IMPUTOR: Phylogenetically Aware Software for Imputation of Errors in Next-Generation Sequencing

Matthew Jobin[1,2,*], Haiko Schurz[3], and Brenna M. Henn[4]

[1]Department of Anthropology, Santa Clara University

[2]UCSC Paleogenomics Lab, Department of Anthropology, University of California Santa Cruz

[3]Division of Molecular Biology and Human Genetics, Stellenbosch University, Tygerberg, South Africa

[4]Department of Anthropology and the Genome Center, University of California, Davis

*Corresponding author: E-mail: mjjobin@ucsc.edu.

Accepted: April 30, 2018

Abstract

We introduce IMPUTOR, software for phylogenetically aware imputation of missing haploid nonrecombining genomic data. Targeted for next-generation sequencing data, IMPUTOR uses the principle of parsimony to impute data marked as missing due to low coverage. Along with efficiently imputing missing variant genotypes, IMPUTOR is capable of reliably and accurately correcting many nonmissing sites that represent spurious sequencing errors. Tests on simulated data show that IMPUTOR is capable of detecting many induced mutations without making erroneous imputations/corrections, with as many as 95% of missing sites imputed and 81% of errors corrected under optimal conditions. We tested IMPUTOR with human Y-chromosomes from pairs of close relatives and demonstrate IMPUTOR's efficacy in imputing missing and correcting erroneous calls.

Key words: next-generation sequencing, parsimony, sequencing errors, imputation, phylogenetic tree.

Introduction

Advances in next-generation sequencing (NGS) have provided researchers with an unprecedented wealth of data, but short-read data have proven variable in its fidelity to the original sample sequence. Recent research has revealed evidence of mutagenic damage extant in commonly used online resources (Chen et al. 2017). A major difficulty for NGS is the separation of actual variation from spurious errors that arise from PCR amplification, library preparation, or even low-level cross-contamination among samples (DePristo et al. 2011). Numerous software pipelines have been constructed in order to process NGS data, with one important step being the assessment and filtering of mutational errors introduced by the NGS process. Both due to these filtering criteria and stochastic variation in read coverage, genomic data sets often contain missing variant calls for numerous sites (Wall et al. 2014; Bobo et al. 2016).

Missing variant calls are typically handled in two different ways. Population genomic data sets may merge samples under the assumption that missing calls represent the reference allele, leading to a reference bias. Alternatively, genomic "imputation" aims to fill-in missing variant calls by comparing a variant call-set to that of a set of reference genomes using haplotypic (i.e., linkage disequilibrium) information to identify similar haplotypes between the two data sets (Marchini and Howie 2010). This is a form of single imputation, that is, where an imputed site may then be used to make further imputations (Zhang 2016). Imputation with the 1000 Genomes Project (1000 Genomes Project Consortium et al. 2015) or other large human population genomic data sets is now standard practice (O'Connell et al. 2014). However, imputation may perform poorly in many diverse human data sets when the reference panel does not contain genetically similar populations (Huang et al. 2013), or imputation within a given experiment may be limited to the small number of genomes sequenced (Okada et al. 2015; Chou et al. 2016).

We recently developed an alternative imputation approach by leveraging the phylogenetic nature of DNA sequences (Wang et al. 2012; Poznik et al. 2013; Bobo et al. 2016). By creating a high confidence phylogenetic tree for a given locus, individual sequences assigned to a tip of the tree should carry all of the derived variant alleles up to the common root. If the sample sequence is missing a variant call, the call can be imputed by assuming the sequence carries the derived variant.

This approach can additionally take into account the possibility of reversions, which are assigned independent locations across the phylogeny. One major advantage to our approach is that the imputation of variants does not require access to an external reference panel of sequences, as long as the sample data set contains more than a handful of individuals.

We implement this approach in IMPUTOR, a software program that imputes mutations for a set of haploid nonrecombining samples via comparison of variants amongst phylogenetic near neighbors. IMPUTOR operates via the principle of parsimony, wherein neighboring sites on a phylogenetic tree that are identical by descent (IBD) for a derived allele are unlikely to experience a reversion to the ancestral allele amongst one of their members. Under the principle of parsimony, originally introduced as the "principle of minimum evolution," the course taken in evolutionary history is most likely to match the course that requires the fewest changes (Edwards and Cavalli-Sforza 1964). In addition to performing this function for variant calls marked as missing, IMPUTOR also searches for variants that are likely erroneous, which can appear on a phylogenetic tree as reversions. Previous studies imputing missing mutations have avoided introducing the possibilities for reversions due to their rarity (Wei et al. 2013), but a higher than expected rate of apparent reversions may indicate sequencing errors. This method of imputation does not require the use of a separate reference data set and can operate on any given haploid data.

Materials and Methods

IMPUTOR takes as input FASTA or VCF files, which are then processed so that only SNP data are handled (Pearson and Lipman 1988; Danecek et al. 2011). For VCF input files, the optional Genotype fields must be used with the GT format symbols to indicate a sample's allelic status. As per the VCF standard, a "." represents missing data, whereas a "0" indicates possession of the reference allele and a numeral of 1 or higher an ALT allele. A phylogenetic tree (either strictly bifurcating or allowing multifurcations) is also necessary for the parsimony-based imputation performed by the software. Users can either import a tree from an external source or generate such a tree from their own data. IMPUTOR provides four options for input: phyloXML import (Han and Zmasek 2009), tree construction by parsimony using Biopython (Cock et al. 2009), or tree construction using maximum likelihood methods via the software packages PhyML and RAxML (Guindon et al. 2010; Stamatakis 2014). Output consists of imputed sequence in FASTA or VCF format, along with a log of attempted imputations and ancillary information including the phylogenetic tree used in the process.

For each site in a set of variant data, IMPUTOR attempts to find the nearest neighbors on the given or constructed tree in order to determine whether an imputation should occur. The default mode of the software is to search a maximum of two steps rootward, and from there search a maximum of three steps leafward. These constraints, which tend to be on the conservative side in making imputations, were derived from tests for sensitivity and accuracy in imputing manually placed code changes, and serve to avoid finding neighbors from too far outside an isolated clade (see supplementary material, Supplementary Material online).

For missing data, an imputation is made if the target sample site's two nearest neighbors match one another and are non-missing, a method successfully used previously in human Y-chromosome data (Poznik et al. 2013). Multiple passes over the data can be performed in order to impute sites based on previous imputations. This last parameter is of use in cases where, for example, missing data can be confidently imputed and subsequently used to allow further imputations. Imputation can also be required if the genotype does not meet allelic depth (AD) and/or Phred-scaled genotype quality (GQ) thresholds. In both such options, the user may set a threshold below which an imputation is made, provided that the other instituted checks have been passed.

For other variable sites in a sample, if the three nearest neighbors carry an identical allele then the sample variant is changed to match the consensus, provided that the sample variant is also found elsewhere on the tree and it thus appears to be a reversion. If, on the other hand, no such other instance of the target site is found outside the near neighbors, the site is assumed to be a singleton mutation and is not imputed or corrected under the default settings. On a phylogenetic tree a reversion will appear when, from a target site, we find nonmatching neighbors, and then, continuing past those neighbors rootward, we encounter the ancestral allele again (Requeno and Colom 2016). The reversion check is active by default, forcing IMPUTOR to be conservative in its corrections of such sites, reflecting a choice to favor leaving some errors which appear to be singletons unaltered over erroneously imputing or correcting large numbers of singletons.

We provide three methods for gathering nearest neighbors in IMPUTOR. The first, **rootward** (see figs. 1 and 2), ascends toward the root from the target site up to a specified number of steps, whereas at each step descending leafward searching for potential neighbors to the target site. This method exits if it finds a threshold number of neighbors or if it exhausts the available branches. The second method, **hops**, counts the number of steps rootward and leafward needed to reach a neighbor from the target site. It returns a collection of neighbors should it find a sufficient number of them under a threshold number of steps rootward and/or leafward. The last method, **distance**, returns a collection of neighbors ordered by distance traversed along the branches from the target site. In order not to include more distant neighbors of isolated targets, a cutoff value is used to stop the search if the next branch length traversed is too high compared with the previous branch.

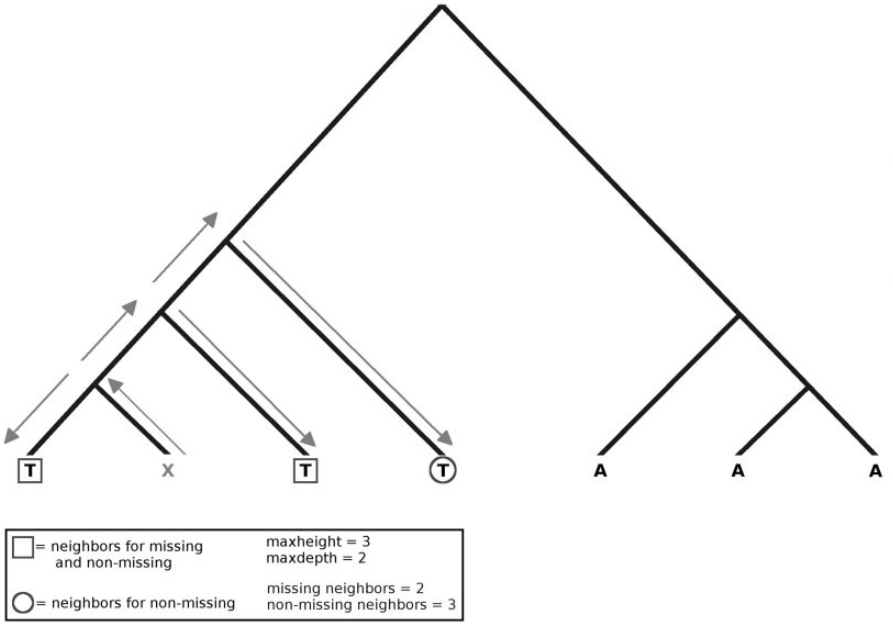

Fig. 1.—Rootward Case 1. A site found within a clearly defined clade of sufficient size to contain the threshold number of neighbors for both missing and nonmissing data.

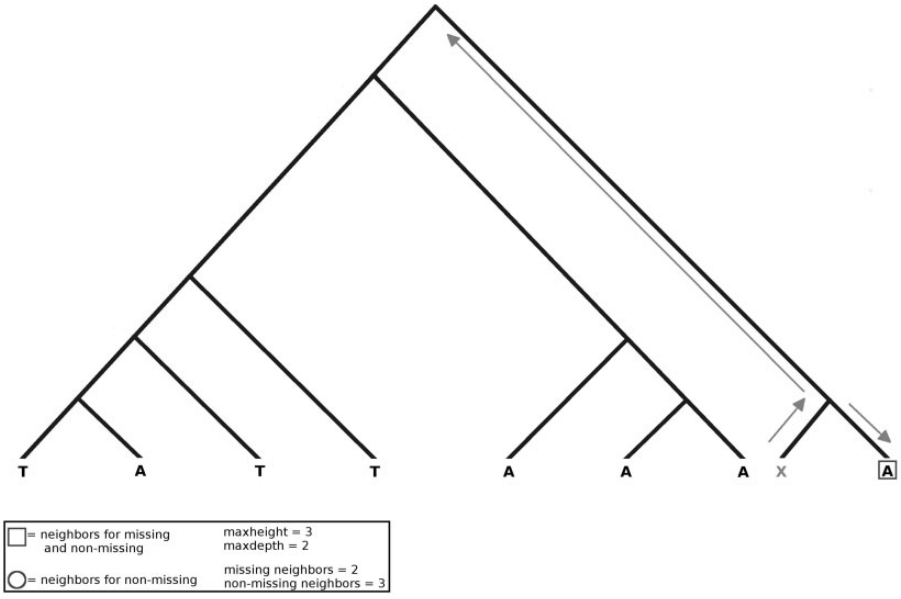

Fig. 2.—Rootward Case 2. A site with insufficient near neighbors to reach the threshold number for either missing or nonmissing data.

For each target site during each pass, IMPUTOR attempts, given the constraints imposed by the user, to determine whether it can find a sufficient number of near neighbors who share the same state but differ from that of the target. If that is true, and if the target appears to be a reversion and not a singleton, then the software will alter the target site to match the state of the neighbors, provided all other checks (including, for example, checks for AD and GQ) also pass. Figure 1 demonstrates a case where, using the **rootward** method, sufficient near neighbors have been found, whereas figure 2 demonstrates the opposite case, with insufficient near neighbors to proceed to imputation.

IMPUTOR's default settings can be altered by the user, allowing the informed researcher to impute at genotypes based on his or her own assessment of what might constitute erroneous data. In addition to altering the scope of the tree search for nearest neighbors mentioned above, the user can defeat the reversion check for all the data or only the data

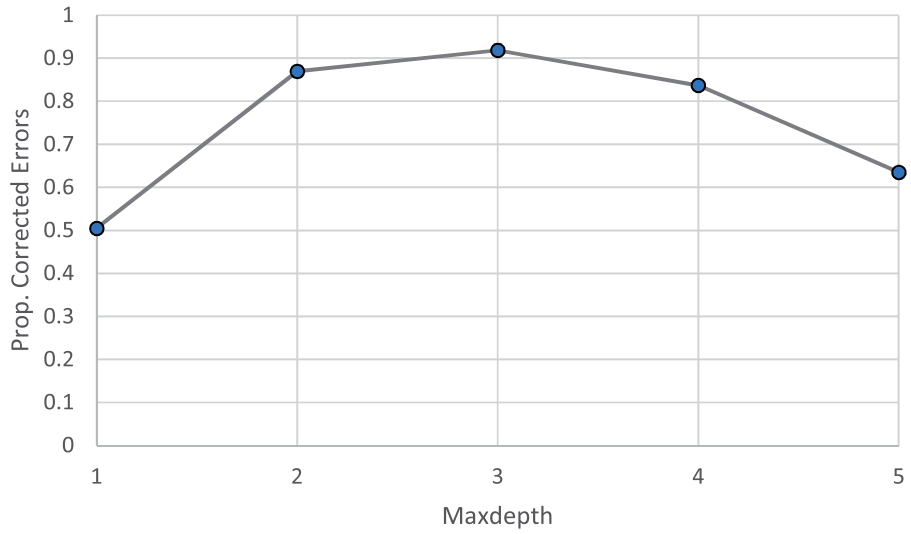

FIG. 3.—Proportion of corrected errors as a function of the **maxdepth** parameter for missing sites and **rootward** neighbor collection method.

under a threshold coverage; change the neighbor-gathering method type, shape parameters and number of compute threads used by PhyML or RAxML; allow the possibility of imputing a missing site with one missing neighbor and one nonmissing; and change the number of passes IMPUTOR takes to attempt imputation of the data.

Results

Father–Son Pairs on the Y Chromosome

Y-chromosome sequence from known father–son or male sibling pairs provide an excellent test scenario for IMPUTOR. Using six human duo and sib pairs Illumina sequenced to 10.1× coverage, we tested the accuracy of IMPUTOR under different options. Assuming a mutation rate of 3.07×10^{-8} mutations per base pair per generation (Helgason et al. 2015) for 9.8 Mb of nonrecombining Y sequence, the number of differences between any pair of father–son Y chromosomes is expected to be ~0.3/pair or twice that for male siblings. However, before imputation, these pairs differed by a multitude of both missing and nonmissing sites. We created a phylogeny based on maximum likelihood using the RAxML software. After iterating systematically through all possible options, the best combination resulted in the proportion of pairwise differences between two individuals reduced to 0.012 of the original distance, averaged across six known pairs, with all but one case reducing the difference between members of a pair to zero (supplementary fig. 42, Supplementary Material online).

Simulated Data

Using simulated data generated by the forward-in-time simulation SFS_CODE (Hernandez 2008), a variety of data configurations, error types and tree construction methods were compared for accuracy of imputation. Missing sites and nonmissing errors were randomly induced throughout a data file, which was then run through IMPUTOR, after which its output was compared with the original, error-free data. Averaging the results of ten randomly altered data files for missing sites can yield up to 95.8% final accuracy as gauged by the similarity of the imputed output file to the original (i.e., no missingness) file. A slight increase in accuracy can result from performing ten passes through the data (see supplementary fig. 40, Supplementary Material online). Changes in the number of sequences in the input file, number of neighbors used to impute/correct, and height or depth of tree searched all affect the proportion of corrected errors (see Supplementary Section 2, Supplementary Material online). Similar tests using sequencing errors yielded up to 81.5% similarity with optimal parameters. IMPUTOR can simultaneously impute and correct for both sequencing errors and missing data. For missing data, tree searches to a depth of three steps tend to result in the greatest accuracy, along with lower requirements for the number of nearest neighbors before flagging a site for imputation (see fig. 3).

Checks for reversions prevent spurious imputations or corrections (see table 1). This check, which is configured by default but can be defeated by the user, only allows imputation or correction when a site appears to be a reversion, leaving alone terminal-branch singleton mutations. The increase in accuracy gained by this feature is especially noticeable in simulated data when the proportion of introduced errors is low (see Supplementary Material online).

For sequencing errors and the **hops** neighbor-collection method, the parameter **maxhops**, which constrains the number of "hops" up and down the branches of the tree may be taken in the search for a neighbor, has the greatest effect on

Table 1
Effect of the Reversion Check Feature of IMPUTOR on Accuracy

Method	Reversion Check	Mean Imputed Distance	S.D. Imputed Distance	Prop. Corrected Errors
Rootward	Y	6.10	2.77	**0.91**
Hops	Y	3.90	2.55	**0.95**
Distance	Y	5.00	2.21	**0.93**
Rootward	N	13.3	1.57	**0.82**
Hops	N	12.1	2.28	**0.84**
Distance	N	13.0	2.36	**0.82**

NOTE.—Simulated data generated in SFS_CODE was randomly altered to create ten new files, replacing bases with missing data. These altered files, which had a mean number of pairwise differences of 73.7 from the original file (S.D. 10.15) were then run in IMPUTOR. The "Prop. Corrected Errors" column above is a metric of accuracy in recovering the original sequence.

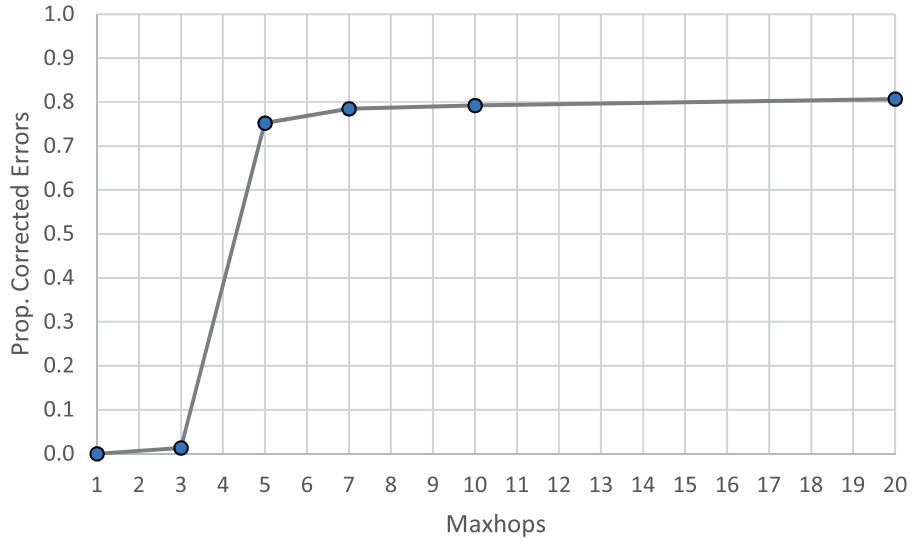

FIG. 4.—Proportion of corrected errors as a function of the **maxhops** parameter for nonmissing sites and **hops** neighbor collection method.

proportion of corrected errors (see figure 4). After a steep rise at a **maxhops** value of 4, accuracy nearly plateaus, moving slowly to a maximum proportion of 0.807 at a **maxhops** of 20. Higher values (not shown) show a gradual decrease in proportion of corrected errors.

The number of sequences, and thus the size of the phylogenetic tree, can have a significant effect on IMPUTOR's ability to impute and/or correct. Changes to the number of sequences demonstrate a relationship to the introduced missingness in SFS_CODE-generated files, as demonstrated below. The missingness was applied as a random chance that any particular site might be replaced with a missing site; low numbers of sequences coupled with low missingness can create scenarios where no actual missingness is generated. In the case of extremely low numbers of sequences (~10), even with a missingness rate sufficient to introduce missing sites, the tree is not sufficient to accurately impute.

SHAPEIT is a software program for phasing from sequencing data, capable also of imputation of missing data (Delaneau et al. 2013). To compare the behavior of IMPUTOR and SHAPEIT for haploid data, ten files were generated with randomly introduced missing sites at two levels of missingness, for multiple sample sizes ranging from 10 to 10,000 sequences. The samples were marked as male for use in SHAPEIT, whose chrX function as used to enable haploid imputation. For each level of missingness and each sample size, the randomly altered files were run in each program, with the mean proportion of corrected errors shown below.

Discussion

IMPUTOR is capable of imputing missing genotypes and correcting erroneous variant calls without use of an external reference panel. This makes IMPUTOR ideal for small sequencing experiments. A phylogeny derived from an external panel is, alternatively, an option for increasing imputation accuracy. IMPUTOR is also capable of correcting nonmissing sites that appear to be reversions in the sample sequence (i.e., due to sequencing errors or reference bias). The primary factors governing the accuracy of IMPUTOR are the phylogenetic tree (generated or input) and the optional method by which the tree is searched for neighbors. The guide information relayed in supplementary material, Supplementary Material online, along with the instruction manual for the software, indicate

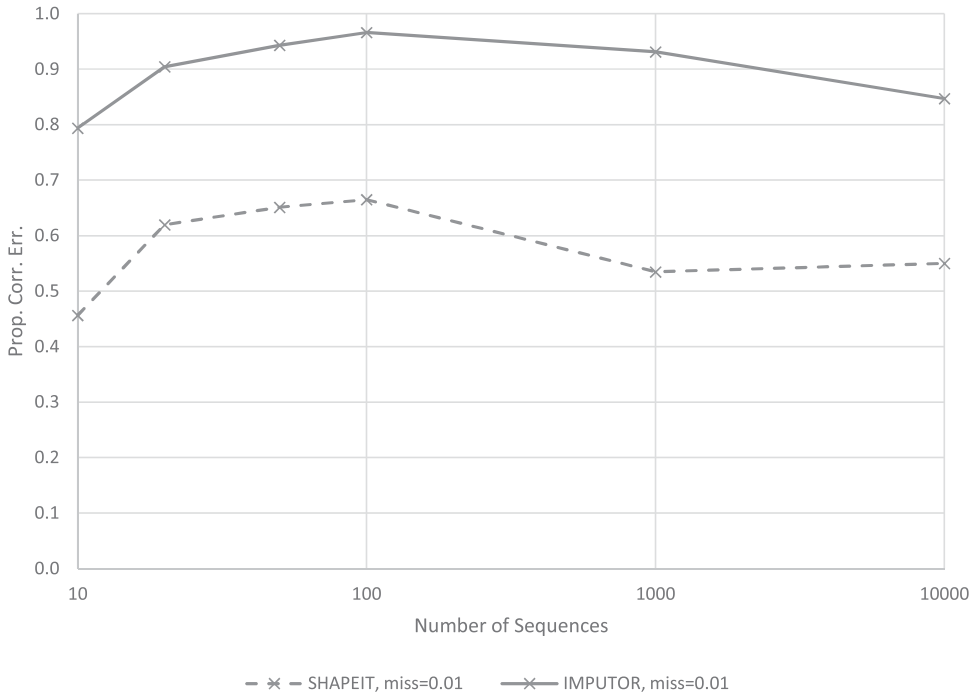

Fig. 5.—Proportion of corrected errors as a function of the number of sequences, for a missingness of 0.01 and $\Theta = 0.01$, for two software programs, SHAPEIT and IMPUTOR.

the best default setting for most applications, along with a display of speed versus accuracy trade-offs for option selection.

Since IMPUTOR relies on a phylogenetic tree and the principle of parsimony in order to make imputations and corrections, the accuracy of that tree is important for proper functioning of the software. While accurate results can be achieved with a small data set, the genetic diversity of the members of that set will have an effect on the structure of that tree and the confidence the user can place in it, and thus accordingly the quality of the results output. In the case of a low-diversity sample set, the structure of the tree may not be such that IMPUTOR will be able to flag a site for imputation. When using trees with low diversity and large numbers of sequences, apparent multifurcations will interfere with IMPUTOR's ability to reliably find groups of neighbors, leading to the decrease in accuracy seen for large sample sizes in figure 5. This figure was generated from simulated data at a fixed rate of mutation for all sample sizes; in general, if a tree appears to be insufficiently resolved to the user, then IMPUTOR's power to make imputations and/or corrections will be similarly reduced. While this fact can limit the kinds of data on which IMPUTOR can fruitfully operate, the software defaults to avoiding imputation except in the cases outlined in the Materials and Methods section. Thus, whereas a low-diversity data set might pose challenges to the improvement of the output sequence, IMPUTOR does not default to making spurious imputations or corrections. Very low-diversity data sets will cause the linked tree construction software such as RAxML to return an error, thus providing information to the user about unworkable data.

Table 2

Effect of Θ on Ratio of Imputed to Unimputed Pairwise Distances to an Original SFS_CODE-generated File

Θ	Prop. Corrected Err.	Variance	Failed/10
0.001	0.93	0.00195	0
0.0001	0.89	0.0148	2
0.00001	0.82	n/a	9

NOTE.—The unimputed file was created by randomly replacing bases with missing codes at a frequency of 0.001, simulating damage. Ten iterations of simulation were run for each value of Θ, with mean and variance shown. RAxML would not run on the number of entries in the Failed/10 column.

In order to illustrate the effect of diversity on imputation accuracy in IMPUTOR, we generated ten iterations of forward-in-time simulated data in SFS_CODE for three different levels of Θ, where $\Theta = 2 \times P \times N_e\mu$, where μ is the mutation rate per site and where P is the ploidy (see table 2). Ten files for each value of Θ were randomly altered so that one in one thousand of the sites was changed to missing data. These randomly damaged files were then run through IMPUTOR to evaluate the accuracy of the imputation process; the ratio of imputed to unimputed files' pairwise distance to the original, undamaged file ("Prop. Corrected Err., below"), used to gauge the accuracy. Decreasing Θ decreases the genetic

diversity of the sample, which results in a decrease in mean accuracy and an increase in variance in accuracy.

At even smaller sample sizes (e.g., $n = 10$) or much lower sequence lengths (see supplementary material, Supplementary Material online), VCF files contain levels of genetic diversity that are too low to be usable for the purposes of imputation and correction by use of a phylogenetic tree. Thus, while IMPUTOR can accurately impute and correct data for relatively small data sets without use of an external reference, a combination of adequate sample size, sequence length, and sequence diversity must be present in order to construct a reliable tree on which to base imputations and corrections.

The proportion of reversions that will occur on a phylogenetic tree will be quite small by comparison to the total number of mutations (see supplementary material, tables 15 and 16, Supplementary Material online). Furthermore, for a real reversion to be erroneously changed by IMPUTOR, the site will need to pass all of the other constraints, which default to avoiding imputation unless a number of constraints are satisfied such as minimal number of neighbors which vary from that target site but match one another. Only terminal-branch reversions are likely to satisfy these conditions under the default parameter values. Additionally, user-defined thresholds such as AD, GQ and minimum coverage can be useful in screening data to distinguish between actual and erroneous reversions.

Data Availability

The source code is available at https://github.com/mjobin/Imputor.

Supplementary Material

Supplementary data are available at *Genome Biology and Evolution* online.

Acknowledgments

The authors would like to acknowledge F. Mendez, C. Gignoux, C. Bustamante, and M. Feldman for sharing data and their helpful contributions to this manuscript.

Funding

Research reported in this publication was supported by the South African Medical Research Council. The content is the sole responsibility of the author (H.S.) affiliated with the South African Medical Research Council and does not necessarily represent the official views of the South African Medical Research Council. B.M.H. was supported by the Research Foundation at Stony Brook University. M.J. was funded by UCSC Paleogenomics.

Literature Cited

1000 Genomes Project Consortium, et al. 2015. A global reference for human genetic variation. Nature 526:68–74.
Bobo D, et al. 2016. False Negatives Are a Significant Feature of Next Generation Sequencing Callsets Cold Spring Harbor Labs Journals.
Chen L, et al. 2017. DNA damage is a pervasive cause of sequencing errors, directly confounding variant identification. Sci Mag. 355(6326):752–756.
Chou W-C, et al. 2016. A combined reference panel from the 1000 Genomes and UK10K projects improved rare variant imputation in European and Chinese samples. Sci Rep. 6(1):1–9.
Cock PJA, et al. 2009. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics 25(11):1422–1423.
Danecek P, et al. 2011. The variant call format and VCFtools. Bioinformatics 27(15):2156–2158.
Delaneau O, et al. 2013. Haplotype estimation using sequencing reads. Am J Hum Genet. 93(4):687–696.
DePristo MA, et al. 2011. A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nat Genet. 43(5):491–498.
Edwards A, Cavalli-Sforza L. 1964. Reconstruction of evolutionary trees. Syst Assoc Publ No. 6:67–76.
Guindon S, et al. 2010. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. Syst Biol. 59(3):307–321.
Han MV, Zmasek CM. 2009. phyloXML: xML for evolutionary biology and comparative genomics. BMC Bioinformatics. 10:356.
Helgason A, et al. 2015. The Y-chromosome point mutation rate in humans. Nat Genet. 47(5):453–457.
Hernandez RD. 2008. A flexible forward simulator for populations subject to selection and demography. Bioinformatics 24(23):2786–2787.
Huang L, et al. 2013. Genotype imputation in a coalescent model with infinitely-many-sites mutation. Theor Popul Biol. 87:62–74.
Marchini J, Howie B. 2010. Genotype imputation for genome-wide association studies. Nat Rev Genet. 11(7):499–511.
O'Connell J, et al. 2014. A general approach for haplotype phasing across the full spectrum of relatedness. PLoS Genet. 10(4): e1004234–e1004221.
Okada Y, et al. 2015. Construction of a population-specific HLA imputation reference panel and its application to Graves' disease risk in Japanese. Nat Genet. 47(7):798–802.
Pearson WR, Lipman DJ. 1988. Improved tools for biological sequence comparison. Proc Natl Acad Sci USA. 85(8):2444–2448.
Poznik GD, et al. 2013. Sequencing Y chromosomes resolves discrepancy in time to common ancestor of males versus females. Science 341(6145):562–565.
Requeno JI, Colom JM. 2016. Evaluation of properties over phylogenetic trees using stochastic logics. BMC Bioinformatics. 17(1):1–14.
Stamatakis A. 2014. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics 30(9):1312–1313.
Wall JD, et al. 2014. Estimating genotype error rates from high-coverage next-generation sequence data. 24:1734–1739.
Wang JR, et al. 2012. Imputation of single-nucleotide polymorphisms in inbred mice using local phylogeny. Genetics 190(2):449–458.
Wei W, et al. 2013. A calibrated human Y-chromosomal phylogeny based on resequencing. Genome Res. 23:388–395.
Zhang Z. 2016. Missing data imputation: focusing on single imputation. Ann Transl Med. 4(1).

Associate editor: Dan Graur