This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2708701, IEEE/ACM Transactions on Computational Biology and Bioinformatics

# Highly Accurate and Efficient Data-Driven Methods For Genotype Imputation

Olivia Choudhury, *Student Member, IEEE*, Ankush Chakrabarty, *Member, IEEE*, and Scott J. Emrich, *Senior Member, IEEE*

**Abstract**—High-throughput sequencing techniques have generated massive quantities of genotype data. Haplotype phasing has proven to be a useful and effective method for analyzing these data. However, the quality of phasing is undermined by the presence of missing information. Imputation provides an effective means of improving the underlying genotype information. For model organisms, imputation can rely on an available reference genotype panel and a physical or genetic map. For non-model organisms, which often do not have a genotype panel, it is important to design an imputation technique that does not rely on reference data. Here, we present ADDIT (Accurate Data-Driven Imputation Technique), which is composed of two data-driven algorithms capable of handling data generated from model and non-model organisms. The non-model variant of ADDIT (referred to as ADDIT-NM) employs statistical inference methods to impute missing genotypes, whereas the model variant (referred to as ADDIT-M) leverages a supervised learning-based approach for imputation. We demonstrate that both variants of ADDIT are more accurate, faster, and require less memory than leading state-of-the-art imputation tools using model (human) and non-model (maize, apple, grape) genotype data.

**Software Availability:** The source code of ADDIT and test data sets are available at https://github.com/NDBL/ADDIT

**Index Terms**—Genotype imputation, single nucleotide polymorphisms (SNPs), next-generation and high-throughput sequencing, machine learning, big data

## 1 Introduction

High-throughput techniques like whole genome sequencing, whole exome sequencing, and genome-wide single nucleotide polymorphism (SNP) microarrays are generating huge volumes of genotype data. To associate phenotypes such as disease susceptibility with underlying genotypes, there has been a rapid growth of phasing-based inference formalisms. Phasing is particularly useful in genome-wide association studies (GWAS) [1] to infer linked alleles on a chromosome. Other downstream analyses include identifying recombinant breakpoints [2], deducing history of human demographics [3], and modeling cis-regulation of gene expression [4].

Missing genotype data is a major hindrance to phasing. This is a result of inherent shortcomings of the underlying techniques that generate such data. Previous efforts have shown that genotype imputation can improve phasing quality in genetic association studies by up to 10% [5]. Although we focus on imputation as a precursor to phasing, secondary benefits of formulating data-driven imputation methods include generating higher fidelity genetic maps and improved metagenomic analysis (see for example, [6], [7]).

A majority of the existing imputation methods require a panel of reference genotypes and a physical or genetic map. The absence of such a reference panel, as in *non-model* organisms, makes the problem of imputation and phasing much more difficult using currently available software. We address this gap with a lightweight framework, referred to as ADDIT-NM, for fast and accurate imputation in non-model organisms that relies only on the underlying statistics of the genotype data. A preliminary version of ADDIT-NM has been reported in [8]. We also demonstrate that the model organism specific variant of ADDIT, referred to as ADDIT-M, can extract available information in the reference panels via supervised learning to significantly improve imputation accuracy. We perform an extensive, comparative numerical study of ADDIT against the leading imputation tools, such as Beagle [9], IMPUTE2 [10] (for model and non-model organisms) and LinkImpute [11] (for non-model organisms). The comparison results are compiled using real data of varying sizes, varying proportions of missing genotypes, and varying sizes of reference panels (for model organisms). In these comparisons, ADDIT consistently outperforms the other tools in terms of speed, memory, and accuracy.

Our primary **contributions** include:

(i) the formulation of data-driven, lightweight imputation algorithms for both model and non-model organisms with high speed and accuracy;
(ii) the incorporation of both local and global information by utilizing adaptive windows and trust metrics;
(iii) the exploitation of adjacent genotype data to significantly expedite imputation under certain conditions;
(iv) the employment of multi-class supervised learning algorithms to extricate information from reference panels of model organisms to enhance the imputation process.

- *O. Choudhury and S.J. Emrich are with the Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN, 46556.*
*E-mail: ochoudhu@nd.edu, semrich@nd.edu*
- *A. Chakrabarty is with the Harvard John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, 02138.*
*E-mail: achakrabarty@seas.harvard.edu*

*Manuscript received December 2, 2016; revised April, 2017.*

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2708701, IEEE/ACM Transactions on Computational Biology and Bioinformatics

The rest of the paper is organized as follows. In Section 2, we present the relevant literature and explain current limitations. Our proposed method is discussed in detail in Section 3, with subsections devoted to the implementation of ADDIT-NM and ADDIT-M for non-model and model organisms, respectively. Results from a comparative study with state-of-the-art imputation algorithms are reported in Section 4 using real data from multiple non-model organisms and a model organism (human). We conclude and discuss future work in Section 5.

## 2 Related Work

Prior work on genotype imputation generally considered either related or unrelated samples. In individuals containing blocks of shared haplotypes, the authors in [2] proposed a method called identity-by-descent (IBD) to impute missing values. For closely related populations of small size with large number of samples, methods such as [12], [13] have demonstrated effectiveness via long-range phasing.

For unrelated samples with strongly linked polymorphisms, Clark's algorithm [14] is one of the earliest approaches for inferring haplotypes. To relax the inherent assumption of tight linkage, the principle of Expectation Maximization (EM) is leveraged in [15]. However, EM-based methods are effective only when high quality prior models are available for training. They are also computationally prohibitive for large-scale genotype data. More recent methods take into consideration the fact that new haplotypes are derived from older haplotypes via mutation and recombination [16].

Based on these observations, widely used phasing tools, such as IMPUTE2 [17], generate approximate coalescent models and hidden Markov models (HMMs) from genotypes for subsequent stochastic EM-based algorithms. PHASE [18] employs Markov chain Monte-Carlo (MCMC) algorithm to explore possible combinations of haplotypes [19]. The combinatorial explosion inherent in MCMC limits the applicability of this tool to small datasets [20]. FastPHASE [21], a faster variant of PHASE, implements a parsimonious clustering of haplotypes and is more amenable to smaller sample sizes. In large datasets, the algorithm uses subsets of haplotypes, resulting in performance degradation. To overcome this challenge in large samples, Beagle [22] employs haplotype clustering at individual loci to compute transition probabilities in HMM.

For model organisms with reference panels, existing tools ( [17], [23], [24]) extricate information from the reference panel to generate an HMM model for phasing and imputation. An imputation algorithm based on a variant of $k$-nearest neighbor interpolation employed in LinkImpute [11] has been demonstrated to perform well in a myriad of heterozygous populations; however, it is usually sluggish when the dataset size is increased, and exhibits limited accuracy.

Machine learning methods have been used in genetic analysis [25] and the genotype imputation problem has been explored using artificial neural networks [26]. Although the authors in [27] report an extensive comparison of modern bi-classification methods to genotype imputation, for general populations, more than two alleles is more viable for imputation. For more general multi-class classification solutions, we refer the reader to [28]–[30]. The authors in [31] demonstrate the effectiveness of reduced-feature models in comparison with the two common missing value treatments: missing data handling via oracle/discarding, and missing data handling via imputation, on a suite of benchmark data sets. We design ADDIT-M based on reduced-feature models as it restricts the selection of training samples to a small neighborhood of the value to be imputed, thereby preserving local distribution properties.

## 3 Methods

In this section we present a detailed description of our proposed ADDIT algorithm. For ADDIT-NM, the imputation framework is segregated into multiple steps (Steps 1–5), and justification for each step is provided in section 3.1. Note that the algorithm in Section 3.1 is described in detail in [8]. For ADDIT-M, we present a windowed, multi-class supervised learning-based imputation algorithm in section 3.2.

Let $N$ be the number of samples in a given population and $M_0$ be the total number of missing genotypes in the entire population. Let $G_q^j$ denote the genotype at the $j$th position of the $q$th sample. Here, $q \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, M\}$. Let $\mathcal{I}_q$ be the set of potential imputable genotypes at $G_q^j$. We denote a window centered at $G_q^j$ by $\mathcal{W}(G_q^j, d)$, where $d$ is the window length and the window contains $(d-1)/2$ elements on either side of the central element $G_q^j$. We use Hamming distance to measure the similarity between two windows of identical length. For example, the Hamming distance between '111000' and '100010' is three. Let $|A|$ represent the cardinality of a set $A$. A list of important notation/symbols used in the subsequent discussion can be found in Table 1.

TABLE 1
**List of symbols used in the ADDIT-NM algorithm description.**

| *Symbol* | *Meaning* |
|---|---|
| # | number of |
| $N$ | # samples in given population |
| $M_0$ | total # missing genotypes |
| $M$ | # missing genotypes with distinct neighbors |
| $q$ | query sample |
| $G_k^j$ | genotype at $j$th position of $k$th sample |
| $\mathcal{W}(c, d)$ | window of length $d$ centered at $c$ |
| $\mathcal{C}_q$ | set of candidate windows for query window $q$ |
| $\{\mathcal{C}_q\}_{q=1}^M$ | family of $\mathcal{C}_q$ for all $M$ query windows |
| $\rho_H$ | Hamming distance |
| $\theta$ | similarity score |
| $\theta_{\min}$ | similarity threshold |
| $\mathcal{T}_q$ | set of trusted candidates for query window $q$ |
| $\{\mathcal{T}_q\}_{q=1}^M$ | family of $\mathcal{T}_q$ for all $M$ query windows |
| $\omega_f$ | decision weight based on allele frequency |
| $\omega_s$ | decision weight based on window similarity |
| $\mathcal{I}_q$ | set of imputable genotypes for query sample $q$ |
| $i_q$ | element of $\mathcal{I}_q$ |
| $\mathcal{P}(i_q)$ | priority level of $i_q$ according to Table 2 |

### 3.1 ADDIT-NM: Imputation for non model organisms

For non-model organisms, ADDIT-NM uses adaptive windows and likelihoods to estimate missing values. An overall schematic of ADDIT-NM is provided in Figure 1.

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

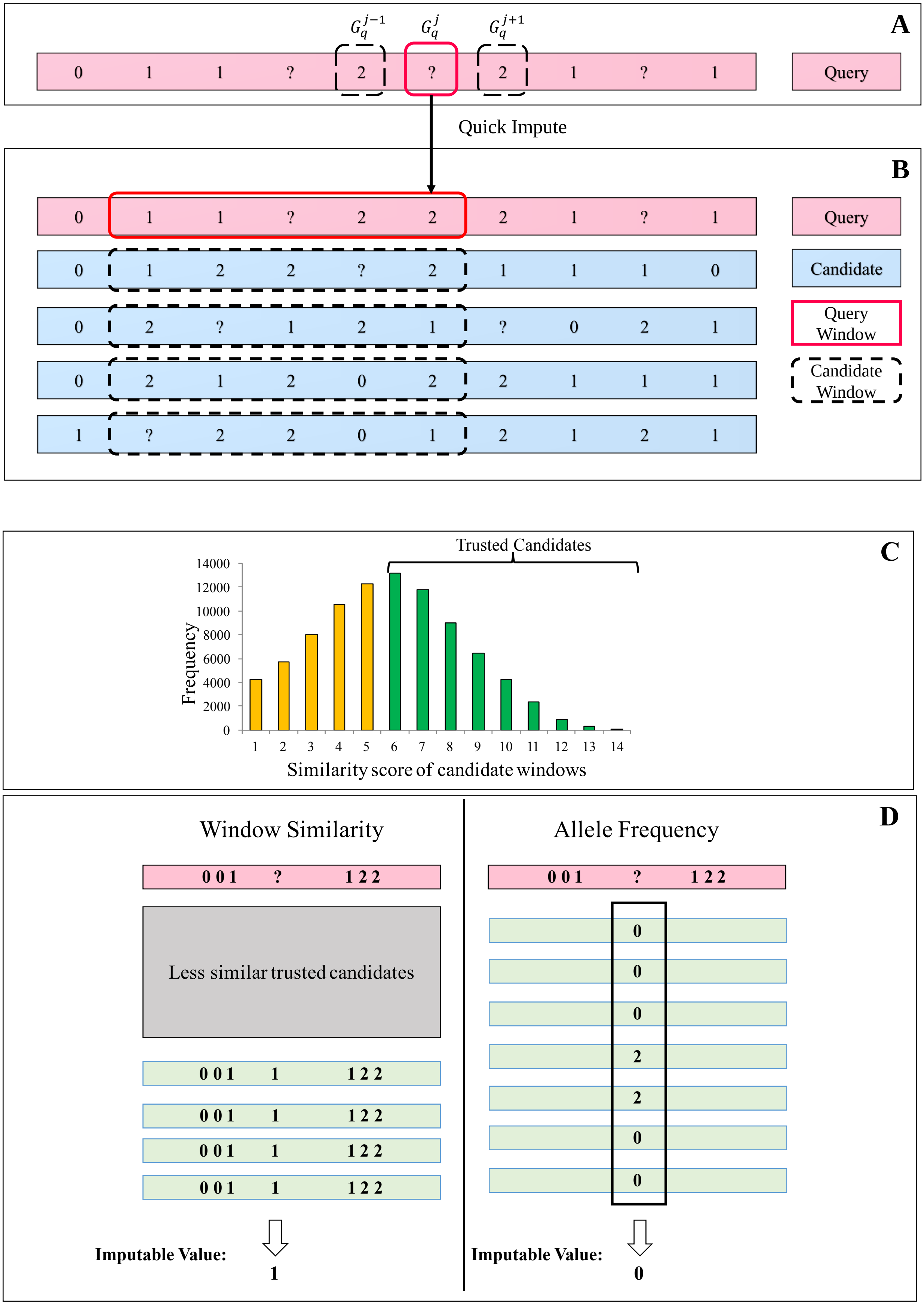

Fig. 1. ADDIT-NM for non-model organisms: (A) quick impute (QI) step. (B) Selection of candidate windows with $d = 5$. (C) Selection of trusted candidates using maximum likelihood. (D) Illustration of priority impute via window similarity (left) and allele frequency (right).

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2708701, IEEE/ACM Transactions on Computational Biology and Bioinformatics

### *Step 1: Quick imputation using immediate neighbors*

Suppose $G_q^j$ is a missing genotype. We begin by performing a quick impute (QI) step by imputing the central value to either of its two neighboring alleles if $G_q^{j-1} = G_q^{j+1}$; this is described in Algorithm 1 and illustrated in Figure 1A. The QI step is effective in substantially reducing the search space during imputation, if the likelihood of double recombination in adjacent alleles is low.

**Algorithm 1** Quick imputation using immediate neighbors

```
Require: Genotype data with $M_0$ missing values
Require: Window length, $d$
for each missing $G_q^j$ do
    if $G_q^{j-1} = G_q^{j+1}$ then
        Quick impute $G_q^j \leftarrow G_q^{j+1}$
    end if
end for
```

### *Step 2: Similarity computation for each missing genotype*

For a missing value $G_q^j$ in the query sample, we create a **query window** of length $d$, denoted $\mathcal{W}(G_q^j, d)$. We construct windows of identical length centered at the $j$th position of each of the remaining population samples, which we call **sample windows**, denoted $\mathcal{W}(G_k^j, d)$, where $k \in \{1, \dots, q-1, q+1, \dots, N\}$.

**Algorithm 2** Similarity computation for each missing genotype

```
Require: $M$ missing genotypes with distinct neighbors
for each missing $G_q^j$ with distinct neighbors do
    Construct query window $\mathcal{W}(G_q^j, d)$
    for each $k \in \{1, \dots, N\}$ except $q$ do
        if $G_k^j$ is not missing data then
            $\mathcal{C}_q \leftarrow$ add $\mathcal{W}(G_k^j, d)$ to the set of candidate windows
        end if
    end for
    for $i =$ each candidate window in $\mathcal{C}_q$ do
        $\theta_i \leftarrow$ similarity score using (1)
    end for
end for
return set of candidate windows for $M$ missing genotypes, $\{\mathcal{C}_q\}_{q=1}^M$
return set of similarity scores for all candidate windows, $\{\theta_q\}_{q=1}^M$
```

We now exclude those sample windows that have missing data at $G_k^j$. The subset of sample windows with no missing data at $G_k^j$ is hereby referred to as the **candidate set** $\mathcal{C}_q$ for the missing genotype $G_q^j$ (Figure 1B). Each sample window is given a **similarity score** using

$$\theta_i = d - \rho_H \left( \mathcal{W}(G_q^j, d), \mathcal{W}(G_i^j, d) \right) \tag{1}$$

where $\rho_H$ is the Hamming distance between the query window and the $i$th sample window. Iterating over the remaining set of missing genotypes after the QI step, we acquire a set of candidate windows, $\{\mathcal{C}_q\}_{q=1}^M$ with a corresponding set of similarity scores $\{\theta_q\}_{q=1}^M$.

### *Step 3: Similarity threshold of candidate windows*

Consider a histogram of similarity scores for the collection of candidate windows denoted as $\mathcal{H}(\theta)$. A maximum likelihood

$$\theta_{\min} = \arg\max_{\theta} \mathcal{H}(\theta) \tag{2}$$

is used to compute a **similarity threshold**, the procedure for which is provided in Algorithm 3.

**Algorithm 3** Similarity threshold of candidate windows

```
Require: $\{\mathcal{C}_q\}_{q=1}^M, \{\theta_q\}_{q=1}^M$ from Algorithm 2
for $r = 1$ to $d-1$ do
    $\mathcal{H}(\theta) \leftarrow$ frequency of windows in all candidate sets $\{\mathcal{C}_q\}_{q=1}^M$ with similarity score $r$
end for
$\theta_{\min} \leftarrow \arg\max_\theta \mathcal{H}(\theta)$
return Similarity threshold, $\theta_{\min}$
```

It is important to note that the value of $\theta_{\min}$ is computed considering all candidate windows of the $M$ missing genotypes remaining after Step 1. The inherent globality in our formulation offers various advantages: First, it avoids bias induced by local candidate windows with low similarity scores. For example, suppose that local candidate windows for a particular missing genotype have similarity scores between 2 and 7, with window length $d = 11$. Also consider that the entire sample population contains windows (with identical $d$) of similarity scores between 2 and 10 with most windows having scores $> 6$. If we compute $\theta_{\min}$ based on local candidates only, then $\theta_{\min}^{\text{local}}$ is (say) 4. However, using the globally-derived similarity threshold $\theta_{\min}$, we obtain $\theta_{\min}^{\text{global}} = 6$ because the global sample population has more candidate windows with higher similarity scores. If we were to impute a value based on $\theta_{\min}^{\text{local}}$, then we would enable windows with lower similarity scores to have an effect on the decision, which is undesirable, as it is likely to recommend an erroneous imputed value. Instead, using $\theta_{\min}^{\text{global}}$ filters out these low-similarity candidates, resulting in more accurate imputation. Finally, our method avoids diluting information; using candidate windows retains relevant local information with embedded global trends of these data. This improves imputation accuracy while lowering required computation.

We will next leverage the notion of the similarity threshold to categorize candidate windows as trusted or untrusted.

### *Step 4: Adaptive classification of trusted candidates*

For the $q$th query window we construct a set of **trusted candidates**, denoted $\mathcal{T}_q$. A candidate window is said to be a trusted candidate if its similarity score is at least the similarity threshold $\theta_{\min}$ (as shown in Figure 1C). Note that $\mathcal{T}_q$ cannot be empty for the choice of $\theta_{\min}$ in (2). This claim can be proven by contradiction, given that there is at least one candidate window for a given window length $d > 2$. Suppose $\mathcal{T}_q$ is empty for a given $\theta_{\min} := \arg\max_\theta \mathcal{H}(\theta)$. This implies that there is no candidate window whose similarity score is at least $\theta_{\min}$. Clearly, this is a contradiction because the set of candidate windows is not empty, so at least one candidate window must have similarity score $\theta_{\min}$

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2708701, IEEE/ACM Transactions on Computational Biology and Bioinformatics

in order to ensure that it is the maximizer of the histogram $\mathcal{H}(\theta)$. Therefore, $\mathcal{T}_q$ must be non-empty for this choice of $\theta_{\min}$.

We refer to this method as *adaptive* because it allows each $\mathcal{T}_q$ to have a variable number of trusted candidate windows, unlike existing frameworks such as those employing $k$-nearest neighbor algorithms. This is advantageous because it exploits only the most similar windows for subsequent imputation. To take into account both window similarity and repetitiveness of the central allele, we introduce the following priority-based weighting scheme.

**Algorithm 4** Adaptive classification of trusted candidates

```
Require: Set of candidates $\{\mathcal{C}_q\}_{q=1}^{M}$ with similarity values
    $\{\theta_q\}_{q=1}^{M}$, obtained in Algorithm 2; similarity threshold,
    $\theta_{\min}$
for each missing genotype $G_q^j$ do
    $\mathcal{C}_q \leftarrow$ set of candidate windows for $G_q^j$
    for each candidate window in $\mathcal{C}_q$ do
        $\theta \leftarrow$ similarity score of candidate window
        if $\theta \geq \theta_{\min}$ then
            $\mathcal{T}_q \leftarrow$ add candidate window to the set of
    trusted candidates
        end if
    end for
end for
return All $M$ sets of trusted candidates, $\{\mathcal{T}_q\}_{q=1}^{M}$
```

### Step 5: Priority-based Imputation scheme

Recall that $\mathcal{I}_q$ is the set of potential imputable genotypes at $G_q^j$. For each imputable genotype $i_q \in \mathcal{I}_q$, we assign **decision weights** based on two criteria: (i) the frequency of $i_q$ at the central element over all trusted candidate windows in $\mathcal{T}_q$; and (ii) the similarity score of trusted candidate windows containing $i_q$ in the central position (refer to Figure 1D). The window similarity decision weight $\omega_s$ indicates the reliability of $T_q^{i_q}$ for imputing the missing genotype with $i_q$. The allele frequency decision weight $\omega_f$ signifies the likelihood of $i_q$, even if the corresponding trusted candidates have low similarity scores with respect to the query window. The decision weights are designed to handle potential bias towards highly frequent genotypes found in trusted candidates with low similarity scores. Mathematically, the decision weights are written as:

$$\omega_f(i_q) = \frac{F_{i_q}}{|\mathcal{T}_q|}, \tag{3a}$$

$$\omega_s(i_q) = \frac{1}{F_{i_q}} \sum_{k \in \mathcal{T}_q^{i_q}} \frac{d - \rho_H\left(\mathcal{W}(G_q^j, d), \mathcal{W}(G_k^j, d)\right)}{d-1}, \tag{3b}$$

where $F_{i_q}$ is the frequency of $i_q$ at the central position of the trusted candidates, $\mathcal{T}_q^{i_q} \subset \mathcal{T}_q$ is the set of trusted candidates with $i_q$ in the central position, and $\rho_H\left(\mathcal{W}(G_q^j, d), \mathcal{W}(G_k^j, d)\right)$ is the Hamming distance between the query window and each window $\mathcal{W}(G_k^j, d) \in \mathcal{T}_q^{i_q}$.

We categorize the values of $\omega_f(i_q)$ and $\omega_s(i_q)$ as high, medium, or low. For this categorization, we use data in the histogram obtained in Step 3 as follows. We first eliminate all windows with similarity scores below $\theta_{\min}$ as in Step 3. Thus, the lowest allowable similarity score is $\theta_{\min}$, which motivates us to classify $< \theta_{\min}/d$ as low. Let $\theta_{\max}$ be the highest similarity score on the histogram with non-zero frequency. Then we classify high as $> \theta_{\max}/d$. Note that $\theta_{\max}/d < 1$ since the maximal similarity is $d-1$. All decision weights in the range $[\theta_{\min}/d, \theta_{\max}/d]$ are considered medium. For each imputable genotype $i_q$, we determine its priority level $\mathcal{P}(i_q)$ using the rules in Table 2.

TABLE 2
**Priority level ($\mathcal{P}$) for possible combinations of decision weights ($\omega_s$ and $\omega_f$) during Step 5 of ADDIT-NM**

| $\omega_s$ | $\omega_f$ | Priority based on | $\mathcal{P}$ |
|---|---|---|---|
| High | - | Window Similarity | 1 |
| Med | High | Allele Frequency | 2 |
| Med | Med/Low | Window Similarity | 3 |
| Low | High/Med | Allele Frequency | 4 |
| Low | Low | - | 5 |

Within Table 2, the priorities are set such that higher weights are given to imputable genotypes supported by highly similar trusted candidates irrespective of $\omega_f$. This is motivated by the fact that, in haplotypes, we expect highly similar samples over local regions to exhibit identical inheritance of genotypes. If the trusted candidates have medium similarity, then we check $\omega_f$. This is because the trusted candidates cannot be completely relied upon to generate a correct imputed genotype. Instead, we also rely on a likelihood-based estimate embedded into $\omega_f$. For an imputable genotype with medium $\omega_s$, if its corresponding $\omega_f$ is high, then that holds more priority than medium or low $\omega_f$. We will assign even less priority to the genotypes that have high or medium frequencies with low $\omega_s$ for similar reasons as discussed above. Finally, we will assign the least priority to a genotype if its supporting trusted candidates are of low similarity and low frequency. For such cases, we investigate the remaining genotypes in $\mathcal{I}_q$.

We impute the genotype $i_q$ with the highest priority level. If there is a clash of priorities, either value can be imputed. This is written mathematically as:

$$\hat{i}_q = \arg\min_{i_q \in \mathcal{I}_q} \mathcal{P}(i_q), \tag{4}$$

where $\hat{i}_q$ is the imputed genotype. The pseudo-code for this step is presented in Algorithm 5.

## 3.2 ADDIT-M: Imputation for model organisms

### Step 1: Construction of training and truth sets from reference panel

Let $N_{\text{train}}$ be the number of training samples collected to train a supervised learning machine $\mathcal{L}$, and $R_k^j$ denote the $j$th position of the $k$th reference sample. Recall that $G_q^j$ is the missing value in the query sample at the $j$th position, and $d$ is a positive integer that denotes the number of features of the training set.

For each $G_q^j$, we begin by constructing a truth set, $\mathcal{S}_{\text{truth}}^j \in \mathbb{R}^{N_{\text{train}}}$, and training set, $\mathcal{S}_{\text{train}}^j \in \mathbb{R}^{N_{\text{train}} \times (d-1)}$: these will be used by a classifier $\mathcal{L}$ to inform the imputation process. The truth set $\mathcal{S}_{\text{truth}}^j$ is constructed using $N_{\text{train}}$

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

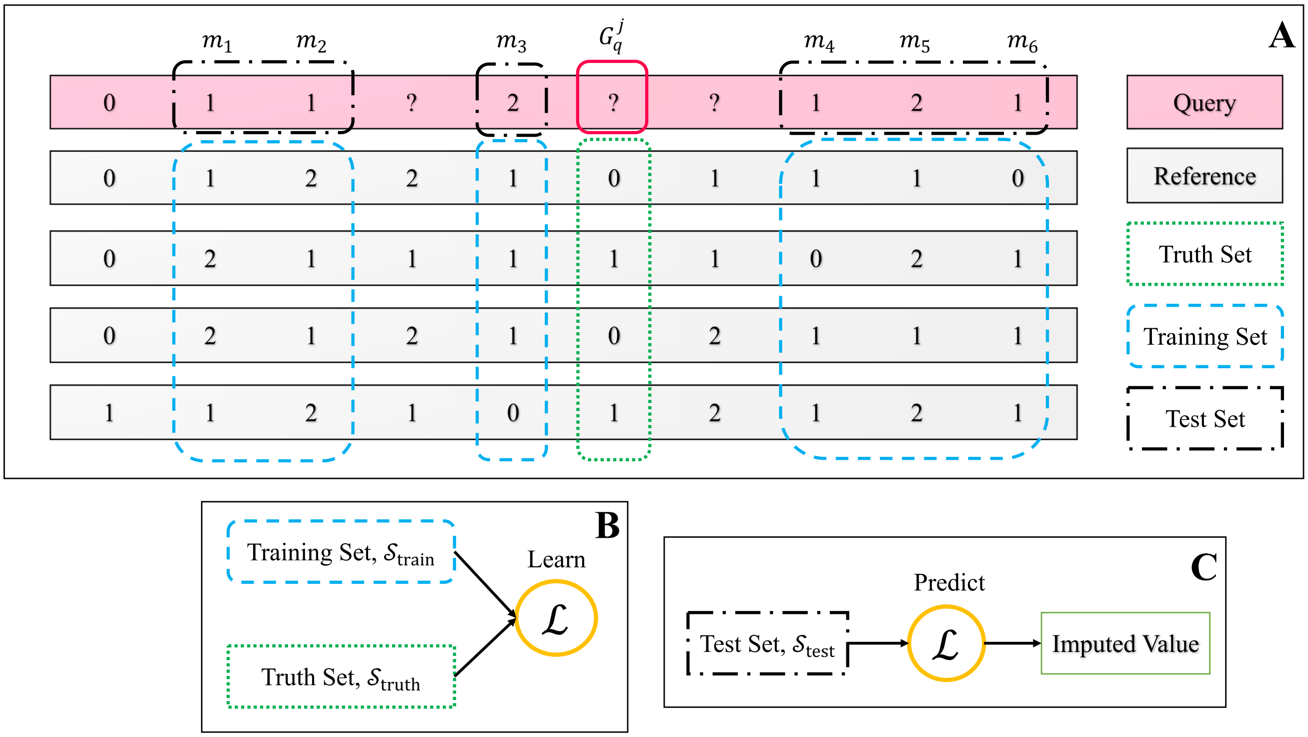

Fig. 2. ADDIT-M for model organisms: **A.** Construction of truth (green dotted rectangle), training (blue dashed rectangle) and testing (black dashed-dot rectangle) sets from query sample (top pink block) and reference panel (gray blocks) for model organisms, assuming $d = 7$. **B.** Training procedure for supervised learning algorithm $\mathcal{L}$. **C.** Imputation procedure using trained classifier $\mathcal{L}$.

**Algorithm 5** Priority-based imputation scheme

```
Require: Set of trusted candidates $\mathcal{T}_q$ for query window $\mathcal{W}(G_q^j, d)$
Require: Set of imputable genotypes $\mathcal{I}_q$
for each imputable genotype $i_q \in \mathcal{I}_q$ do
    $\mathcal{T}_q^{i_q}$ ← trusted candidates with $i_q$ in the central position
    $\omega_f(i_q)$, $\omega_s(i_q)$ ← decision weights using eqn. (3)
    Categorize $\omega_f$ and $\omega_s$ as ‘high’, ‘medium’, or ‘low’
    $\mathcal{P}(i_q)$ ← priority level according to Table 2
end for
$\hat{i}_q$ ← using eqn. (4)
return Imputation decision, $\hat{i}_q$
```

genotypes from the reference panel that do not have missing data at the $j$th position, that is:

$$\mathcal{S}^j_{\text{truth}} = \left[R^j_1 \quad R^j_2 \quad \cdots \quad R^j_{N_{\text{train}}}\right].$$

The training set construction (feature selection) is more involved. One cannot select a training set containing reference data with indices belonging to a window of length $d$ centered at $R^j_k$ for $k = 1, 2, \ldots, N_{\text{train}}$, because the corresponding testing set (a window of length $d$ centered at $G^j_q$) could contain missing data, which may result in low-quality predictions. Instead the training set is selected from the reference panels corresponding to indices in the neighborhood of $G^j_q$ that do not contain missing values. This can be written more rigorously as

$$\mathcal{S}^j_{\text{train}} = \begin{bmatrix} R_1^{m_1} & R_1^{m_2} & \cdots & R_1^{m_{d-1}} \\ R_2^{m_1} & R_2^{m_2} & \cdots & R_2^{m_{d-1}} \\ \vdots & \vdots & \ddots & \vdots \\ R_{N_{\text{train}}}^{m_1} & R_{N_{\text{train}}}^{m_2} & \cdots & R_{N_{\text{train}}}^{m_{d-1}} \end{bmatrix},$$

where $\{m_i\}_{i=1}^{d-1}$ is a set of indices representing $d-1$ nearest neighbors to $G^j_q$ containing no missing values. The corresponding test set is given by

$$\mathcal{S}^j_{\text{test}} = \left[G_q^{m_1} \quad G_q^{m_2} \quad \cdots \quad G_q^{m_{d-1}}\right].$$

The formation of the training, testing, and truth set is illustrated in Figure 2A.

### *Step 2: Imputation based on identical truth values*

Clearly, if all the labels in the truth set $\mathcal{S}^j_{\text{truth}}$ are identical, there is no need to train the classifier $\mathcal{L}$. In such a case, the imputed value is the label in $\mathcal{S}^j_{\text{truth}}$.

### *Step 3: Quick imputation*

This is an optional step, which performs effectively for data exhibiting a low degree of double recombination in adjacent positions. The implementation of this step has been previously discussed in Step 1 of Section 3.1.

### *Step 4: Imputation via multi-class supervised learning*

If the conditions in the earlier steps are not satisfied, this implies that the truth set $\mathcal{S}^j_{\text{truth}}$ contains more than one unique label. In fact, $\mathcal{S}^j_{\text{truth}}$ could contain multiple labels;

for example, three labels if the genotypes are encoded with $\{0, 1, 2\}$. The learning machine $\mathcal{L}$ is, therefore, referred to as a multi-class learning machine [32]. The multi-class classifier $\mathcal{L}$ learns from the training set $\mathcal{S}^j_{\text{train}}$ and the corresponding multi-class truth values $\mathcal{S}^j_{\text{truth}}$, and can consequently be used to predict the value of $G^j_q$ using the test set $\mathcal{S}^j_{\text{test}}$. This procedure is illustrated in Figure 2B–C.

## 4 Results and Discussion

### 4.1 Testing ADDIT-NM

#### 4.1.1 Data Acquisition

We test ADDIT-NM on three benchmark plant datasets considered in [11] (see Table 3 for details). In summary, we use genotype by sequencing (GBS) data from members of the grape genus *Vitis* generated by Illumina Hi-Seq and mapped to its reference genome [33], [34]. Some SNPs based on missing values, heterozygosity, and minor allele frequency (MAF) as in [11] are discarded. A similar apple dataset generated from members of the genus *Malus* is acquired from the 1995 accession from the US Department of Agriculture repository in Geneva, NY. The samples are double-digested with restriction enzymes and sequenced with Illumina Hi-Seq. The reads are mapped to the reference genome of *Malus domestica* version 1.0 [35]. Similar to the above grape data, variants are also filtered. Finally, we consider a large maize (corn) dataset available at the International Maize and Wheat Improvement Center [36] to verify the scalability of our proposed algorithm. For this dataset, a pre-processing stage eliminates bi-allelic SNPs with $< 20\%$ missing data, minor allele frequencies (MAF) of $> 1\%$, and samples with $> 20\%$ missing values.

#### 4.1.2 Comparative Analysis

We implement ADDIT-NM and compare its performance with contemporary imputation algorithms such as Beagle 3.3.2, LinkImpute, and IMPUTE2. Performance metrics used for this comparison include (i) percentage of genotype imputation errors; (ii) runtime; and (iii) memory usage. The results of this comparative study are tabulated in Table 3. It is clear that ADDIT-NM significantly outperforms the competition. For example, the genotype errors of grape and maize imputation are less than half the minimum of the errors produced by the other methods. The runtime of ADDIT-NM is consistently small, at times an order of magnitude smaller than the corresponding runtimes of Beagle and/or LinkImpute. Importantly, this large speed-up does not result in prohibitive use of memory. This is demonstrated by a 2–3 order-of-magnitude reduction of memory usage in comparison with Beagle or LinkImpute, and significantly less memory (around half or less) as IMPUTE2.

As discussed in [20], Beagle is more accurate than IMPUTE2 for large sample sizes. IMPUTE2 implements prephasing, wherein genotypes are first phased and then haplotypes are imputed. This reduces runtime and memory usage at the cost of accuracy. LinkImpute requires similar runtime as Beagle, although it has slightly higher accuracy. It incurs high computational overhead since it uses a genome-wide similarity search based on $k$-nearest neighbor imputation (kNNi) [37]. Contrary to these, ADDIT-NM relies on an adaptive number of reliable trusted candidate windows, which helps in increasing imputation accuracy. It also can significantly reduce runtime and memory use via an initial pruning of the search space that we call quick imputation. Unlike Beagle and IMPUTE2, we do not require a large genotype panel, which further reduces the lookup time and memory required and makes ADDIT-NM applicable to less studied organisms.

#### 4.1.3 Effectiveness of Quick Imputation

A major reason for the computational efficiency of ADDIT-NM is due to the quick imputation step. To illustrate the performance of each imputation stage (that is, quick versus priority-based imputation), we refer the reader to Figure 3. We observe that the number of quick imputes (QI) is significant for each dataset. However, the corresponding number of quick impute errors (QI Error) are small. For the grape, apple, and maize datasets, 1 out of 1487 ($< 0.1\%$), 210 out of 6326 ($< 4\%$), and 168 out of 6078 ($< 3\%$) genotypes, respectively, are incorrectly imputed in the QI stage. Figure 3 also contains information regarding the number of priority imputations (PIs) and their corresponding imputation errors (PI Error). For the real datasets, the proportion of PI Errors is low, ranging from $< 10\%$ in apple and grape, to $< 17\%$ in maize. This trend suggests that for these plant data, the QI step can be exploited because it combines high computational speeds along with a relatively lower imputation error rate.

This is further supported by comparing imputation performance of ADDIT-NM with and without the QI step (Table 4). We observe that the maximum memory used for both the configurations are identical for all datasets, and the error percentage is comparable; a minuscule increase in the number of errors is noted when the quick impute step is skipped. The most noteworthy result obtained from this investigation is the runtime differences: the lack of the quick impute step results in higher execution time for each dataset, particularly for larger datasets, as expected.

### 4.2 Testing ADDIT-M

We also test the performance of ADDIT-M on human model organism data. We use the multi-class support vector machine (MC-SVM) as an exemplar supervised learning algorithm. The MC-SVM is implemented via Python's `scikit-learn` module. The rationale behind choosing the SVM as our supervised learning method is its ability to handle high-dimensional data using the kernel-trick, its efficiency with smaller-sized training sets [29], and its effectiveness in the imputation problem, as reported in the comparative study [27].

#### 4.2.1 Data Acquisition

For testing ADDIT-M, we obtain genotype data of phase 3 human chromosome 20 from the 1000 Genomes Project [38]. This data comprises 2504 individuals from 26 populations. We select a subset comprising 8 populations (GBR, TSI, CHS, STU, GIH, LWK, CHB, IT) and randomly mask 1%, 2%, and 5% of the data for subsequent imputation. We further use 75%, 90%, and 95% of the remaining phase 3 data as reference panels for running Beagle and IMPUTE2 and as the training set for the learning algorithm of ADDIT-M.

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2708701, IEEE/ACM Transactions on Computational Biology and Bioinformatics

TABLE 3
Comparison of the performance of ADDIT-NM with Beagle, LinkImpute, and IMPUTE2. The performance metrics include genotype error, runtime, and maximum memory footprint of the tools. For varying sizes of data containing varying proportions of missing genotypes, ADDIT-NM performed better than the other tools when tested on three real plant datasets of grape, apple, and maize.

| Dataset | # Samples ($N$) | # SNPs | # Missing Genotypes ($M_0$) | Method | Error (%) | Runtime (s) | Memory (MB) |
|---|---|---|---|---|---|---|---|
| Grape | 77 | 8506 | 2000 | Beagle | 11.0 | 16 | 371 |
| | | | | LinkImpute | 9.5 | 28 | 3465 |
| | | | | IMPUTE2 | 13.4 | 18 | 19 |
| | | | | **ADDIT-NM** | **2.6** | **14** | **7** |
| Apple | 711 | 8404 | 10000 | Beagle | 7.6 | 424 | 804 |
| | | | | LinkImpute | 7.4 | 104 | 6941 |
| | | | | IMPUTE2 | 9.2 | 98 | 45 |
| | | | | **ADDIT-NM** | **5.3** | **86** | **17** |
| Maize | 4300 | 43695 | 10000 | Beagle | 18.7 | 16585 | 927 |
| | | | | LinkImpute | 18.1 | 7608 | 11333 |
| | | | | IMPUTE2 | 21.4 | 7492 | 686 |
| | | | | **ADDIT-NM** | **8.7** | **7233** | **378** |

TABLE 4
Comparison of ADDIT-NM with quick impute (QI) and without (No QI).

| Dataset | Error (%) | | Runtime (min) | | Memory (MB) | |
|---|---|---|---|---|---|---|
| | QI | No QI | QI | No QI | QI | No QI |
| Grape | 2.6 | 3.0 | 0.2 | 1.1 | 7 | 7 |
| Apple | 5.3 | 5.5 | 1.4 | 4.9 | 17 | 17 |
| Maize | 8.7 | 9.0 | 120.6 | 417.0 | 378 | 378 |

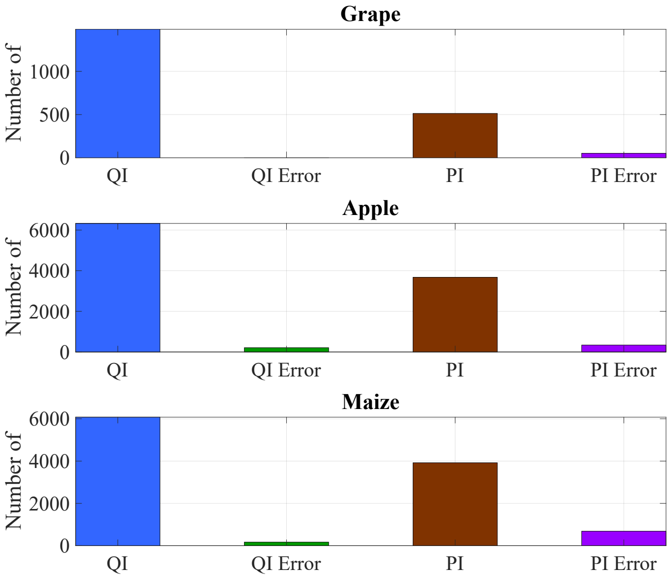

Fig. 3. Illustration of the number of quick imputations (QI) in ADDIT-NM: blue, quick imputation error (QI Error): green, priority-based imputations (PI): dark red, and priority-based imputation error (PI Error): purple, for the non-model organisms: grape, apple, and maize.

### 4.2.2 Comparative Analysis

To test our proposed ADDIT-M imputation algorithm, we again consider the overall imputation error percent, total runtime, and maximum memory used. For our experiments, since IMPUTE2 required at least 130× computation time higher than Beagle (also shown in [9]), we only consider Beagle for comparison with the now optional QI step of ADDIT-M (see previous section) turned off.

The results of our comparative study is tabulated in Table 5(A) and (B). In Table 5(A), we demonstrate the effect of increasing the proportion of missing values. We randomly fix 75% of the data as the reference panel containing no missing data, and mask the remaining data by 1%, 2%, and 5%. We note that ADDIT-M consistently requires less memory (about 1/4th) and demonstrates speedups of two orders of magnitude. Additionally, the overall imputation error percent is considerably lower for ADDIT-M. In Table 5(B), we demonstrate results for 5% missing data when the training set size varies amongst 95%, 90% and 75%. As expected, decreasing the number of training samples worsens the performance of the supervised learning algorithm: the total error percent for ADDIT-M gradually increases from 1.9% to 2.8%. Note that the error percent of Beagle exhibits a more accelerated increase with reduction of training size relative to our proposed approach.

### 4.2.3 When should we use QI?

As mentioned before, the effectiveness of QI is most pronounced when the adjacent alleles exhibit a low degree of double recombination. Although ADDIT-M with QI completes roughly 7% faster on the human data obtained from [38], it does perform worse (see Table 6) using a 75% training set and 5% missing data. In Figure 4, we illustrate the distribution of imputation errors over the three decision-making steps of ADDIT-M: the identical truth value (Step 2), QI (optional Step 3), and supervised learning based imputation (Step 4). Since we have a high level of trust in the reference genotype panel, we give Step 2 the highest priority in terms of determining the imputed value. Thus, the percentage of values imputed in Step 2 remains unaltered with and without QI. It is clear from the figure that the QI step only affects the other 70% of the missing values: specifically 17% of the missing genotypes are eligible for QI. Of these 17%, 10% are imputed incorrectly. The SVM performance, both with and without QI, are very similar and exhibit 4% imputation error (this is because the training samples are identical for both runs). It follows that for these data QI performs relatively worse as compared to using available reference data. Thus we do not recommend the QI step unless adjacent alleles exhibit low degrees of double recombination.

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2708701, IEEE/ACM Transactions on Computational Biology and Bioinformatics

TABLE 5
Comparative study of ADDIT-M (without quick impute) with Beagle. (A) Performance comparison of ADDIT-M versus Beagle on human data with fixed proportion of missing values and varying size of training set. (B) Performance comparison of ADDIT-M versus Beagle on human data with fixed training set and varying percentage of missing values.

**(A)** Fixed Training Set (75%)

| Missing (%) | Method | Error (%) | Runtime(s) | Mem(GB) |
|---|---|---|---|---|
| 1 | Beagle | 6.6 | 1900 | 3.8 |
| | ADDIT-M | 1.0 | 58.7 | 1.0 |
| 2 | Beagle | 8.1 | 2040 | 3.8 |
| | ADDIT-M | 2.5 | 64.5 | 1.0 |
| 5 | Beagle | 11.0 | 2080 | 3.8 |
| | ADDIT-M | 2.8 | 88.7 | 1.0 |

**(B)** Fixed Missing Genotypes (5%)

| Training Size (%) | Method | Error (%) | Runtime(s) | Mem(GB) |
|---|---|---|---|---|
| 95 | Beagle | 2.6 | 2200 | 3.5 |
| | ADDIT-M | 1.9 | 94.1 | 1.1 |
| 90 | Beagle | 5.1 | 2160 | 3.8 |
| | ADDIT-M | 2.6 | 91.7 | 1.1 |
| 75 | Beagle | 11.0 | 2080 | 3.8 |
| | ADDIT-M | 2.8 | 88.7 | 1.0 |

### 4.2.4 *Importance of Multi-class Supervised Learning*

We believed that using a supervised learning algorithm would enhance imputation accuracy. As a result, we expect that Beagle and ADDIT-M will outperform ADDIT-NM by exploiting the information embedded in the reference genotype panel. This is indeed the case, as deduced from Table 7. Among the two imputation tools for model organisms, ADDIT-M outperformed Beagle in terms of imputation accuracy, runtime, and memory. A considerable subset of the query data was filtered for identical truth (IT) and quick impute (QI)-based deduction, that lead to accurate and expedited imputation. For the remaining set of missing values, the supervised learning approach enabled accurate imputations. The results presented in Table 7 show that for model organisms, utilizing genotype information in reference panel, as in the case of Beagle and ADDIT-M, provide more accurate imputations.

TABLE 6
Comparison of genotype error (%), runtime (s), and maximum memory footprint (GB) with and without the quick impute step for ADDIT-M tested on human data.

| Dataset | Error (%) | | Runtime (s) | | Memory (GB) | |
|---|---|---|---|---|---|---|
| | QI | No QI | QI | No QI | QI | No QI |
| Human | 3.8 | 2.8 | 84.1 | 90.4 | 1.0 | 1.0 |

TABLE 7
Comparison of genotype error (%), runtime (s), and maximum memory footprint (GB) for Beagle, ADDIT-M, and ADDIT-NM when tested on human data. 75% of the data was used in the reference panel for imputing 5% of the masked data.

| Tools | Error (%) | Runtime (s) | Memory (GB) |
|---|---|---|---|
| Beagle | 11.0 | 2080 | 3.8 |
| ADDIT-M | 2.8 | 90 | 1.3 |
| ADDIT-NM | 14.6 | 1064 | 0.06 |

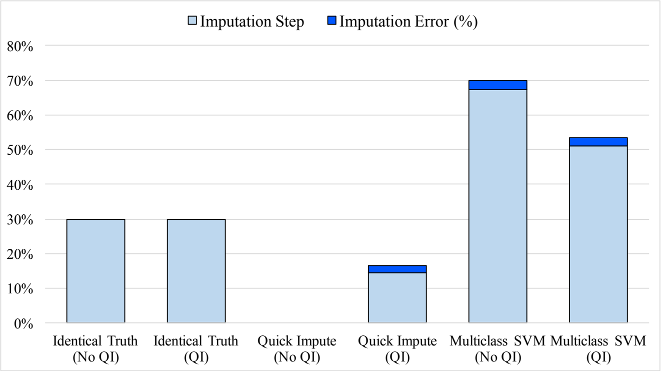

Fig. 4. Distribution of algorithm steps (identical truths, quick impute, and multi-class classification via SVM) used for imputation of ADDIT-M with and without the quick impute step in human data. The light blue lower blocks denote the percent of missing data that are imputed via each step in the ADDIT-M formalism for model organisms. The upper dark blue blocks denote the error (%) corresponding to each of those steps.

## 5 Conclusions

Genotype imputation is an essential precursor for improving the quality of haplotype phasing in applications like genome-wide association studies. Although model organisms can resort to available reference genotype panel for imputation, the problem becomes more challenging for non-model organisms that lack such reference data. Here, we present accurate and efficient window-based data-driven approaches for imputation of missing genotypes in both model and non-model organisms. We test our proposed methods on real datasets of non-model and model organisms, including humans. For varying sizes of data, proportions of missing genotypes, and sizes of training samples, our method consistently performs better than the leading tools like Beagle, IMPUTE2, and LinkImpute.

Although the multiclass classifier approach used in ADDIT-M generated accurate imputations, there still remains a scope to investigate other data-driven supervised learning approaches in Step 4 of section 3.2. One can further analyze

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

the performance of our imputation tools when plugged in to different phasing algorithms. Finally, a natural extension of genotype imputation is the devlopment of an accurate haplotype phasing mechanism for downstream analysis. In this regard, one can employ a graph-based phasing approach [39] or further explore sophisticated hidden Markov model-based algorithms.

## Acknowledgments

This work was supported by the Eck Institute for Global Health (EIGH) PhD fellowship to Choudhury and University support provided to Emrich as director of bioinformatics. We also thank the anonymous reviewers for their thought-provoking questions and improving suggestions.

## References

[1] L. A. Hindorff, P. Sethupathy, H. A. Junkins, E. M. Ramos, J. P. Mehta, F. S. Collins, and T. A. Manolio, "Potential etiologic and functional implications of genome-wide association loci for human diseases and traits," *Proceedings of the National Academy of Sciences*, vol. 106, no. 23, pp. 9362–9367, 2009.

[2] A. Kong, G. Masson, M. L. Frigge, A. Gylfason, P. Zusmanovich, G. Thorleifsson, P. I. Olason, A. Ingason, S. Steinberg, T. Rafnar *et al.*, "Detection of sharing by descent, long-range phasing and haplotype imputation," *Nature Genetics*, vol. 40, no. 9, pp. 1068–1075, 2008.

[3] S. A. Tishkoff, E. Dietzsch, W. Speed, A. J. Pakstis, J. R. Kidd, K. Cheung, B. Bonne-Tamir, A. S. Santachiara-Benerecetti, P. Moral, M. Krings *et al.*, "Global patterns of linkage disequilibrium at the CD4 locus and modern human origins," *Science*, vol. 271, no. 5254, pp. 1380–1387, 1996.

[4] H. Tao, D. R. Cox, and K. A. Frazer, "Allele-specific KRT1 expression is a complex trait," *PLoS Genetics*, vol. 2, no. 6, p. e93, 2006.

[5] C. C. Spencer, Z. Su, P. Donnelly, and J. Marchini, "Designing genome-wide association studies: sample size, power, imputation, and the choice of genotyping chip," *PLoS Genetics*, vol. 5, no. 5, p. e1000477, 2009.

[6] C. J. Willer, S. Sanna, A. U. Jackson, A. Scuteri, L. L. Bonnycastle, R. Clarke, S. C. Heath, N. J. Timpson, S. S. Najjar, H. M. Stringham *et al.*, "Newly identified loci that influence lipid concentrations and risk of coronary artery disease," *Nature Genetics*, vol. 40, no. 2, pp. 161–169, 2008.

[7] K. A. Frazer, D. G. Ballinger, D. R. Cox, D. A. Hinds, L. L. Stuve, R. A. Gibbs, J. W. Belmont, A. Boudreau, P. Hardenbol, S. M. Leal *et al.*, "A second generation human haplotype map of over 3.1 million SNPs," *Nature*, vol. 449, no. 7164, pp. 851–861, 2007.

[8] O. Choudhury, A. Chakrabarty, and S. J. Emrich, "HAPI-Gen: Highly Accurate Phasing and Imputation of Genotype Data," in *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM, 2016, pp. 78–87.

[9] B. L. Browning and S. R. Browning, "Genotype imputation with millions of reference samples," *The American Journal of Human Genetics*, vol. 98, no. 1, pp. 116–126, 2016.

[10] B. Howie, C. Fuchsberger, M. Stephens, J. Marchini, and G. R. Abecasis, "Fast and accurate genotype imputation in genome-wide association studies through pre-phasing," *Nature Genetics*, vol. 44, no. 8, pp. 955–959, 2012.

[11] D. Money, K. Gardner, Z. Migicovsky, H. Schwaninger, G.-Y. Zhong, and S. Myles, "LinkImpute: Fast and Accurate Genotype Imputation for Nonmodel Organisms," *G3: Genes, Genomes, Genetics*, vol. 5, no. 11, pp. 2383–2390, 2015.

[12] J. M. Hickey, B. P. Kinghorn, B. Tier, J. F. Wilson, N. Dunstan, and J. H. van der Werf, "A combined long-range phasing and long haplotype imputation method to impute phase for SNP genotypes," *Genet Sel Evol*, vol. 43, no. 12, pp. 10–1186, 2011.

[13] H. D. Daetwyler, G. R. Wiggans, B. J. Hayes, J. A. Woolliams, and M. E. Goddard, "Imputation of missing genotypes from sparse to high density using long-range phasing," *Genetics*, vol. 189, no. 1, pp. 317–327, 2011.

[14] A. G. Clark, "Inference of haplotypes from PCR-amplified samples of diploid populations." *Molecular biology and evolution*, vol. 7, no. 2, pp. 111–122, 1990.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (methodological)*, pp. 1–38, 1977.

[16] G. A. McVean and N. J. Cardin, "Approximating the coalescent with recombination," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 360, no. 1459, pp. 1387–1393, 2005.

[17] B. N. Howie, P. Donnelly, and J. Marchini, "A flexible and accurate genotype imputation method for the next generation of genome-wide association studies," *PLoS Genet*, vol. 5, no. 6, p. e1000529, 2009.

[18] M. Stephens, N. J. Smith, and P. Donnelly, "A new statistical method for haplotype reconstruction from population data," *The American Journal of Human Genetics*, vol. 68, no. 4, pp. 978–989, 2001.

[19] N. Li and M. Stephens, "Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data," *Genetics*, vol. 165, no. 4, pp. 2213–2233, 2003.

[20] S. R. Browning and B. L. Browning, "Haplotype phasing: existing methods and new developments," *Nature Reviews Genetics*, vol. 12, no. 10, pp. 703–714, 2011.

[21] 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," *The American Journal of Human Genetics*, vol. 78, no. 4, pp. 629–644, 2006.

[22] B. L. Browning and S. R. Browning, "A fast, powerful method for detecting identity by descent," *The American Journal of Human Genetics*, vol. 88, no. 2, pp. 173–182, 2011.

[23] Y. Li, C. J. Willer, J. Ding, P. Scheet, and G. R. Abecasis, "MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes," *Genetic epidemiology*, vol. 34, no. 8, pp. 816–834, 2010.

[24] J. A. Ward, J. Bhangoo, F. Fernández-Fernández, P. Moore, J. Swanson, R. Viola, R. Velasco, N. Bassil, C. A. Weber, and D. J. Sargent, "Saturated linkage map construction in Rubus idaeus using genotyping by sequencing and genome-independent imputation," *BMC genomics*, vol. 14, no. 1, p. 2, 2013.

[25] R. Upstill-Goddard, D. Eccles, S. Ennis, S. Rafiq, W. Tapper, J. Fliege, and A. Collins, "Support vector machine classifier for estrogen receptor positive and negative early-onset breast cancer," *PloS one*, vol. 8, no. 7, p. e68606, 2013.

[26] Y. V. Sun and S. L. Kardia, "Imputing missing genotypic data of single-nucleotide polymorphisms using neural networks," *European Journal of Human Genetics*, vol. 16, no. 4, pp. 487–495, 2008.

[27] A. Mikhchi, M. Honarvar, N. E. J. Kashan, and M. Aminafshar, "Assessing and comparison of different machine learning methods in parent-offspring trios for genotype imputation," *Journal of theoretical biology*, vol. 399, pp. 148–158, 2016.

[28] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, no. Aug, pp. 975–1005, 2004.

[29] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[30] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.

[31] M. Saar-Tsechansky and F. Provost, "Handling missing values when applying classification models," *Journal of Machine Learning Research*, vol. 8, no. Jul, pp. 1623–1657, 2007.

[32] M. Aly, "Survey on multiclass classification methods," *Neural Networks*, pp. 1–9, 2005.

[33] A.-F. Adam-Blondon, O. Jaillon, S. Vezzulli, A. Zharkikh, M. Troggio, R. Velasco, J. Martinez-Zapater *et al.*, "Genome sequence initiatives," *Genetics, Genomics, and Breeding of Grapes*, pp. 211–234, 2011.

[34] O. Jaillon, J.-M. Aury, B. Noel, A. Policriti, C. Clepet, A. Casagrande, N. Choisne, S. Aubourg, N. Vitulo, C. Jubin *et al.*, "The grapevine genome sequence suggests ancestral hexaploidization in major angiosperm phyla," *Nature*, vol. 449, no. 7161, pp. 463–467, 2007.

[35] R. Velasco, A. Zharkikh, J. Affourtit, A. Dhingra, A. Cestaro, A. Kalyanaraman, P. Fontana, S. K. Bhatnagar, M. Troggio,

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2017.2708701, IEEE/ACM Transactions on Computational Biology and Bioinformatics

D. Pruss *et al.*, "The genome of the domesticated apple (*Malus domestica Borkh*)," *Nature genetics*, vol. 42, no. 10, pp. 833–839, 2010.

[36] S. Hearne, C. Chen, E. Buckler, and S. Mitchell, "Unimputed GBS derived SNPs for maize landrace accessions represented in the SeeD-maize GWAS panel," 2014, [Online; accessed 21-February-2016].

[37] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.

[38] . G. P. Consortium *et al.*, "A global reference for human genetic variation," *Nature*, vol. 526, no. 7571, pp. 68–74, 2015.

[39] S. T O'Neil and S. J. Emrich, "Haplotype and minimum-chimerism consensus determination using short sequence data," *BMC genomics*, vol. 13, no. Suppl 2, p. S4, 2012.

**Olivia Choudhury** is a doctoral candidate and Eck Institute for Global Health (EIGH) Fellow at the Department of Computer Science and Engineering, University of Notre Dame, IN. She received a B.Tech. in Computer Science and Engineering from the West Bengal University of Technology, Kolkata, India. She is interested in bioinformatics, high performance computing, predictive modeling, and data-driven methods.

**Ankush Chakrabarty** is a postdoctoral fellow at the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA. He received a B.E. with first-class honors in Electrical Engineering at Jadavpur University, Kolkata, India, and received his Ph.D. in Automatic Control at the School of Electrical and Computer Engineering at Purdue University, West Lafayette, IN. He is interested in nonlinear systems, unknown input observers, biomedical control, and data-driven methods.

**Scott J. Emrich** received the BS degree in biology and computer science from Loyola College in Maryland and the PhD degree in bioinformatics and computational biology from Iowa State University. His research interests include computational biology, bioinformatics and parallel computing, including arthropod genome analysis with applications to global health and ecology. He is a member of the IEEE Computer Society.

1545-5963 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.