Imputor User Guide

Matthew Jobin

1. Overview

IMPUTOR is a software program written in the Python language for the purpose of identifying miscalled bases in sequence or variant data. Advances in next-generation sequencing have provided researchers with a wealth of data, but the data generated has proven variable in its fidelity to the original sequence **(Bobo et al. 2016; Wall et al. 2014)**. Numerous software pipelines have been constructed in order to process data, with one important step being the assessment and correction of chemical or mechanical errors introduced by the NGS process. In some cases, these methods have used reference sequences derived from sequencing methods with greater coverage or reliability, while some have also implemented methods to impute missing data via the principle of parsimony **(Wang et al. 2012; Poznik et al. 2013)**. IMPUTOR imputes missing or suspected mis-called bases via parsimony, identifying sites where a combination of mechanical error and conservative base calls may have misidentified the site.

2. Setup and installation

2.1. Python packages

IMPUTOR is written in Python 2.7 and requires some standard Python packages to run properly. Python distributions vary in what comes installed—in the case where a needed package has not been installed, the recommended method for installing packages is the **PIP installer program**. Following the given link should allow the user to install all needed Python packages. The list of needed packages for this release of IMPUTOR follows:

- biopython
- progressbar2

2.2. External programs

Two external maximum likelihood software programs can be used with IMPUTOR in order to construct phylogenetic trees. Please note that for these programs to work with IMPUTOR, they must be compiled and the executable placed in the same folder as IMPUTOR or elsewhere in the system's search path.

2.2.1. PhyML

Unlike methods of maximum parsimony, maximum likelihood methods make use of the branch lengths of a proposed tree .

2.2.1.1. How to install PhyML for use with Imputor

1. Download and install PhyML for your system as per the **PhyML user guide**.
2. Rename the resulting executable ‘phyml’ (no quotes).
3. Move your phyml executable to a location that is part of your system’s search path. On UNIX-like systems including macOS and Linux, you will usually want to add the line **export PATH="/your path/:$PATH**" to the .bashrc or the .bash profile located in your home directory (where your path is the path to the directory that contains PhyML binary).

2.2.2. RAxML

1. Download and install RAxML for your system as described in the **RAxML user guide**.
2. Compile the version best suited to your needs. Please see the RaXML docs about parallel computing, makefiles, etc.
3. Rename your compiled executable to 'raxmlHPC' (no quotes).
4. Move your raxmlHPC executable to a location that is part of your system's search path. On UNIX-like systems including macOS and Linux, you will usually want to add the line **export PATH="/your path/:\$PATH**" to the .bashrc or the .bash profile located in your home directory (where your path is the path to the directory that contains RAxML binary).

3. Input

IMPUTOR accepts input from two file types: FASTA files and VCF files. A reference sequence must also be input VCF format files, in order to generate the sequences for every sample. For VCF files, the sequence of the reference file is used to generate sequence from the defined variants inside the software, allowing either FASTA or VCF format output.

A phylogenetic tree is also necessary for the phylogeny-aware imputation performed by the software, and thus it is necessary either to import or generate such a tree. IMPUTOR provides four options for input: phyloxml import, tree construction by parsimony using Biopython, maximum likelihood via PhyML or maximum likelihood via RAxML. Software options necessary to modify the output of the tree have been included in IMPUTOR, however for detailed descriptions of their functions the individual software packages' under manuals will be the user's best guide (see below).

3.1. Input options

Imputor handles the following input arguments:

- -file (required): The raw input data, in FASTA or VCF format. Note that VCF format files must include the optional FORMAT column with the GT subheader, and include columns where individual samples are marked.
- -ref (required): The reference sequence, in either TXT or OBJ format. In the case of FASTA files, the first sequence can be copied into a separate reference sequence file for the purposes of generating variants.
- -tree: The method for importing or constructing a tree, with the following options:
  - *.xml: Input of a file with extension .xml will invoke reading as a phyloxml format tree **(Han & Zmasek 2009)**.
  - PhyML: Invokes the PhyML software **(Guindon et al. 2010)**.
  - RAxML: Invokes the RAxML software **(Stamatakis 2014)**.
- -alpha: The value of the gamma shape parameter used in PhyML. Default is *e*.
- -boot: The number of bootstrap replicates for PhyML. Please see the **PhyML user guide** for detailed instructions . Default is 100.
- -rmodel: The model type for RAxML. Please see the **RAxML user guide** for instructions. Default is GTRCAT.
- -mutrate: The point mutation rate. Default is 8.7e-10.
- -threshold: The likelihood at which IMPUTOR will accept that a potential back mutation is actual, and avoid imputing at the target site. Default is 0.05.
- -tstv: Transition/transversion ratio. Default is 2.0.

4. Output

IMPUTOR provides output information to the console for the researcher to examine during run-time. After the run completes, a number of possible files can be written to storage, depending on the options chosen from the list in the next section.

4.1. Output options

- -out: The type of output file requested. Options are: fasta or vcf. Defauls to fasta.
- -outimpute: Write the list of imputed mutations to <inputfile minus extension> + “impute.txt”. Defaults to true.
- -outtree: The output file format of the phylogenetic tree, with the following options:
    - newick: Newick format
    - nexus: Nexus format
    - phyloxml (default): phyloxml format
- -impout: Output list of imputed sites and their likelihoods. Default is true.

5. Imputation

A major challenge for Next Generation Sequencing is the separation of actual variation from errors that arise from the mechanical process of the method **(DePristo et al. 2011)** .Imputation methods have been used to identify mistakenly called sites due to mechanical or laboratory error **(Wang et al. 2012)** . These efforts are often made in order to “fill in” calls marked “missing” by the sequencing process **(Lippold et al. 2014)** . One of the primary methods by which imputation is carried out for Next-Generation Sequencing (NGS) is by comparing NGS sequence reads to that of a set of reference sequences **(Marchini & Howie 2010)** . An example of such a reference sequence set is Sanger sequencing, which is slower than NGS but has a lower error rate **(Wall et al. 2014)** .

5.1. Imputation by parsimony

IMPUTOR imputes mutations for an input set of data via comparison of variants amongst near neighbors, via the principle of parsimony, wherein neighboring samples 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 & Cavalli-Sforza 1964)**. The principle of parsimony or minimum evolution has been applied via a phylogenetic tree for the purposes of imputing missing data **(Poznik et al. 2013)**. In addition performing to this function, IMPUTOR searches for likely candidates for falsely negatively called mutations, which can appear in NGS as reversions aka back mutations. Previous studies imputing missing mutations have avoided introducing the possibilities for reversions due to their rarity **(Wei et al. 2013)**.

One possible scenario for a false negative call is an apparent back mutation on a the phylogenetic tree. On such a tree, a back mutation will appear, from the point of view of any site if, when following its descendants we encounter a different allele, and then, continuing on the same descending path, encounter the original allele again **(Requeno & Colom 2016)**. The same logic holds in the reverse direction, that if while ascending toward the root of a phylogenetic tree, one finds a different allele than the target allele, after which, on continuing to ascend, one finds a reversion to the target allele. Back mutations are not impossible, but rare, since the occurrence of a mutation on a lineage must be compounded with that of another on a branch of that lineage that happens to revert to the ancestral allele.

According to the principle of parsimony, should a target site differ from its three nearest neighbors, and should those three neighbors possess the same allele, and furthermore that target site be found among more distant neighbors (implying a possible reversion), the more likely scenario requiring fewer mutations would be the one generating the putative reversion than the mutations necessary to separately generate the mutations in its non-matching near neighbors. The length of the private branch upon which the target sequence sits represents the proportion of changes in that sequence relative t the sequence size. That length, or distance, multiplied by the rate of mutation, is the likelihood that a reversion has truly occurred, since at a constant rate of mutation the branch implies time. A short private branch is less likely to have provided the opportunity for a true reversion to have occurred, and is thus more likely that an identified reversion is the result of NGS sequencing error or an error of some other form.

The decision rule for imputation involves each sequence and its closest neighbors on the constructed or imported phylogenetic tree. IMPUTOR searches site by site for each sample, iterating through the list of samples in random order and implementing the following decision rule:

1. For each target, iterated in random order:
    1. Step back toward the root of the tree collecting nearest siblings of the target
    2. Keep stepping back until root is reached or threshold number of siblings found (default = 3)
2. For each variant in the set of all variants in the input file:
    1. If the target is marked as missing:
        1. If the nearest two neighbors are the same and not missing, impute their state for the target **(Poznik et al. 2013)**
    2. If the target is not marked as missing:
        1. If the target does not match its three nearest neighbors and the three neighbors match each other and are non-missing
            1. If the target state is found farther away than the three nearest neighbors (i.e. a potential reversion **(Requeno & Colom 2016)**
                1. Calculate the likelihood that a back mutation could have occurred under these circumstances (Poisson):
                2. If the chance of back mutation multiplied by the given transition or transversion ratio does not exceed the user-supplied threshold chance (default = 0.05) impute the target to be the same as its neighbor

5.1.1. Probability of Mutation

The rate of point mutation has been assessed for various compartments of the human genome, e.g. 8.71 x 10^{-10} for much of the Y chromosome **(Helgason et al. 2015)**.

The chance that a particular mutational event might occur is governed by the Poisson distribution. The probability of observing *k* events during a period of time is given by the probability mass function of the Poisson distribution:

$$P = (\lambda t)^k e^{-\lambda t}/k! \qquad (1)$$

where λ is the rate of mutation, *t* is time on the branch and k is the number of events, in this case always 1. The above function operates for a single observation of the Poisson distribution function, and is thus applicable to each observation of a potential imputation. The choice of whether to impute is thus set using a user-specified tolerance threshold with a default of 0.05. In other words, once IMPUTOR calculates that the chance for a back mutation to occur along the branch exceeds the threshold, it assumes that the back mutation is real and does not impute. Otherwise, the software assumes an mistake has been made and imputes.

6. References

Bobo, D. et al., 2016. *False Negatives Are a Significant Feature of Next Generation Sequencing Callsets*, Cold Spring Harbor Labs Journals. Available at: http://biorxiv.org/lookup/doi/10.1101/066043.

DePristo, M.A. et al., 2011. A framework for variation discovery and genotyping using next-generation dNA sequencing data. *Nature Genetics*, 43(5), pp.491–498. Available at: http://www.nature.com/ng/journal/v43/n5/full/ng.806.html%3FWT.ec_id=NG-201105.

Edwards, A.W.F. & Cavalli-Sforza, L.L., 1964. Reconstruction of evolutionary trees. *Syst. Assoc. Publ.,* No. 6, pp.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), pp.307–321. Available at: http://sysbio.oxfordjournals.org/content/59/3/307.full.

Han, M.V. & Zmasek, C.M., 2009. PhyloXML: XML for evolutionary biology and comparative genomics. *BMC bioinformatics*, 10(1), pp.356–6. Available at: http://www.biomedcentral.com/1471-2105/10/356.

Helgason, A. et al., 2015. The y-chromosome point mutation rate in humans. *Nature Publishing Group*, 47(5), pp.453–457. Available at: http://dx.doi.org/10.1038/ng.3171.

Lippold, S. et al., 2014. Human paternal and maternal demographic histories: Insights from high-resolution y chromosome and mtDNA sequences. *Investigative Genetics*, 5(1), p.13. Available at: http://investigativegenetics.biomedcentral.com/articles/10.1186/2041-2223-5-13.

Marchini, J. & Howie, B., 2010. Genotype imputation for genome-wide association studies. *Nature Reviews Genetics*, 11(7), pp.499–511. Available at: http://dx.doi.org/10.1038/nrg2796.

Poznik, G.D. et al., 2013. Sequencing Y chromosomes resolves discrepancy in time to common ancestor of males versus females. *Science*, 341(6145), pp.562–565. Available at: http://www.sciencemag.org/cgi/doi/10.1126/science.1237619.

Requeno, J.I. & Colom, J.M., 2016. Evaluation of properties over phylogenetic trees using stochastic logics. *BMC bioinformatics*, pp.1–14. Available at: http://dx.doi.org/10.1186/s12859-016-1077-7.

Stamatakis, A., 2014. RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9), pp.1312–1313. Available at: http://bioinformatics.oxfordjournals.org/content/30/9/1312.full.

Wall, J.D. et al., 2014. Estimating genotype error rates from high-coverage next-generation sequence data. *Genome Research*, 24(11), pp.1734–1739. Available at: http://genome.cshlp.org/content/24/11/1734.full.

Wang, J.R. et al., 2012. Imputation of Single-Nucleotide Polymorphisms in Inbred

Mice Using Local Phylogeny. *Genetics*, 190(2), pp.449–458. Available at: http://www.genetics.org/cgi/doi/10.1534/genetics.111.132381.

Wei, W. et al., 2013. A calibrated human y-chromosomal phylogeny based on resequencing. *Genome Research*, 23(2), pp.388–395. Available at: http://genome.cshlp.org/content/23/2/388.full.