5.2, 5.3, and 5.4 together as Table 5.9, which are the average imputation accuracies on the human datasets at three density levels all with missing rate 0.5%. The counterparts on the mouse datasets are collected as Table 5.10, as well as with the other five missing rates presented in the Appendix, Section A.3.

From Tables 5.9 and 5.10, one can see that from low to medium to high densities, the imputation

Table 5.9: Average imputation accuracies on the human datasets at three density levels with missing rate 0.5%. Data reproduced from Tables 5.2, 5.3, and 5.4.

	Human datasets with missing rate 0.5%		
	0.01	0.1	1
fastPHASE	0.69	0.7782	0.9617
NN	0.7552	0.8089	0.9224
WeightedNN	0.7302	0.7875	0.9163
SVM	0.6541	0.741	0.9027
NeuralNet	0.6478	0.6301	0.7839
Neighbor1NN	0.6579	0.6532	0.6415
MC	0.6553	0.7046	0.7725
BaseLine	0.6589	0.6554	0.6504
MIKNN	0.606	0.7198	0.9123

Table 5.10: Average imputation accuracies on the mouse datasets at three density levels with missing rate 0.5%. Data reproduced from Tables 5.5, 5.6, and 5.7.

	Mouse datasets with missing rate 0.5%		
	0.01	0.1	1
fastPHASE	0.9107	0.9334	0.944
NPUTE	0.8371	0.873	0.8819
NN	0.9223	0.9411	0.9504
WeightedNN	0.924	0.9432	0.9514
SVM	0.8657	0.8861	0.9035
NeuralNet	0.832	0.8647	0.8544
Neighbor1NN	0.7846	0.7616	0.7503
MC	0.8388	0.8666	0.8733
BaseLine	0.8213	0.8252	0.8179
MIKNN	0.8995	0.9271	0.9414

accuracies of each imputation method increase, except those of NeuralNet decrease from 0.6478 at low density to 0.6301 at medium density on human datasets. Nevertheless, its average imputation accuracy comes back at 0.7839 at high density, and thus we might suspect that the decrement is caused by the simulated datasets. Surprisingly, similar phenomena happen to NeuralNet, Neighbor1NN, and BaseLine on the mouse datasets, for which we are not able to explain confidently.

Prior to our work, NPUTE was one of the best missing SNP haplotype allele imputation program. From Table 5.10 (and Tables 5.5–5.7), we see that fastPHASE and our methods NN, WeightedNN, MIKNN, and SVM all performed statistically significantly better than NPUTE. One possible reason is that the dependencies among the neighboring SNP markers. The number of neighboring SNP loci employed by NPUTE for the imputation, that is the window size of a missing SNP locus, is fixed but it is mostly region dependent. Our employment of genetic distance threshold more accurately reflects such a dependency to match well with the concept of genetic distance, which describes the likelihood of recombination events. More specifically, compared to NPUTE, our local imputation approaches, including NN, WeightedNN, and MIKNN, allow the covering window size to vary from a locus to another and neighboring SNPs are included as features for local imputation only if they are within the genetic distance threshold from the target missing SNP locus.

5.6 Effects of Genetic Distance Threshold

We also investigated the effects of the chosen genetic distance threshold to the imputation methods. We show in this section the results on datasets with missing rate 0.5%, while the readers might refer to the Appendix, Section A.4, for results associated with the other five missing rates. Table 5.11 summarizes the average imputation accuracies, each over the associated 10 simulated datasets, of the imputation methods on the human datasets at three density levels with missing rate 0.5%. Figure 5.7 plots these average imputation accuracies.

Table 5.11: Average imputation accuracies on the human datasets at three density levels with missing rate 0.5%, where the imputation methods were run with five corresponding genetic distance thresholds.

Density-0.01 human datasets	1cM	2cM	3cM	4cM	5cM
fastPHASE	0.7085	0.6852	0.6854	0.6854	0.6854
NN	0.7454	0.75	0.7674	0.759	0.7542
WeightedNN	0.7501	0.7521	0.7278	0.7132	0.7076
SVM	0.6744	0.6572	0.6486	0.6444	0.6458
NeuralNet	0.6493	0.6493	0.6507	0.6438	0.6458
Neighbor1NN	0.6763	0.6517	0.6556	0.6514	0.6549
MC	0.6762	0.6495	0.6507	0.6493	0.6507
Baseline	0.6763	0.6544	0.6535	0.6556	0.6549
MIKNN	0.6539	0.5888	0.6076	0.584	0.5958
Density-0.1 human datasets	0.1cM	0.2cM	0.3cM	0.4cM	0.5cM
fastPHASE	0.7943	0.7782	0.7737	0.7725	0.7725
NN	0.8193	0.8121	0.8066	0.8045	0.8021
WeightedNN	0.8141	0.7986	0.7851	0.7736	0.7662
SVM	0.7543	0.7451	0.7387	0.7329	0.7338
NeuralNet	0.6323	0.6323	0.6314	0.6285	0.6261
Neighbor1NN	0.6527	0.654	0.6533	0.6529	0.6534
MC	0.7127	0.7049	0.7028	0.7011	0.7016
Baseline	0.6578	0.6563	0.6543	0.6544	0.654
MIKNN	0.7462	0.7216	0.7108	0.707	0.7136
Density-1 human datasets	0.01cM	0.02cM	0.03cM	0.04cM	0.05cM
fastPHASE	0.9682	0.9632	0.9602	0.9588	0.9579
NN	0.9391	0.9302	0.9214	0.9147	0.9064
WeightedNN	0.9372	0.9276	0.9158	0.9053	0.8955
SVM	0.9097	0.9095	0.9046	0.8982	0.8914
NeuralNet	0.7688	0.7835	0.7869	0.79	0.7905
Neighbor1NN	0.6303	0.6401	0.6441	0.6459	0.6469
MC	0.7757	0.7735	0.7719	0.7711	0.7703
Baseline	0.6504	0.6504	0.6504	0.6506	0.6505
MIKNN	0.922	0.9165	0.9107	0.9072	0.9051

From Table 5.11 and Figure 5.7, one can see that the genetic distance threshold does play a role in the imputation on the human datasets, causing the average imputation accuracy to vary a signifi-

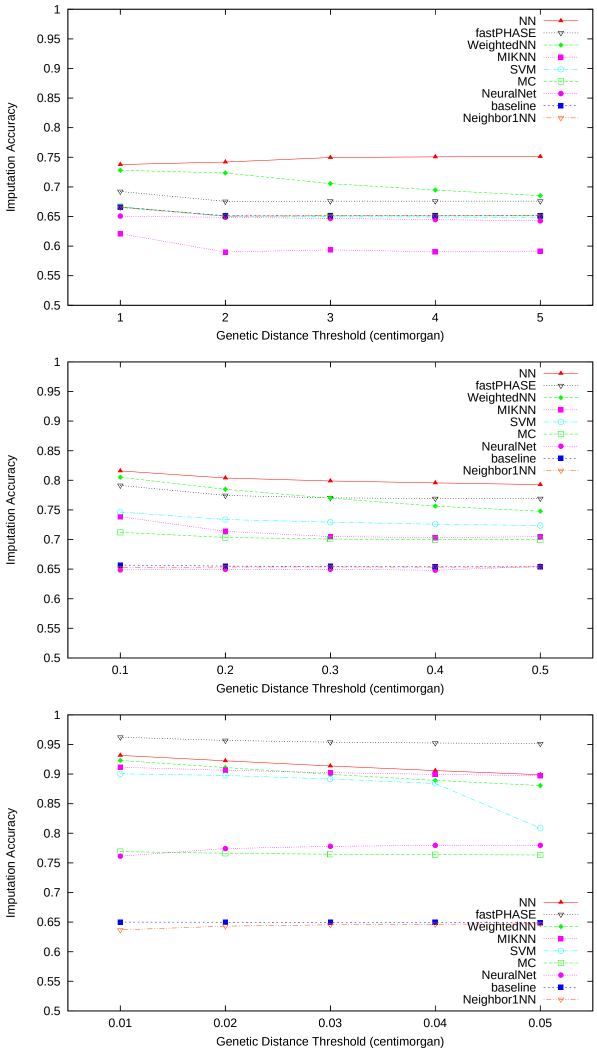

Figure 5.7: Average imputation accuracies on the human datasets at three density levels, 0.01, 0.1, and 1, respectively, with missing rate 0.5%, where the imputation methods were run with five corresponding genetic distance thresholds.

cant percentage up to 5. Another interesting pattern, also holds at the other five missing rates, can be seen from the table and plots is that there is no unique threshold that works the best for all methods. Indeed, perhaps a better way is to learn a suitable threshold for each imputation beforehand. Note that fastPHASE (as well as NPUTE and BaseLine) does not do imputation based on any genetic distance threshold. Yet one might have seen its average imputation accuracies changing throughout as listed in Table 5.11. Here the reason is that different genetic distance thresholds change the numbers of target missing SNP genotype values, since some masked values would not have sufficiently many neighboring SNP loci and thus excluded for performance evaluation. Recall that we imposed an additional constraint on the minimum of 4 neighboring SNP loci.

Table 5.12 summarizes the average imputation accuracies, each over the associated 10 simu-

Table 5.12: Average imputation accuracies on the mouse datasets at three density levels with missing rate 0.5%, where the imputation methods were run with five corresponding genetic distance thresholds.

Density-0.01 mouse datasets	0.2cM	0.4cM	0.6cM	0.8cM	1.0cM
fastPHASE	0.9099	0.9097	0.9118	0.9111	0.9111
NPUTE	0.8344	0.836	0.8393	0.838	0.838
NN	0.9119	0.9192	0.9285	0.9278	0.9241
WeightedNN	0.913	0.923	0.9294	0.9315	0.9231
SVM	0.8445	0.8642	0.8718	0.875	0.8731
NeuralNet	0.8297	0.8304	0.8324	0.8361	0.8315
Neighbor1NN	0.7338	0.776	0.7975	0.8065	0.8093
MC	0.8393	0.8358	0.8393	0.8417	0.838
Baseline	0.82	0.8196	0.8235	0.8213	0.8222
MIKNN	0.8927	0.9002	0.9072	0.8991	0.8981
Density-0.1 mouse datasets	0.02cM	0.04cM	0.06cM	0.08cM	0.1cM
fastPHASE	0.9354	0.9339	0.9326	0.9326	0.9327
NPUTE	0.8743	0.8733	0.8724	0.8724	0.8725
NN	0.9351	0.9417	0.9428	0.943	0.9431
WeightedNN	0.9386	0.9435	0.9447	0.9447	0.9444
SVM	0.8692	0.883	0.8885	0.8936	0.8962
NeuralNet	0.8429	0.854	0.8675	0.8774	0.8819
Neighbor1NN	0.7186	0.7489	0.7659	0.7828	0.7919
MC	0.8695	0.8672	0.8658	0.8625	0.8679
Baseline	0.8289	0.8255	0.8239	0.824	0.8238
MIKNN	0.9249	0.9265	0.9288	0.928	0.9274
Density-1 mouse datasets	0.002cM	0.004cM	0.006cM	0.008cM	0.01cM
fastPHASE	0.9441	0.9441	0.944	0.9439	0.9438
NPUTE	0.8818	0.882	0.882	0.8819	0.8817
NN	0.9459	0.95	0.9515	0.9522	0.9522
WeightedNN	0.9474	0.9515	0.9525	0.9528	0.9527
SVM	0.8896	0.903	0.9065	0.909	0.9095
NeuralNet	0.8497	0.857	0.8559	0.8558	0.8537
Neighbor1NN	0.705	0.7411	0.7588	0.7695	0.777
MC	0.8743	0.8735	0.8731	0.8729	0.8726
Baseline	0.8175	0.8179	0.8178	0.8182	0.8181
MIKNN	0.9383	0.9416	0.9421	0.9425	0.9425

lated datasets, of the imputation methods on the mouse datasets at three density levels with missing rate 0.5%. Figure 5.8 plots these average imputation accuracies. From them, we are able to draw the analogous conclusions that the genetic distance threshold does play a role in the missing SNP haplotype allele imputation, and that there is no unique threshold that works the best for all methods.

5.7 Imputation Speed Comparison

For imputation time comparison, we reported here the performance of all methods on the human and mouse datasets at all three density levels with missing rate 0.5%. These running time were collected on our “Heldar” CPU cluster, which has the following specifications: (1) Dual AMD Opteron 2350 quad core 64-bit CPU’s, (2) The CPU’s run at 2.0 GHz, have an 800 MHz HyperTransport bus, with a primary cache of 64KB I + 64KB D per core, a secondary cache of 512 KB I+D per core,

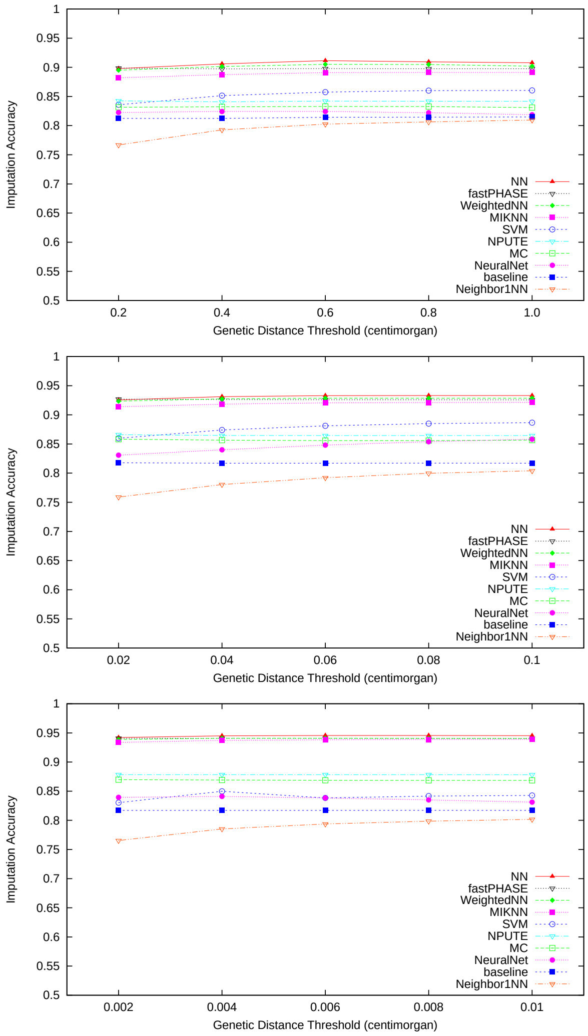

Figure 5.8: Average imputation accuracies on the mouse datasets at three density levels, 0.01, 0.1, and 1, respectively, with missing rate 0.5%, where the imputation methods were run with five corresponding genetic distance thresholds.

and a 2MB L3 cache per chip. Table 5.13 lists the average running time for each approach over the 10 simulated instances, where the genetic distance thresholds are 5cM, 0.5cM, and 0.05cM for the density-0.01, -0.1, and -1 human datasets, respectively, and 1cM, 0.1cM, and 0.01cM for the density-0.01, -0.1, and -1 mouse datasets, respectively.

As plotted in Figure 5.9, fastPHASE was the most time-consuming approach among all ten methods on both the human and mouse datasets, because of its internal EM algorithm. On a simulated high density human dataset, fastPHASE took around one day to finish. SVM also needed a relatively longer time during the training process to find the optimal parameters C and γ. The imputation of NPUTE is divided into two phases, training to identify the best window size and the real imputation. We reported the training time and the real imputation time separately on the mouse datasets, which are shown in Table 5.13 as before and after the $+$ sign. Recall that NPUTE does not

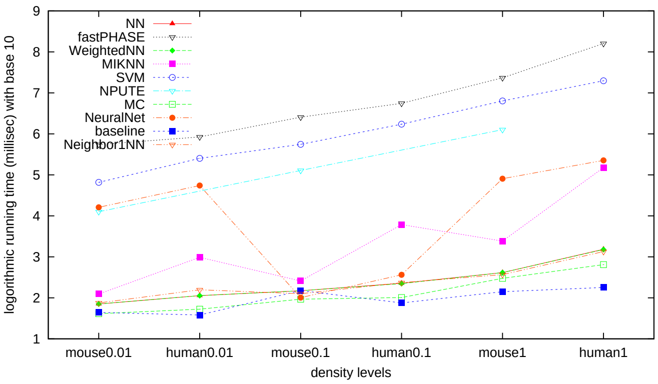

Figure 5.9: Running time comparison for the imputation methods on the human and mouse datasets at all three density levels with missing rate 0.5%. Here y-axis is the time at the logarithmic scale of base 10. fastPHASE and SVM were the most and the second most time-consuming methods.

work on the human datasets.

Table 5.13: Running time comparison for the imputation methods on the human and mouse datasets at all three density levels with missing rate 0.5%.

	Density-0.01 human	Density-0.01 mouse
fastPHASE	14m2.792s	9m9.141s
NPUTE	–	0m12.455s + 0m0.153s
NN	0m0.113s	0m0.071s
WeightedNN	0m0.113s	0m0.071s
SVM	4m13.124s	1m5.563s
NeuralNet	0m54.895s	0m16.150s
Neighbor1NN	0m0.157s	0m0.075s
MC	0m0.053s	0m0.041s
BaseLine	0m0.038s	0m0.044s
MIKNN	0m0.975s	0m0.124s

	Density-0.1 human	Density-0.1 mouse
fastPHASE	92m36.385s	42m36.845s
NPUTE	–	2m7.661s + 0m0.683s
NN	0m0.227s	0m0.149s
WeightedNN	0m0.222s	0m0.1494s
SVM	28m44.012s	9m15.495s
NeuralNet	0m0.364s	0m0.101s
Neighbor1NN	0m0.2346s	0m0.126s
MC	0m0.1014s	0m0.0929s
BaseLine	0m0.075s	0m0.151s
MIKNN	0m6.137s	0m0.259s

	Density-1 human	Density-1 mouse
fastPHASE	2667m12.436s	389m22.608s
NPUTE	–	21m1.529s + 0m4.763s
NN	0m1.532s	0m0.412s
WeightedNN	0m1.504s	0m0.4126s
SVM	329m2.821s	106m7.214s
NeuralNet	3m45.352s	1m20.809s
Neighbor1NN	0m1.348s	0m0.369s
MC	0m0.6458s	0m0.2994s
BaseLine	0m0.179s	0m0.141s
MIKNN	2m30.568s	0m2.4266s

Chapter 6

Conclusions

We have investigated the use of different machine learning approaches to tackle the missing SNP value imputation problem for SNP datasets generated by the current high-throughput genotyping technologies. Those missing values in the datasets can severely confound the downstream GWAS. We implemented nearest neighbor (NN) and its variants (WeightedNN and MIKNN), neural network (NeuralNet), SVM, and first order Markov chains (MC) to impute the missing values locally. In this dissertation, we focused on the direct SNP missing genotype imputation and the missing SNP haplotype allele imputation, the latter is also regarded as the post-haplotyping imputation, both without using reference haplotype panels.

For the local imputation, we introduced the use of genetic distance threshold to define the covering window for the target missing value, and use the known SNP values inside the window as features for inferring the missing values. We firstly non-trivially extended NPUTE [25] based on a fast k-nearest neighbor algorithm for both direct missing SNP genotype imputation and missing SNP haplotype allele imputation. We observed that from the genetic map [10] that the distribution of SNP loci is not uniform along the genome, and thus SNPs at different loci contribute differently to the target missing value imputation since their genetic distance to the target locus varies. NPUTE does not address this issue, but uses a fixed window size obtained from its training phase. We instead presented a *local* nearest neighbor NN in which the covering window size is determined by the genetic distance threshold, and two weighted variants WeightedNN and MIKNN, two *local* first order Markov chains, a *local* SVM with the RBF kernel, and a *local* neural networks that are constructed using the genotypes inside the window. Apparently, this is an improvement on using windows over NPUTE, that the covering window size is derived from a genetic distance threshold to the target missing SNP locus — *locality*.

Throughout our studies, we found out that, on low to medium density SNP datasets, our proposed methods NN and its weighted variant WeightedNN outperformed the currently best imputation programs fastPHASE [26] and NPUTE, in terms of missing SNP genotype and missing SNP haplotype allele imputation accuracy. Moreover, our methods are way faster than fastPHASE. On high-density SNP datasets, fastPHASE maintains to be the winner, achieving the highest missing SNP genotype imputation accuracy, which confirmed their claim of addressing haplotyping and imputation for high-density population SNP data. But when it comes to missing SNP haplotype allele imputation, our NN and WeightedNN again win out.

NPUTE is deigned for missing SNP haplotype allele imputation only, and does not work for missing SNP genotype imputation. Surprisingly, we found out its performance in our experiments does not catch up with what it is claim in its paper [25].

To conclude, for missing SNP haplotype allele imputation problem, our methods NN and WeightedNN are recommended, as they always won out in our extensive simulation studies. For missing SNP genotype imputation problem, when the density of a SNP dataset is high enough, fastPHASE should be used; in the other cases, use our NN and WeightedNN. For the currently hot topic of genome-wide SNP imputation, where some reference haplotype panels might exist, some sampling individuals have been high-density genotyped, and the other individuals are low (to medium) density genotyped due to cost consideration, and the goal is to impute these low density genotyped individuals, it seems that none of existing imputation methods can be convincingly employed. Our next step is to develop a novel framework for this genome-wide SNP imputation, based on fastPHASE and experience and lessons we learned from this dissertation work.

Bibliography

[1] D. Altshuler, Daly M. J., and E. S. Lander. Genetic mapping in human disease. *Science*, 323:881– 888, 2005.

[2] P. J. Avery and D. A. Henderson. Fitting Markov chain models to discrete state series such as DNA sequences. *Journal of Applied Statistics*, 48:53–61, 1999.

[3] P. Baldi and S. Brunak. *Bioinformatics The Machine Learning Approach (2nd Edition)*. MIT press, 2001.

[4] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[5] D. Botstein and N. Risch. Discovering genotypes underlying human phenotypes: past success for Mendelian disease, future approaches for complex disease. *Nature Genetics Supplement*, 33:228–237, 2003.

[6] A. J. Brrokes. The essence of SNPs. *Gene*, 234:177–186, 1999.

[7] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[8] F. S. Collins and L. D. Chakravarti. A dna polymorphism discovery resource for research on human genetic variation. *Genome Research*, 8:1229–1231, 1998.

[9] International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome, 2001.

[10] The International Consortium. A haplotype map of the human genome. http://www.hapmap.org, 2005.

[11] J. Y. Dai, I. Runcinski, M. LeBlanc, and C. Kooperberg. Imputation methods to improve inference in SNP association studies. *Genetic Epidemiology*, 30:690–702, 2006.

[12] E. Eskin *et al.* Efficient reconstruction of haplotype structure via perfect phylogeny. *Journal of Bioinformatics and Computational Biology*, 1:1–20, 2003.

[13] M. Huentelman *et al.* SNiPer: improved SNP genotype calling for Affymetrix 10K GeneChip microarray data. *BMC Genomics*, 6:149, 2005.

[14] S. J. Kang *et al.* Tradeoff between no-call reduction in genotyping error rate and loss of sample size for genetic case/control association studies. *The Pacific Symposium on Biocomputing*, 9:116–127, 2004.

[15] Z. S. Qin *et al.* Partition-ligation-expectation maximization algorithm for haplotype inference with single nucleotide polymorphisms. *American Journal of Genetics*, 71:1242–1247, 2002.

[16] L. Excoffier and M. Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution*, 12:921–927, 1995.

[17] W. Fu, Yi. Wang, Ying Wang, R. Li, R. Lin, and L. Jin. Missing call bias in high-throughput genotyping. *BMC Genomics*, 10:106, 2009.

[18] A. Griffiths, D. Suzuki J. Miller, R. Lewontin, and W. Gelbart. *An Introduction to Genetic Analysis (7th Edition)*. W. H. Freeman, 2000.

[19] L. Huang, Y. Li, A. B. Singleton, J. A. Hardy, G. Abecasis, N. A. Rosenberg, and P. Scheet. Genotype-imputation accuracy across worldwide human populations. *American Journal of Human Genetics*, 84:235–250, 2009.

[20] S. Lin, A. Chakravarti, and D. J. Cutler. Haplotype and missing data inference in nuclear families. *Genome Research*, 14:1624–1632, 2004.

[21] S. Lin, D. J. Cutler, M. E. Zwick, and A. Chakravarti. Haplotype inference in random population samples. *American Journal of Human Genetics*, 71:1129–1137, 2002.

[22] J. Marchini, B. Howie, S. Myers, G. McVean, and P. Donnelly. A new multipoint method for genome-wide association studies by imputation of genotypes. *Nature Genetics*, 39:906–913, 2006.

[23] T. Niu, Z. Qin, X. Xu, and J.S. Liu. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *American Journal of Genetics*, 70:157–169, 2002.

[24] J. Perkel. SNP genotyping: six technologies that keyed a revolution. *Nature Methods*, 5:447–453, 2008.

[25] A. Roberts, L. McMillan, W. Wang, J. Parker, I. Rusyn, and D. Threagal. Inferring missing genotypes in large SNP panels using fast nearest-neighbor searches over sliding windows. *Bioinformatics*, 23:i401–i407, 2007.

[26] P. Scheet and M. Stephens. A fast and flexible statistical model for large-scale population genotype data: Applications to inferring missing genotypes and haplotypic phase. *American Journal of Genetics*, 78:629–644, 2006.

[27] C. Sinoquet. Iterative two-pass algorithm for missing data imputation in SNP arrays. *Journal of Bioinformatics and Computational Biology*, 7:833–852, 2009.

[28] M. Stephens and P. Donnelly. A comparison of Bayesian methods for haplotype reconstruction from population genotype data. *American Journal of Human Genetics*, 73:1162–1169, 2003.

[29] M. Stephens and P. Donnely. A new statistical method for haplotype reconstruction from population data. *American Journal of Genetics*, 68:978 – 989, 2001.

[30] M. Stephens and P. Scheet. Accounting for decay of linkage disequilibrium in haplotype inference and missing-data imputation. *American Journal of Genetics*, 76:449–462, 2005.

[31] S. Su, C.-C. J. Kuo, and T. Chen. Inference of missing SNPs and information quantity measurements for haplotype blocks. *Bioinformatics*, 21:2001–2007, 2005.

[32] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, 1998.

[33] N. Waddell. Microarray-based DNA profiling to study genomic aberrations. *Life*, 60:437–440, 2008.

[34] J.-Y. Xu, G.-B. Xu, and S.-L. Chen. A new method for SNP discovery. *BioTechniques*, 46:201–208, 2009.

Appendix A

More Experimental Results

A.1 Additional p-values

Table A.1: The right-tailed t-test p-values for pairwise comparisons on the human datasets at three density levels, with missing rate 1, where the hypothesis is the average imputation accuracy of a row method is greater than the average imputation accuracy of a column method.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	Density-0.01 human datasets								
(1) fastPHASE	0.500	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	0.000	0.500	0.001	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	0.000	0.999	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(4) SVM	1.000	1.000	1.000	0.500	0.214	0.679	0.557	0.718	0.000
(5) NeuralNet	1.000	1.000	1.000	0.786	0.500	0.903	0.835	0.921	0.000
(6) Neighbor1NN	1.000	1.000	1.000	0.321	0.097	0.500	0.369	0.548	0.000
(7) MC	1.000	1.000	1.000	0.443	0.165	0.631	0.500	0.674	0.000
(8) BaseLine	1.000	1.000	1.000	0.282	0.079	0.452	0.326	0.500	0.000
(9) MIKNN	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.500
	Density-0.1 human datasets								
(1) fastPHASE	0.500	1.000	0.996	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	0.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	0.004	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(4) SVM	1.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000
(5) NeuralNet	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000	1.000
(6) Neighbor1NN	1.000	1.000	1.000	1.000	0.000	0.500	1.000	0.773	1.000
(7) MC	1.000	1.000	1.000	1.000	0.000	0.000	0.500	0.000	1.000
(8) BaseLine	1.000	1.000	1.000	1.000	0.000	0.227	1.000	0.500	1.000
(9) MIKNN	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000	0.500
	Density-1 human datasets								
(1) fastPHASE	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	1.000	0.500	0.003	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	1.000	0.997	0.500	0.000	0.000	0.000	0.000	0.000	0.178
(4) SVM	1.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(5) NeuralNet	1.000	1.000	1.000	1.000	0.500	0.000	0.000	0.000	1.000
(6) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(7) MC	1.000	1.000	1.000	1.000	1.000	0.000	0.500	0.000	1.000
(8) BaseLine	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(9) MIKNN	1.000	1.000	0.822	0.000	0.000	0.000	0.000	0.000	0.500

Table A.2: The right-tailed t-test p-values for pairwise comparisons on the human datasets at three density levels, with missing rate 2%, where the hypothesis is the average imputation accuracy of a row method is greater than the average imputation accuracy of a column method.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	Density-0.01 human datasets								
(1) fastPHASE	0.500	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	0.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	0.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(4) SVM	1.000	1.000	1.000	0.500	0.317	0.574	0.871	0.571	0.000
(5) NeuralNet	1.000	1.000	1.000	0.683	0.500	0.752	0.956	0.747	0.000
(6) Neighbor1NN	1.000	1.000	1.000	0.426	0.248	0.500	0.829	0.498	0.000
(7) MC	1.000	1.000	1.000	0.129	0.044	0.171	0.500	0.173	0.000
(8) BaseLine	1.000	1.000	1.000	0.429	0.253	0.502	0.827	0.500	0.000
(9) MIKNN	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.500
	Density-0.1 human datasets								
(1) fastPHASE	0.500	1.000	0.268	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	0.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	0.732	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(4) SVM	1.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000
(5) NeuralNet	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000	1.000
(6) Neighbor1NN	1.000	1.000	1.000	1.000	0.000	0.500	1.000	0.831	1.000
(7) MC	1.000	1.000	1.000	1.000	0.000	0.000	0.500	0.000	1.000
(8) BaseLine	1.000	1.000	1.000	1.000	0.000	0.169	1.000	0.500	1.000
(9) MIKNN	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000	0.500
	Density-1 human datasets								
(1) fastPHASE	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	1.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.715
(4) SVM	1.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(5) NeuralNet	1.000	1.000	1.000	1.000	0.500	0.000	0.000	0.000	1.000
(6) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(7) MC	1.000	1.000	1.000	1.000	1.000	0.000	0.500	0.000	1.000
(8) BaseLine	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(9) MIKNN	1.000	1.000	0.285	0.000	0.000	0.000	0.000	0.000	0.500

Table A.3: The right-tailed t-test p-values for pairwise comparisons on the human datasets at three density levels, with missing rate 5%, where the hypothesis is the average imputation accuracy of a row method is greater than the average imputation accuracy of a column method.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	Density-0.01 human datasets								
(1) fastPHASE	0.500	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	0.000	0.500	0.001	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	0.000	0.999	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(4) SVM	1.000	1.000	1.000	0.500	0.214	0.679	0.557	0.718	0.000
(5) NeuralNet	1.000	1.000	1.000	0.786	0.500	0.903	0.835	0.921	0.000
(6) Neighbor1NN	1.000	1.000	1.000	0.321	0.097	0.500	0.369	0.548	0.000
(7) MC	1.000	1.000	1.000	0.443	0.165	0.631	0.500	0.674	0.000
(8) BaseLine	1.000	1.000	1.000	0.282	0.079	0.452	0.326	0.500	0.000
(9) MIKNN	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.500
	Density-0.1 human datasets								
(1) fastPHASE	0.500	1.000	0.996	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	0.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	0.004	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(4) SVM	1.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000
(5) NeuralNet	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000	1.000
(6) Neighbor1NN	1.000	1.000	1.000	1.000	0.000	0.500	1.000	0.773	1.000
(7) MC	1.000	1.000	1.000	1.000	0.000	0.000	0.500	0.000	1.000
(8) BaseLine	1.000	1.000	1.000	1.000	0.000	0.227	1.000	0.500	1.000
(9) MIKNN	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000	0.500
	Density-1 human datasets								
(1) fastPHASE	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	1.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.985
(4) SVM	1.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(5) NeuralNet	1.000	1.000	1.000	1.000	0.500	0.000	0.000	0.000	1.000
(6) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(7) MC	1.000	1.000	1.000	1.000	1.000	0.000	0.500	0.000	1.000
(8) BaseLine	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(9) MIKNN	1.000	1.000	0.015	0.000	0.000	0.000	0.000	0.000	0.500

Table A.4: The right-tailed t-test p-values for pairwise comparisons on the human datasets at three density levels, with missing rate 10%, where the hypothesis is the average imputation accuracy of a row method is greater than the average imputation accuracy of a column method.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	Density-0.01 human datasets								
(1) fastPHASE	0.500	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	0.000	0.500	0.001	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	0.000	0.999	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(4) SVM	1.000	1.000	1.000	0.500	0.214	0.679	0.557	0.718	0.000
(5) NeuralNet	1.000	1.000	1.000	0.786	0.500	0.903	0.835	0.921	0.000
(6) Neighbor1NN	1.000	1.000	1.000	0.321	0.097	0.500	0.369	0.548	0.000
(7) MC	1.000	1.000	1.000	0.443	0.165	0.631	0.500	0.674	0.000
(8) BaseLine	1.000	1.000	1.000	0.282	0.079	0.452	0.326	0.500	0.000
(9) MIKNN	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.500
	Density-0.01 human datasets								
(1) fastPHASE	0.500	1.000	0.996	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	0.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	0.004	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(4) SVM	1.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000
(5) NeuralNet	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000	1.000
(6) Neighbor1NN	1.000	1.000	1.000	1.000	0.000	0.500	1.000	0.773	1.000
(7) MC	1.000	1.000	1.000	1.000	0.000	0.000	0.500	0.000	1.000
(8) BaseLine	1.000	1.000	1.000	1.000	0.000	0.227	1.000	0.500	1.000
(9) MIKNN	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000	0.500
	Density-0.01 human datasets								
(1) fastPHASE	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	1.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.999
(4) SVM	1.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(5) NeuralNet	1.000	1.000	1.000	1.000	0.500	0.000	0.000	0.000	1.000
(6) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(7) MC	1.000	1.000	1.000	1.000	1.000	0.000	0.500	0.000	1.000
(8) BaseLine	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(9) MIKNN	1.000	1.000	0.001	0.000	0.000	0.000	0.000	0.000	0.500

Table A.5: The right-tailed t-test p-values for pairwise comparisons on the human datasets at three density levels, with missing rate 20%, where the hypothesis is the average imputation accuracy of a row method is greater than the average imputation accuracy of a column method.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	Density-0.01 human datasets								
(1) fastPHASE	0.500	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	0.000	0.500	0.001	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	0.000	0.999	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(4) SVM	1.000	1.000	1.000	0.500	0.214	0.679	0.557	0.718	0.000
(5) NeuralNet	1.000	1.000	1.000	0.786	0.500	0.903	0.835	0.921	0.000
(6) Neighbor1NN	1.000	1.000	1.000	0.321	0.097	0.500	0.369	0.548	0.000
(7) MC	1.000	1.000	1.000	0.443	0.165	0.631	0.500	0.674	0.000
(8) BaseLine	1.000	1.000	1.000	0.282	0.079	0.452	0.326	0.500	0.000
(9) MIKNN	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.500
	Density-0.1 human datasets								
(1) fastPHASE	0.500	1.000	0.996	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	0.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	0.004	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(4) SVM	1.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000
(5) NeuralNet	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000	1.000
(6) Neighbor1NN	1.000	1.000	1.000	1.000	0.000	0.500	1.000	0.773	1.000
(7) MC	1.000	1.000	1.000	1.000	0.000	0.000	0.500	0.000	1.000
(8) BaseLine	1.000	1.000	1.000	1.000	0.000	0.227	1.000	0.500	1.000
(9) MIKNN	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000	0.500
	Density-1 human datasets								
(1) fastPHASE	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NN	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(3) WeightedNN	1.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000	1.000
(4) SVM	1.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(5) NeuralNet	1.000	1.000	1.000	1.000	0.500	0.000	0.000	0.000	1.000
(6) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(7) MC	1.000	1.000	1.000	1.000	1.000	0.000	0.500	0.000	1.000
(8) BaseLine	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(9) MIKNN	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.500

Table A.6: The right-tailed t-test p-values for pairwise comparisons on the mouse datasets at three density levels, with missing rate 0.5%, where the hypothesis is the average imputation accuracy of a row method is greater than the average imputation accuracy of a column method.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	Density-0.01 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.002
(2) NPUTE	1.000	0.500	1.000	1.000	0.898	0.000	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.307	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	0.693	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.102	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	1.000	1.000	1.000	1.000	0.500	0.000	0.969	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.031	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	0.998	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500
	Density-0.1 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NPUTE	1.000	0.500	1.000	1.000	1.000	0.001	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.109	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	0.891	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	0.999	1.000	1.000	1.000	0.500	0.000	0.342	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.658	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500
	Density-1 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NPUTE	1.000	0.500	1.000	1.000	1.000	0.000	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	1.000	1.000	1.000	1.000	0.500	0.000	1.000	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.000	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500

Table A.7: The right-tailed t-test p-values for pairwise comparisons on the mouse datasets at three density levels, with missing rate 1%, where the hypothesis is the average imputation accuracy of a row method is greater than the average imputation accuracy of a column method.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	Density-0.01 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.002
(2) NPUTE	1.000	0.500	1.000	1.000	0.898	0.000	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.307	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	0.693	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.102	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	1.000	1.000	1.000	1.000	0.500	0.000	0.969	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.031	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	0.998	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500
	Density-0.1 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NPUTE	1.000	0.500	1.000	1.000	1.000	0.001	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.109	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	0.891	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	0.999	1.000	1.000	1.000	0.500	0.000	0.342	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.658	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500
	Density-1 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NPUTE	1.000	0.500	1.000	1.000	1.000	0.000	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	1.000	1.000	1.000	1.000	0.500	0.000	1.000	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.000	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500

Table A.8: The right-tailed t-test p-values for pairwise comparisons on the mouse datasets at three density levels, with missing rate 2%, where the hypothesis is the average imputation accuracy of a row method is greater than the average imputation accuracy of a column method.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	Density-0.01 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.002
(2) NPUTE	1.000	0.500	1.000	1.000	0.898	0.000	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.307	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	0.693	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.102	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	1.000	1.000	1.000	1.000	0.500	0.000	0.969	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.031	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	0.998	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500
	Density-0.1 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NPUTE	1.000	0.500	1.000	1.000	1.000	0.001	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.109	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	0.891	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	0.999	1.000	1.000	1.000	0.500	0.000	0.342	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.658	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500
	Density-1 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NPUTE	1.000	0.500	1.000	1.000	1.000	0.000	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	1.000	1.000	1.000	1.000	0.500	0.000	1.000	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.000	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500

Table A.9: The right-tailed t-test p-values for pairwise comparisons on the mouse datasets at three density levels, with missing rate 5%, where the hypothesis is the average imputation accuracy of a row method is greater than the average imputation accuracy of a column method.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	Density-0.01 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.002
(2) NPUTE	1.000	0.500	1.000	1.000	0.898	0.000	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.307	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	0.693	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.102	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	1.000	1.000	1.000	1.000	0.500	0.000	0.969	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.031	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	0.998	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500
	Density-0.1 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NPUTE	1.000	0.500	1.000	1.000	1.000	0.001	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.109	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	0.891	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	0.999	1.000	1.000	1.000	0.500	0.000	0.342	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.658	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500
	Density-1 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NPUTE	1.000	0.500	1.000	1.000	1.000	0.000	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	1.000	1.000	1.000	1.000	0.500	0.000	1.000	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.000	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500

Table A.10: The right-tailed t-test p-values for pairwise comparisons on the mouse datasets at three density levels, with missing rate 10%, where the hypothesis is the average imputation accuracy of a row method is greater than the average imputation accuracy of a column method.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	Density-0.01 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.002
(2) NPUTE	1.000	0.500	1.000	1.000	0.898	0.000	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.307	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	0.693	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.102	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	1.000	1.000	1.000	1.000	0.500	0.000	0.969	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.031	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	0.998	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500
	Density-0.1 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NPUTE	1.000	0.500	1.000	1.000	1.000	0.001	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.109	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	0.891	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	0.999	1.000	1.000	1.000	0.500	0.000	0.342	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.658	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500
	Density-1 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NPUTE	1.000	0.500	1.000	1.000	1.000	0.000	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	1.000	1.000	1.000	1.000	0.500	0.000	1.000	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.000	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500

Table A.11: The right-tailed t-test p-values for pairwise comparisons on the mouse datasets at three density levels, with missing rate 20%, where the hypothesis is the average imputation accuracy of a row method is greater than the average imputation accuracy of a column method.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	Density-0.01 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.002
(2) NPUTE	1.000	0.500	1.000	1.000	0.898	0.000	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.307	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	0.693	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.102	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	1.000	1.000	1.000	1.000	0.500	0.000	0.969	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.031	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	0.998	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500
	Density-0.1 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NPUTE	1.000	0.500	1.000	1.000	1.000	0.001	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.109	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	0.891	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	0.999	1.000	1.000	1.000	0.500	0.000	0.342	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.658	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500
	Density-1 mouse datasets									
(1) fastPHASE	0.500	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
(2) NPUTE	1.000	0.500	1.000	1.000	1.000	0.000	0.000	0.000	0.000	1.000
(3) NN	0.000	0.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(4) WeightedNN	0.000	0.000	1.000	0.500	0.000	0.000	0.000	0.000	0.000	0.000
(5) SVM	1.000	0.000	1.000	1.000	0.500	0.000	0.000	0.000	0.000	1.000
(6) NeuralNet	1.000	1.000	1.000	1.000	1.000	0.500	0.000	1.000	0.000	1.000
(7) Neighbor1NN	1.000	1.000	1.000	1.000	1.000	1.000	0.500	1.000	1.000	1.000
(8) MC	1.000	1.000	1.000	1.000	1.000	0.000	0.000	0.500	0.000	1.000
(9) BaseLine	1.000	1.000	1.000	1.000	1.000	1.000	0.000	1.000	0.500	1.000
(10) MIKNN	1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.500

A.2 Best Imputation Accuracy versus Missing Rate

For each of the 36 combinations of (species, density, missing rate), there are 10 associated simulated datasets; on each simulated datasets, 5 genetic distance thresholds are set to run the imputation methods. Among the 50 imputation accuracies for each imputation method, the best one is reported in the following tables.

Table A.12: Best imputation accuracies at the 6 missing rates across the associated 10 density-0.01 human datasets.

Methods	Missing Rate					
	0.5%	1%	2%	5%	10%	20%
fastPHASE	0.7407	0.7547	0.7483	0.6946	0.6938	0.6922
NN	0.8264	0.8056	0.8024	0.7715	0.7507	0.7465
WeightedNN	0.81	0.7703	0.7865	0.7553	0.7299	0.7247
SVM	0.76	0.7204	0.7098	0.6922	0.6732	0.6755
NeuralNet	0.7153	0.6944	0.7158	0.6724	0.6627	0.6575
Neighbor1NN	0.75	0.7188	0.7211	0.6864	0.6722	0.6734
MC	0.7545	0.7204	0.7098	0.6804	0.6788	0.6791
BaseLine	0.76	0.724	0.7188	0.6855	0.6731	0.6737
MIKNN	0.72	0.6462	0.6538	0.6247	0.623	0.6138

Table A.13: Best imputation accuracies at the 6 missing rates across the associated 10 density-0.1 human datasets.

Methods	Missing Rate					
	0.5%	1%	2%	5%	10%	20%
fastPHASE	0.8081	0.8161	0.8097	0.8029	0.7911	0.7752
NN	0.8392	0.8403	0.8307	0.8257	0.8155	0.8017
WeightedNN	0.8295	0.8394	0.8205	0.8125	0.7994	0.7801
SVM	0.7696	0.7714	0.7635	0.7605	0.7432	0.7297
NeuralNet	0.6646	0.6868	0.6391	0.6869	0.6737	0.671
Neighbor1NN	0.6764	0.6789	0.6619	0.6638	0.66	0.6589
MC	0.7339	0.7338	0.7271	0.7226	0.7142	0.7084
BaseLine	0.6762	0.6791	0.6636	0.6674	0.6621	0.6616
MIKNN	0.7625	0.7732	0.7608	0.7497	0.7376	0.7175

Table A.14: Best imputation accuracies at the 6 missing rates across the associated 10 density-1 human datasets.

Methods	Missing Rate					
	0.5%	1%	2%	5%	10%	20%
fastPHASE	0.97	0.9708	0.9682	0.9672	0.9634	0.9473
NN	0.941	0.941	0.9376	0.9348	0.9276	0.9156
WeightedNN	0.9398	0.9384	0.9339	0.9259	0.914	0.8952
SVM	0.912	0.9131	0.9107	0.906	0.896	0.8752
NeuralNet	0.7965	0.7934	0.7889	0.7839	0.7741	0.758
Neighbor1NN	0.652	0.6495	0.6493	0.6507	0.6486	0.6482
MC	0.7808	0.7791	0.7775	0.7738	0.7669	0.7528
BaseLine	0.6571	0.6535	0.6534	0.6543	0.6508	0.6505
MIKNN	0.924	0.9224	0.9205	0.9153	0.9054	0.8888

Table A.15: Best imputation accuracies at the 6 missing rates across the associated 10 density-0.01 mouse datasets.

Methods	Missing Rate					
	0.5%	1%	2%	5%	10%	20%
fastPHASE	0.9537	0.9401	0.9164	0.9117	0.9079	0.8888
NPUTE	0.913	0.9032	0.8776	0.8656	0.8501	0.8332
NN	0.9722	0.9585	0.9332	0.9224	0.9139	0.8905
WeightedNN	0.963	0.9631	0.9393	0.9169	0.9061	0.8795
SVM	0.9259	0.9217	0.8779	0.8738	0.8725	0.8508
NeuralNet	0.9065	0.8802	0.8525	0.8425	0.8385	0.8218
Neighbor1NN	0.8704	0.8618	0.8203	0.8278	0.8242	0.8199
MC	0.8981	0.8726	0.8685	0.8593	0.8443	0.8379
BaseLine	0.9149	0.8618	0.8289	0.8332	0.8251	0.8215
MIKNN	0.963	0.9539	0.9122	0.9052	0.8997	0.8788

Table A.16: Best imputation accuracies at the 6 missing rates across the associated 10 density-0.01 mouse datasets.

Methods	Missing Rate					
	0.5%	1%	2%	5%	10%	20%
fastPHASE	0.9514	0.9381	0.9357	0.9315	0.9277	0.9151
NPUTE	0.8978	0.8783	0.8749	0.8727	0.8661	0.857
NN	0.9585	0.9492	0.9441	0.9379	0.9314	0.9153
WeightedNN	0.9575	0.9497	0.9423	0.9315	0.9216	0.9079
SVM	0.9095	0.9036	0.9005	0.8923	0.8862	0.8678
NeuralNet	0.895	0.8943	0.8871	0.8827	0.8316	0.8212
Neighbor1NN	0.8065	0.8183	0.8116	0.8159	0.8149	0.8125
MC	0.8849	0.8778	0.8689	0.8626	0.8584	0.8472
BaseLine	0.8527	0.8386	0.825	0.8203	0.8192	0.8161
MIKNN	0.9398	0.9359	0.9308	0.9269	0.9218	0.9099

Table A.17: Best imputation accuracies at the 6 missing rates across the associated 10 density-0.01 mouse datasets.

Methods	Missing Rate					
	0.5%	1%	2%	5%	10%	20%
fastPHASE	0.948	0.9501	0.9461	0.9436	0.9391	0.932
NPUTE	0.8915	0.8885	0.885	0.881	0.8773	0.8706
NN	0.957	0.9557	0.9523	0.9485	0.942	0.9311
WeightedNN	0.9577	0.9546	0.9504	0.9418	0.9331	0.9245
SVM	0.9139	0.9131	0.9102	0.8986	0.8983	0.8827
NeuralNet	0.8648	0.8617	0.8524	0.8424	0.8339	0.8216
Neighbor1NN	0.7853	0.7967	0.8061	0.8124	0.8147	0.8127
MC	0.8829	0.8814	0.8765	0.8723	0.8673	0.8568
BaseLine	0.8247	0.8234	0.8195	0.8182	0.8187	0.8158
MIKNN	0.9503	0.9488	0.9443	0.9408	0.9366	0.9286

A.3 Imputation Accuracy versus Density Level

Table 5.9 and the following Tables A.18–A.22 are rearrangements of Tables 5.2, 5.3, and 5.4; Likewise, Table 5.10 and the following Tables A.23–A.27 are rearrangements of Tables 5.5, 5.6, and 5.7. These tables are assembled to see the effects on SNP density level on the imputation accuracy.

Table A.18: Average imputation accuracies on the human datasets at three density levels with missing rate 1%. Data reproduced from Tables 5.2, 5.3, and 5.4.

	Human datasets with missing rate 1%		
	0.01	0.1	1
fastPHASE	0.6782	0.7797	0.9616
NN	0.7521	0.8079	0.9214
WeightedNN	0.7161	0.7859	0.9135
SVM	0.6544	0.7395	0.9021
NeuralNet	0.6431	0.6342	0.7834
Neighbor1NN	0.653	0.6532	0.6419
MC	0.6505	0.7053	0.7718
BaseLine	0.6535	0.6552	0.6496
MIKNN	0.5945	0.7232	0.9112

Table A.19: Average imputation accuracies on the human datasets at three density levels with missing rate 2%. Data reproduced from Tables 5.2, 5.3, and 5.4.

	Human datasets with missing rate 2%		
	0.01	0.1	1
fastPHASE	0.6785	0.7838	0.9611
NN	0.749	0.8056	0.9189
WeightedNN	0.7102	0.7817	0.9091
SVM	0.6488	0.7385	0.8918
NeuralNet	0.6465	0.6274	0.7804
Neighbor1NN	0.6497	0.6493	0.6427
MC	0.654	0.7053	0.7705
BaseLine	0.6497	0.6508	0.6493
MIKNN	0.6032	0.7213	0.9104

Table A.20: Average imputation accuracies on the human datasets at three density levels with missing rate 5%. Data reproduced from Tables 5.2, 5.3, and 5.4.

	Human datasets with missing rate 5%		
	0.01	0.1	1
fastPHASE	0.6787	0.78	0.9597
NN	0.7471	0.8041	0.9158
WeightedNN	0.7065	0.774	0.901
SVM	0.6587	0.7351	0.8693
NeuralNet	0.6528	0.6732	0.7769
Neighbor1NN	0.6585	0.6553	0.6452
MC	0.6606	0.7065	0.768
BaseLine	0.6588	0.6567	0.6502
MIKNN	0.598	0.7156	0.9062

Table A.21: Average imputation accuracies on the human datasets at three density levels with missing rate 10%. Data reproduced from Tables 5.2, 5.3, and 5.4.

	Human datasets with missing rate 10%		
	0.01	0.1	1
fastPHASE	0.6756	0.7716	0.9502
NN	0.7393	0.7969	0.91
WeightedNN	0.6936	0.7624	0.891
SVM	0.6488	0.7261	0.8715
NeuralNet	0.6432	0.6706	0.769
Neighbor1NN	0.6507	0.6554	0.6455
MC	0.6513	0.7026	0.7614
BaseLine	0.6511	0.6566	0.6492
MIKNN	0.5919	0.7077	0.898

Table A.22: Average imputation accuracies on the human datasets at three density levels with missing rate 20%. Data reproduced from Tables 5.2, 5.3, and 5.4.

	Human datasets with missing rate 20%		
	0.01	0.1	1
fastPHASE	0.6735	0.7566	0.9378
NN	0.7349	0.7848	0.8984
WeightedNN	0.6877	0.7454	0.8734
SVM	0.6516	0.7104	0.8217
NeuralNet	0.6463	0.6651	0.7534
Neighbor1NN	0.6541	0.6536	0.6456
MC	0.6528	0.6946	0.7483
BaseLine	0.6542	0.6549	0.6488
MIKNN	0.5902	0.6908	0.882

Table A.23: Average imputation accuracies on the mouse datasets at three density levels with missing rate 1%. Data reproduced from Tables 5.5, 5.6, and 5.7.

	Mouse datasets with missing rate 1%		
	0.01	0.1	1
fastPHASE	0.9072	0.9314	0.9452
NPUTE	0.856	0.8684	0.8821
NN	0.9184	0.9398	0.9511
WeightedNN	0.9197	0.9407	0.9508
SVM	0.854	0.8842	0.9042
NeuralNet	0.8271	0.8629	0.8524
Neighbor1NN	0.7844	0.768	0.7697
MC	0.8319	0.8635	0.8753
BaseLine	0.8133	0.8189	0.8182
MIKNN	0.9035	0.9271	0.9425

Table A.24: Average imputation accuracies on the mouse datasets at three density levels with missing rate 2%. Data reproduced from Tables 5.5, 5.6, and 5.7.

	Mouse datasets with missing rate 2%		
	0.01	0.1	1
fastPHASE	0.8968	0.9278	0.9439
NPUTE	0.8458	0.8661	0.8816
NN	0.9101	0.9355	0.9498
WeightedNN	0.9088	0.9343	0.9468
SVM	0.8507	0.8808	0.8969
NeuralNet	0.8217	0.8579	0.8478
Neighbor1NN	0.7816	0.7795	0.7887
MC	0.8276	0.8568	0.8737
BaseLine	0.8034	0.8155	0.8178
MIKNN	0.8887	0.923	0.9412

Table A.25: Average imputation accuracies on the mouse datasets at three density levels with missing rate 5%. Data reproduced from Tables 5.5, 5.6, and 5.7.

	Mouse datasets with missing rate 5%		
	0.01	0.1	1
fastPHASE	0.8997	0.9271	0.9417
NPUTE	0.8445	0.8659	0.8789
NN	0.9081	0.9324	0.9466
WeightedNN	0.9014	0.9267	0.9391
SVM	0.8581	0.878	0.6477
NeuralNet	0.8277	0.8549	0.8371
Neighbor1NN	0.8068	0.798	0.8044
MC	0.8384	0.8561	0.8705
BaseLine	0.8179	0.8145	0.8168
MIKNN	0.8902	0.9199	0.9386

Table A.26: Average imputation accuracies on the mouse datasets at three density levels with missing rate 10%. Data reproduced from Tables 5.5, 5.6, and 5.7.

	Mouse datasets with missing rate 10%		
	0.01	0.1	1
fastPHASE	0.8917	0.9244	0.9383
NPUTE	0.839	0.8617	0.8758
NN	0.899	0.927	0.9408
WeightedNN	0.8887	0.9177	0.9321
SVM	0.8495	0.875	0.8837
NeuralNet	0.8164	0.8254	0.8245
Neighbor1NN	0.8059	0.8072	0.8103
MC	0.8289	0.8538	0.8655
BaseLine	0.8117	0.8159	0.8167
MIKNN	0.883	0.9152	0.9343

Table A.27: Average imputation accuracies on the mouse datasets at three density levels with missing rate 20%. Data reproduced from Tables 5.5, 5.6, and 5.7.

	Mouse datasets with missing rate 20%		
	0.01	0.1	1
fastPHASE	0.8803	0.9125	0.9305
NPUTE	0.8281	0.8531	0.8695
NN	0.8808	0.9112	0.9294
WeightedNN	0.8678	0.9009	0.9213
SVM	0.8398	0.8597	0.7648
NeuralNet	0.809	0.8114	0.806
Neighbor1NN	0.8103	0.808	0.8105
MC	0.8267	0.8429	0.8553
BaseLine	0.8143	0.8134	0.8149
MIKNN	0.8666	0.9023	0.9252

A.4 Imputation Accuracy versus Genetic Distance Threshold

Table 5.11 summarizes the average imputation accuracies, each over the associated 10 simulated datasets, of the imputation methods on the human datasets at three density levels with missing rate 0.5%. The following five more tables on human datasets at five other missing rates, and six more tables on mouse datasets at six missing rates, show further the effects of genetic distance threshold on the imputation accuracy.

Table A.28: Average imputation accuracies on the human datasets at three density levels with missing rate 1%, where the imputation methods were run with five corresponding genetic distance thresholds.

Density-0.01 human datasets	1cM	2cM	3cM	4cM	5cM
fastPHASE	0.6915	0.6746	0.675	0.675	0.675
NN	0.7375	0.7406	0.7552	0.7615	0.766
WeightedNN	0.7159	0.7346	0.7125	0.7139	0.7035
SVM	0.6623	0.6536	0.6538	0.6528	0.6493
NeuralNet	0.6479	0.6438	0.6382	0.6444	0.6413
Neighbor1NN	0.6628	0.6504	0.6503	0.6497	0.6517
MC	0.6643	0.6466	0.6469	0.6479	0.6469
BaseLine	0.6647	0.6511	0.651	0.65	0.6507
MIKNN	0.6122	0.5912	0.5941	0.591	0.584
Density-0.1 human datasets	0.1cM	0.2cM	0.3cM	0.4cM	0.5cM
fastPHASE	0.7968	0.7793	0.7747	0.7739	0.7738
NN	0.8223	0.8109	0.8058	0.8022	0.7981
WeightedNN	0.8201	0.7978	0.7835	0.7677	0.7605
SVM	0.7527	0.7399	0.738	0.7353	0.7315
NeuralNet	0.626	0.6255	0.6247	0.6227	0.6718
Neighbor1NN	0.6502	0.6532	0.6539	0.6542	0.6545
MC	0.7142	0.7058	0.7027	0.702	0.7018
BaseLine	0.6558	0.6559	0.6552	0.6547	0.6543
MIKNN	0.7503	0.7243	0.7139	0.7135	0.7137
Density-1 human datasets	0.01cM	0.02cM	0.03cM	0.04cM	0.05cM
fastPHASE	0.9681	0.963	0.9603	0.9588	0.9578
NN	0.9389	0.9297	0.9205	0.9127	0.9053
WeightedNN	0.9363	0.9248	0.9126	0.9015	0.8922
SVM	0.9104	0.9086	0.9036	0.8974	0.8903
NeuralNet	0.7684	0.7824	0.7876	0.789	0.7897
Neighbor1NN	0.6325	0.6412	0.644	0.6455	0.6464
MC	0.7755	0.7728	0.771	0.7702	0.7696
BaseLine	0.6501	0.6497	0.6493	0.6496	0.6494
MIKNN	0.9205	0.9146	0.9099	0.907	0.9042

Table A.29: Average imputation accuracies on the human datasets at three density levels with missing rate 2%, where the imputation methods were run with five corresponding genetic distance thresholds.

Density-0.01 human datasets	1cM	2cM	3cM	4cM	5cM
fastPHASE	0.6942	0.6743	0.6747	0.6747	0.6747
NN	0.7411	0.7467	0.7516	0.7492	0.7562
WeightedNN	0.7357	0.7291	0.7095	0.6938	0.683
SVM	0.6625	0.647	0.6447	0.6444	0.6454
NeuralNet	0.6525	0.6458	0.647	0.6452	0.6421
Neighbor1NN	0.6621	0.6463	0.6461	0.6475	0.6466
MC	0.6645	0.6506	0.6516	0.6513	0.6516
BaseLine	0.6614	0.6459	0.647	0.6471	0.647
MIKNN	0.6281	0.604	0.5983	0.5948	0.5908
Density-0.1 human datasets	0.1cM	0.2cM	0.3cM	0.4cM	0.5cM
fastPHASE	0.8014	0.7828	0.7789	0.778	0.778
NN	0.8228	0.8063	0.8028	0.8007	0.7956
WeightedNN	0.8143	0.7943	0.7775	0.766	0.7566
SVM	0.7546	0.7391	0.7354	0.7334	0.7302
NeuralNet	0.629	0.6285	0.6291	0.6266	0.6237
Neighbor1NN	0.6483	0.6496	0.6498	0.6495	0.6492
MC	0.7161	0.7053	0.7024	0.7013	0.7012
BaseLine	0.6535	0.6503	0.6501	0.6499	0.65
MIKNN	0.7494	0.7215	0.7136	0.7112	0.711
Density-1 human datasets	0.01cM	0.02cM	0.03cM	0.04cM	0.05cM
fastPHASE	0.9677	0.9624	0.9598	0.9584	0.9575
NN	0.9363	0.9269	0.9183	0.9099	0.9029
WeightedNN	0.9325	0.9194	0.9082	0.8972	0.888
SVM	0.9089	0.9067	0.9019	0.8956	0.846
NeuralNet	0.7659	0.7794	0.7841	0.7861	0.7864
Neighbor1NN	0.6348	0.6421	0.6444	0.646	0.6463
MC	0.7742	0.7712	0.7697	0.7689	0.7683
BaseLine	0.6499	0.6494	0.6491	0.6491	0.6489
MIKNN	0.919	0.9134	0.9095	0.9063	0.9039

Table A.30: Average imputation accuracies on the human datasets at three density levels with missing rate 5%, where the imputation methods were run with five corresponding genetic distance thresholds.

Density-0.01 human datasets	1cM	2cM	3cM	4cM	5cM
fastPHASE	0.6884	0.676	0.6764	0.6764	0.6764
NN	0.7379	0.7452	0.7461	0.7533	0.753
WeightedNN	0.7339	0.7216	0.7033	0.6911	0.6826
SVM	0.6701	0.6555	0.6543	0.6582	0.6555
NeuralNet	0.656	0.6571	0.6541	0.6506	0.6461
Neighbor1NN	0.6672	0.6555	0.6567	0.6566	0.6566
MC	0.6699	0.6578	0.6584	0.6584	0.6583
BaseLine	0.6675	0.6561	0.657	0.6572	0.656
MIKNN	0.6149	0.5906	0.5903	0.5938	0.6002
Density-0.1 human datasets	0.1cM	0.2cM	0.3cM	0.4cM	0.5cM
fastPHASE	0.7969	0.7795	0.7753	0.7741	0.774
NN	0.8199	0.8063	0.8015	0.7979	0.795
WeightedNN	0.8081	0.7869	0.7707	0.7571	0.7471
SVM	0.7509	0.7369	0.7324	0.7287	0.7266
NeuralNet	0.6703	0.6738	0.6746	0.6747	0.6728
Neighbor1NN	0.6548	0.6551	0.6561	0.6553	0.6554
MC	0.7161	0.7065	0.7042	0.7029	0.7027
BaseLine	0.6589	0.6567	0.6567	0.6555	0.6557
MIKNN	0.7409	0.7167	0.7076	0.7062	0.7066
Density-1 human datasets	0.01cM	0.02cM	0.03cM	0.04cM	0.05cM
fastPHASE	0.9663	0.9611	0.9584	0.9569	0.9559
NN	0.9334	0.9241	0.9149	0.9067	0.8997
WeightedNN	0.9243	0.9113	0.8997	0.8892	0.8802
SVM	0.905	0.9024	0.8962	0.8891	0.7537
NeuralNet	0.7633	0.7765	0.7805	0.7821	0.7821
Neighbor1NN	0.6392	0.6449	0.6465	0.6474	0.6479
MC	0.7718	0.7688	0.7672	0.7664	0.7658
BaseLine	0.6507	0.6503	0.6501	0.65	0.6498
MIKNN	0.9142	0.9091	0.9052	0.9023	0.9

Table A.31: Average imputation accuracies on the human datasets at three density levels with missing rate 10%, where the imputation methods were run with five corresponding genetic distance thresholds.

Density-0.01 human datasets	1cM	2cM	3cM	4cM	5cM
fastPHASE	0.6861	0.6725	0.6732	0.6732	0.6732
NN	0.7331	0.7377	0.7412	0.7442	0.7402
WeightedNN	0.7172	0.7059	0.6939	0.681	0.67
SVM	0.6585	0.6459	0.6465	0.6473	0.646
NeuralNet	0.6465	0.6462	0.6445	0.6412	0.6378
Neighbor1NN	0.6584	0.648	0.649	0.6492	0.6489
MC	0.661	0.6485	0.6491	0.6489	0.6492
BaseLine	0.6596	0.6488	0.6486	0.649	0.6492
MIKNN	0.6092	0.5832	0.5862	0.5903	0.5905
Density-0.1 human datasets	0.1cM	0.2cM	0.3cM	0.4cM	0.5cM
fastPHASE	0.7877	0.7712	0.7671	0.7661	0.766
NN	0.8121	0.7997	0.7941	0.7903	0.7884
WeightedNN	0.797	0.7752	0.7592	0.745	0.7357
SVM	0.7411	0.7286	0.7239	0.72	0.7171
NeuralNet	0.6695	0.6711	0.6718	0.6712	0.6695
Neighbor1NN	0.6555	0.6559	0.6552	0.6551	0.6554
MC	0.7113	0.7029	0.7002	0.6993	0.6993
BaseLine	0.6585	0.6569	0.6562	0.6558	0.6556
MIKNN	0.732	0.7078	0.7003	0.6988	0.6995
Density-1 human datasets	0.01cM	0.02cM	0.03cM	0.04cM	0.05cM
fastPHASE	0.9573	0.9516	0.9487	0.9471	0.946
NN	0.9269	0.9179	0.9093	0.9014	0.8945
WeightedNN	0.9131	0.9006	0.8896	0.88	0.8718
SVM	0.8952	0.8912	0.8841	0.8763	0.8106
NeuralNet	0.7571	0.769	0.7725	0.7736	0.773
Neighbor1NN	0.6417	0.6453	0.6464	0.6469	0.6472
MC	0.765	0.7622	0.7607	0.76	0.7594
BaseLine	0.6496	0.6494	0.6492	0.6491	0.6489
MIKNN	0.9048	0.9007	0.8972	0.8947	0.8925

Table A.32: Average imputation accuracies on the human datasets at three density levels with missing rate 20%, where the imputation methods were run with five corresponding genetic distance thresholds.

Density-0.01 human datasets	1cM	2cM	3cM	4cM	5cM
fastPHASE	0.6849	0.6703	0.6707	0.6707	0.6707
NN	0.7314	0.7317	0.7364	0.7374	0.7373
WeightedNN	0.7159	0.6979	0.686	0.6749	0.6639
SVM	0.6644	0.6486	0.6487	0.6472	0.6489
NeuralNet	0.6516	0.649	0.6455	0.6436	0.6417
Neighbor1NN	0.6648	0.6508	0.6514	0.6518	0.6515
MC	0.6639	0.6496	0.6501	0.6502	0.6501
BaseLine	0.665	0.6512	0.6515	0.6516	0.6518
MIKNN	0.6073	0.5821	0.5856	0.5878	0.5883
Density-0.1 human datasets	0.1cM	0.2cM	0.3cM	0.4cM	0.5cM
fastPHASE	0.7715	0.7562	0.7524	0.7514	0.7514
NN	0.7986	0.7874	0.7829	0.7782	0.7769
WeightedNN	0.7773	0.7568	0.7425	0.7295	0.7209
SVM	0.7248	0.7125	0.708	0.7044	0.7022
NeuralNet	0.6648	0.6663	0.6666	0.6649	0.6631
Neighbor1NN	0.6537	0.6539	0.6537	0.6535	0.6535
MC	0.7028	0.6948	0.6924	0.6916	0.6914
BaseLine	0.6567	0.6551	0.6545	0.6541	0.654
MIKNN	0.7129	0.6912	0.6841	0.6824	0.6836
Density-1 human datasets	0.01cM	0.02cM	0.03cM	0.04cM	0.05cM
fastPHASE	0.9458	0.9393	0.9362	0.9345	0.9334
NN	0.9145	0.9058	0.8976	0.8903	0.8837
WeightedNN	0.8944	0.8821	0.872	0.8632	0.8554
SVM	0.8738	0.8676	0.8588	0.8479	0.6602
NeuralNet	0.7442	0.7542	0.7566	0.7565	0.7555
Neighbor1NN	0.6431	0.6454	0.6462	0.6467	0.6468
MC	0.7516	0.749	0.7476	0.7469	0.7464
BaseLine	0.6492	0.649	0.6488	0.6486	0.6485
MIKNN	0.8881	0.8844	0.8813	0.8792	0.8771

Table A.33: Average imputation accuracies on the mouse datasets at three density levels with missing rate 1%, where the imputation methods were run with five corresponding genetic distance thresholds.

Density-0.01 mouse datasets	1cM	2cM	3cM	4cM	5cM
fastPHASE	0.9062	0.9076	0.9076	0.9074	0.9074
NPUTE	0.8557	0.8556	0.8562	0.8562	0.8562
NN	0.9017	0.917	0.9242	0.923	0.9263
WeightedNN	0.9081	0.9203	0.9214	0.9258	0.923
SVM	0.8301	0.8519	0.8595	0.8631	0.8654
NeuralNet	0.8231	0.8329	0.8295	0.8267	0.8235
Neighbor1NN	0.7387	0.7792	0.7916	0.8023	0.8101
MC	0.825	0.8373	0.8374	0.829	0.8309
BaseLine	0.8093	0.8111	0.8142	0.8152	0.8166
MIKNN	0.8963	0.9011	0.9025	0.9097	0.9078
Density-0.1 mouse datasets	0.1cM	0.2cM	0.3cM	0.4cM	0.5cM
fastPHASE	0.9322	0.9317	0.9311	0.931	0.931
NPUTE	0.8692	0.8684	0.8681	0.8682	0.8681
NN	0.9329	0.9403	0.9418	0.9418	0.9422
WeightedNN	0.9346	0.9405	0.9429	0.9425	0.9432
SVM	0.8653	0.8801	0.8883	0.8925	0.8945
NeuralNet	0.8367	0.8512	0.8656	0.8752	0.8859
Neighbor1NN	0.722	0.7546	0.7767	0.7895	0.7974
MC	0.8653	0.8643	0.8611	0.864	0.863
BaseLine	0.8199	0.8184	0.8185	0.8191	0.8186
MIKNN	0.9232	0.9273	0.9278	0.9279	0.9291
Density-1 mouse datasets	0.01cM	0.02cM	0.03cM	0.04cM	0.05cM
fastPHASE	0.9456	0.9452	0.9452	0.945	0.945
NPUTE	0.8821	0.8821	0.8821	0.882	0.882
NN	0.9478	0.9511	0.9519	0.9525	0.9524
WeightedNN	0.9489	0.9516	0.9519	0.9513	0.9504
SVM	0.8914	0.9036	0.9073	0.9093	0.9096
NeuralNet	0.8487	0.8543	0.8546	0.8537	0.8505
Neighbor1NN	0.7284	0.7623	0.7776	0.787	0.7929
MC	0.8761	0.8752	0.8751	0.8749	0.8753
BaseLine	0.8181	0.8182	0.8183	0.8181	0.8182
MIKNN	0.9396	0.9428	0.9433	0.9434	0.9436

Table A.34: Average imputation accuracies on the mouse datasets at three density levels with missing rate 2%, where the imputation methods were run with five corresponding genetic distance thresholds.

Density-0.01 mouse datasets	1cM	2cM	3cM	4cM	5cM
fastPHASE	0.9008	0.896	0.8962	0.8956	0.8956
NPUTE	0.8501	0.8442	0.8451	0.8447	0.8447
NN	0.9051	0.911	0.9151	0.9111	0.9083
WeightedNN	0.9038	0.9109	0.9147	0.9104	0.9041
SVM	0.833	0.8519	0.8546	0.8565	0.8574
NeuralNet	0.8208	0.8206	0.8247	0.8217	0.821
Neighbor1NN	0.7403	0.7788	0.7947	0.7949	0.7995
MC	0.8293	0.8285	0.8254	0.8306	0.824
BaseLine	0.8014	0.8013	0.8044	0.806	0.8041
MIKNN	0.8892	0.8894	0.8904	0.8866	0.888
Density-0.1 mouse datasets	0.1cM	0.2cM	0.3cM	0.4cM	0.5cM
fastPHASE	0.9282	0.9279	0.9275	0.9276	0.9276
NPUTE	0.8669	0.8659	0.8659	0.866	0.866
NN	0.9287	0.9349	0.9378	0.9383	0.9376
WeightedNN	0.93	0.936	0.9367	0.9352	0.9338
SVM	0.86	0.8775	0.8851	0.8897	0.8915
NeuralNet	0.831	0.8459	0.8598	0.8714	0.8813
Neighbor1NN	0.7396	0.7709	0.7879	0.7974	0.8017
MC	0.8587	0.8561	0.8563	0.856	0.8566
BaseLine	0.815	0.8157	0.8151	0.8154	0.816
MIKNN	0.9185	0.9226	0.9237	0.9249	0.9251
Density-1 mouse datasets	0.01cM	0.02cM	0.03cM	0.04cM	0.05cM
fastPHASE	0.9442	0.9439	0.9438	0.9438	0.9437
NPUTE	0.8818	0.8817	0.8816	0.8816	0.8815
NN	0.9471	0.9502	0.951	0.9507	0.9501
WeightedNN	0.9471	0.9487	0.9475	0.9459	0.9447
SVM	0.888	0.897	0.8961	0.8998	0.9034
NeuralNet	0.8459	0.8503	0.85	0.8474	0.8453
Neighbor1NN	0.757	0.7851	0.7959	0.801	0.8042
MC	0.8745	0.8741	0.8733	0.8732	0.8733
BaseLine	0.8178	0.8177	0.8177	0.8181	0.8178
MIKNN	0.9387	0.9414	0.942	0.942	0.9421

Table A.35: Average imputation accuracies on the mouse datasets at three density levels with missing rate 5%, where the imputation methods were run with five corresponding genetic distance thresholds.

Density-0.01 mouse datasets	1cM	2cM	3cM	4cM	5cM
fastPHASE	0.9009	0.8991	0.8995	0.8994	0.8994
NPUTE	0.8456	0.8433	0.8444	0.8446	0.8446
NN	0.8999	0.9078	0.9134	0.9111	0.9081
WeightedNN	0.8981	0.9022	0.9058	0.9008	0.8999
SVM	0.8413	0.8556	0.8608	0.8657	0.8669
NeuralNet	0.8265	0.8279	0.8296	0.8292	0.8254
Neighbor1NN	0.7874	0.8072	0.8112	0.8124	0.8157
MC	0.8396	0.8358	0.8391	0.8394	0.8379
BaseLine	0.8188	0.8181	0.8173	0.8172	0.818
MIKNN	0.8818	0.8887	0.8908	0.8945	0.895
Density-0.1 mouse datasets	0.1cM	0.2cM	0.3cM	0.4cM	0.5cM
fastPHASE	0.9275	0.9273	0.927	0.9269	0.9268
NPUTE	0.8675	0.8656	0.8656	0.8655	0.8655
NN	0.927	0.9324	0.9342	0.934	0.9345
WeightedNN	0.9261	0.9282	0.9277	0.9263	0.9251
SVM	0.8584	0.8744	0.8825	0.8863	0.8882
NeuralNet	0.828	0.8432	0.8567	0.8687	0.8776
Neighbor1NN	0.7746	0.7952	0.8037	0.8071	0.8094
MC	0.8566	0.8561	0.8557	0.8561	0.8561
BaseLine	0.8137	0.8144	0.8149	0.8149	0.8147
MIKNN	0.9144	0.9191	0.9214	0.9221	0.9227
Density-1 mouse datasets	0.01cM	0.02cM	0.03cM	0.04cM	0.05cM
fastPHASE	0.942	0.9418	0.9416	0.9416	0.9415
NPUTE	0.8791	0.879	0.8789	0.8788	0.8788
NN	0.945	0.9472	0.9472	0.947	0.9465
WeightedNN	0.9408	0.94	0.9388	0.9381	0.9376
SVM	0.7003	0.6921	0.7051	0.6804	0.6729
NeuralNet	0.8394	0.841	0.8388	0.8349	0.8311
Neighbor1NN	0.7903	0.8033	0.8076	0.8099	0.811
MC	0.8712	0.8707	0.8704	0.87	0.87
BaseLine	0.8168	0.8167	0.8167	0.8168	0.8168
MIKNN	0.9356	0.9386	0.9394	0.9398	0.9397

Table A.36: Average imputation accuracies on the mouse datasets at three density levels with missing rate 10%, where the imputation methods were run with five corresponding genetic distance thresholds.

Density-0.01 mouse datasets	1cM	2cM	3cM	4cM	5cM
fastPHASE	0.8938	0.891	0.8913	0.8913	0.8913
NPUTE	0.8409	0.8376	0.8388	0.8388	0.8388
NN	0.8937	0.899	0.9026	0.9002	0.8995
WeightedNN	0.8862	0.8879	0.8899	0.8895	0.8899
SVM	0.8364	0.8472	0.8533	0.855	0.8556
NeuralNet	0.8185	0.8195	0.8189	0.8142	0.8111
Neighbor1NN	0.7963	0.8053	0.8082	0.8097	0.8102
MC	0.8301	0.8294	0.828	0.8287	0.8283
BaseLine	0.8116	0.8103	0.8117	0.8127	0.8124
MIKNN	0.8771	0.8821	0.8842	0.8847	0.8867
Density-0.1 mouse datasets	0.1cM	0.2cM	0.3cM	0.4cM	0.5cM
fastPHASE	0.9248	0.9244	0.9243	0.9242	0.9241
NPUTE	0.8629	0.8615	0.8614	0.8614	0.8614
NN	0.923	0.927	0.9284	0.9284	0.9281
WeightedNN	0.9168	0.9172	0.9179	0.9181	0.9187
SVM	0.8581	0.8717	0.8793	0.8824	0.8835
NeuralNet	0.8271	0.8287	0.8267	0.8242	0.8203
Neighbor1NN	0.7964	0.8058	0.8099	0.8116	0.8125
MC	0.8548	0.8536	0.8536	0.8537	0.8534
BaseLine	0.8158	0.8156	0.8159	0.816	0.816
MIKNN	0.9093	0.9137	0.9168	0.918	0.9182
Density-1 mouse datasets	0.01cM	0.02cM	0.03cM	0.04cM	0.05cM
fastPHASE	0.9386	0.9383	0.9382	0.9381	0.9381
NPUTE	0.876	0.8758	0.8757	0.8757	0.8757
NN	0.9395	0.9416	0.9415	0.941	0.9405
WeightedNN	0.9311	0.9321	0.9324	0.9325	0.9325
SVM	0.8814	0.8912	0.8739	0.8924	0.8798
NeuralNet	0.8326	0.8302	0.8253	0.8199	0.8147
Neighbor1NN	0.8043	0.8098	0.8116	0.8126	0.813
MC	0.8665	0.8659	0.8652	0.8651	0.865
BaseLine	0.8167	0.8166	0.8166	0.8168	0.8168
MIKNN	0.9303	0.9342	0.9353	0.9357	0.936

Table A.37: Average imputation accuracies on the mouse datasets at three density levels with missing rate 20%, where the imputation methods were run with five corresponding genetic distance thresholds.

Density-0.01 mouse datasets	1cM	2cM	3cM	4cM	5cM
fastPHASE	0.8806	0.8799	0.8803	0.8804	0.8804
NPUTE	0.8284	0.8272	0.8282	0.8284	0.8284
NN	0.8742	0.8812	0.8846	0.8835	0.8806
WeightedNN	0.8612	0.8643	0.8696	0.8714	0.8723
SVM	0.8292	0.8377	0.8441	0.8448	0.8435
NeuralNet	0.817	0.8141	0.8099	0.8046	0.7992
Neighbor1NN	0.805	0.8096	0.8118	0.8122	0.8128
MC	0.8258	0.8267	0.8278	0.827	0.8264
BaseLine	0.8129	0.814	0.8146	0.8144	0.8157
MIKNN	0.8551	0.863	0.8702	0.8727	0.872
Density-0.1 mouse datasets	0.1cM	0.2cM	0.3cM	0.4cM	0.5cM
fastPHASE	0.9132	0.9126	0.9123	0.9123	0.9122
NPUTE	0.8544	0.8529	0.8528	0.8528	0.8528
NN	0.9061	0.9109	0.9126	0.913	0.9134
WeightedNN	0.8943	0.8987	0.9015	0.9042	0.9056
SVM	0.8479	0.8576	0.8627	0.8648	0.8657
NeuralNet	0.8193	0.8168	0.8125	0.8068	0.8016
Neighbor1NN	0.8029	0.807	0.8091	0.8101	0.811
MC	0.844	0.8426	0.8424	0.8426	0.8429
BaseLine	0.813	0.8131	0.8135	0.8136	0.8136
MIKNN	0.8937	0.9005	0.9043	0.9059	0.9073
Density-1 mouse datasets	0.01cM	0.02cM	0.03cM	0.04cM	0.05cM
fastPHASE	0.9308	0.9305	0.9305	0.9304	0.9303
NPUTE	0.8698	0.8695	0.8695	0.8695	0.8694
NN	0.9271	0.9298	0.9303	0.93	0.9298
WeightedNN	0.9165	0.9208	0.9225	0.9232	0.9237
SVM	0.7311	0.8119	0.7415	0.7587	0.7807
NeuralNet	0.8207	0.8134	0.8057	0.7986	0.7916
Neighbor1NN	0.8073	0.81	0.8111	0.8118	0.8121
MC	0.8561	0.8557	0.8552	0.8549	0.8548
BaseLine	0.8148	0.8149	0.8149	0.815	0.815
MIKNN	0.9195	0.9248	0.9266	0.9274	0.9279