AlphaPlantImpute2

Release 1.5.3

Steve Thorn, Andrew Whalen, John M Hickey

Sep 21, 2020

CONTENTS:

CHAPTER

ONE

INTRODUCTION

AlphaPlantImpute2 is a genotype imputation algorithm for haploid and diploid plants. AlphaPlantImpute2 supports individuals genotyped with SNP array data.

Please report any issues to alphagenes@roslin.ed.ac.uk.

1.1 Conditions of use

AlphaPlantImpute2 is part of a suite of software that the AlphaGenes group has developed. It is only availble for academic use. All other interested parties are requested to contact John M Hickey (John.Hickey@roslin.ed.ac.uk) to discuss the terms of use.

1.2 Authorship

AlphaPlantImpute2 is part of a body of imputation software developed by the AlphaGenes group under John M Hickey. It is based on the hidden Markov model (HMM) developed by Roberto Antolín in AlphaImpute (Antolín et al. 2017), and is similar to the algorithm used in MaCH (Li et al. 2010). AlphaPlantImpute2 was written by Steve Thorn and Andrew Whalen.

1.3 Disclaimer

While every effort has been made to ensure that AlphaPlantImpute2 does what it claims to do, there is absolutely no guarantee that the results provided are correct. Use of AlphaPlantImpute2 is at your own risk.

CHAPTER

TWO

ALGORITHM DESCRIPTION

AlphaPlantImpute2 splits imputation into two distinct parts: haplotype library creation and imputation. These steps are described below.

2.1 Haplotype library creation

AlphaPlantImpute2 will create a haplotype library if the `-createlib` option is given.

Haplotype library creation starts with randomly phasing the genotypes of high-density individuals. High-density individuals are those with a fraction of non-missing markers greater than a given threshold, which can be specified with the `-hd_threshold` option. Homozygous loci are de-facto phased; heterozygous loci are randomly assigned with equal probability and missing loci are randomly assigned according to allele frequency.

The haplotypes are refined in an iterative process. Each haplotype in the library is refined using a hidden Markov model (HMM), which can be a haploid model (`-haploid`), diploid model (the default), or *joint* model, which models loci as either haploid or diploid (`-joint`). The hidden states (at each locus) are either haplotypes in the library, or pairs of haplotypes. The model randomly generates a haplotype, or pair of haplotypes, according to the HMM probabilities. At each iteration, the number of haplotypes considered by the HMM is reduced by randomly sampling a number of haplotypes (`-n_haplotypes`). This number is a trade-off between higher accuracy (higher numbers of haplotypes) and faster computation time (lower numbers). Once each haplotype has been generated, the process iterates for a number of rounds (`-n_sample_rounds`) to refine the haplotypes. A small number of rounds (for example, 5) is usually sufficient to accurately estimate the haplotypes.

Once a haplotype library has been built, it can be updated with new genotypes using a similar process. The new genotypes are iteratively phased using the existing haplotypes from the library as reference. The newly phased haplotypes are added to the library and then written to file. Haplotype libraries can be reused multiple times to impute low-density individuals for the same crop.

2.2 Imputation

AlphaPlantImpute2 will impute individuals if the `-impute` option is given.

Each individual is imputed using the same hidden Markov model used for library creation, with haplotypes from the provided haplotype library (`-library` or `-libphase`) as hidden states. The number of haplotypes considered (`-n_haplotypes`) is reduced by a *targeted* sampling of haplotypes from the library. Targeted sampling chooses haplotypes that have the fewest opposite homozygous markers compared to the focal individual within a window of markers. The number of windows the chromosome is divided into is specified by `-n_windows`. Haplotypes are randomly sampled in the case where many haplotypes have the same number of opposite homozygous markers. The hidden Markov model determines the probabilities of the possible genotypes for a focal individual at all loci. These probabilities are used to calculate the genotype dosages (expectation value of the genotypes) and called genotypes (the genotype with the largest probability). The `-calling_threshold` option can be used to call uncertain loci

as missing. For example, with the diploid HMM, there are three probabilities per locus — one for each of the three possible genotypes: {0, 1, 2}. Setting a calling threshold of 0.4 would set an imputed marker to missing if the maximum probability of the three possible genotypes was less than this threshold. A value less than 0.333 (diploid) or less than 0.5 (haploid) will not change any marker to missing.

If pedigree information is known, it can be used to target haplotypes used for imputation. A founders file (`-founders`) specifies the parents or founders of the focal individuals. The imputation is then restricted to using only founder haplotypes contained in the haplotype library.

CHAPTER

THREE

PROGRAM OPTIONS

AlphaPlantImpute2 takes a number of command line arguments to control its behaviour. To display a list of arguments, use either `AlphaPlantImpute2` or `AlphaPlantImpute2 -h`.

3.1 Core options

```
-createlib            Create a haplotype library.
-impute               Impute individuals.
-out OUT              The output file prefix.
```

The `-createlib` and `-impute` options specify which kind of task AlphaPlantImpute2 should do.

The `-out` argument specifies the output file prefix. For example, with `-out imputed`, AlphaPlantImpute2 outputs the imputed genotypes to `imputed.ped` and genotype dosages to `imputed.dosages`.

3.2 Input options

```
-ped [PED [PED ...]]  A genotype file in PLINK plain text format (.ped)
-library LIBRARY      A haplotype library file in PLINK plain text format
                      (.ped)
-founders FOUNDERS    A file that gives the founder individuals for each
                      individual.
-startsnp STARTSNP    The first marker to consider. The first marker in the
                      file is marker '1'. Default: 1.
-stopsnp STOPSNP      The last marker to consider. Default: all markers
                      considered.
-seed SEED            A random seed to use for debugging.
-genotypes [GENOTYPES [GENOTYPES ...]]
                      A file in AlphaGenes format.
-libphase LIBPHASE    A haplotype library file in AlphaGenes phase format
                      (.phase)
-decode [DECODE [DECODE ...]]
                      Decode .ped files to AlphaGenes format.
```

AlphaPlantImpute2 supports genotype files in PLINK plain text (.ped) format as specified by the `-ped` and `-library` options. AlphaGenes format is also supported with the `-genotypes` and `-libphase` options. It is not recommened to mix the two file types in an imputation workflow.

3.3 Algorithm options

```
-hd_threshold HD_THRESHOLD
                      Fraction of non-missing markers required to classify
                      an individual as high-density. Only high-density
                      individuals are used to build the haplotype library.
                      Default: 0.0.
-n_haplotypes N_HAPLOTYPES
                      Number of haplotypes to sample from the haplotype
                      library. Default: 100.
-haploid              Run using a haploid HMM instead of the default diploid
                      HMM.
-joint                Run using a joint HMM instead of the default diploid
                      HMM.
-calling_threshold CALLING_THRESHOLD
                      Genotype calling threshold. Use a value less than 0.25
                      for best-guess genotypes. Default: 0.1.
-n_sample_rounds N_SAMPLE_ROUNDS
                      Number of rounds of library refinement. Default: 10.
-n_windows N_WINDOWS  Number of windows for targeted haplotype sampling.
                      Default: 5.
-error ERROR          Genotyping error rate. Default: 0.01.
-recomb RECOMB        Recombination rate per chromosome. Default: 1.
-incorrect_loci       When building a haplotype library, print the average
                      number of loci that were incorrectly sampled from the
                      hidden Markov model.
-overwrite            Overwrite imputed genotypes with original values at
                      non-missing loci. Default: False
```

The `-incorrect_loci` option displays diagnostic information that can be used to assess the quality of haplotype library creation. It shows the average number (averaged over individuals) of loci that were incorrectly sampled from the hidden Markov model. A smaller number is better. In particular, it shows when additional sampling rounds (`-n_sample_rounds` option) give no improvement (the number of incorrect loci does not improve). If this happens, the accuracy can be increased by increasing the number of sampled haplotypes (`-n_haplotypes`) rather than using more sampling rounds.

3.4 Multithreading options

```
-maxthreads MAXTHREADS
                      Maximum number of threads to use for analysis.
                      Default: 1.
-iothreads IOTHREADS  Number of threads to use for input and output.
                      Default: 1.
```

AlphaPlantImpute2 supports multithreading for running the analysis, and for reading in and writing out (large amounts of) data. The parameter `-maxthreads` controls the number of threads used by the analysis. The parameter `-iothreads` controls the number of threads used for input and output — setting this option to a value greater than 1 is only recommended for very large files (>10,000 individuals).

CHAPTER

FOUR

EXAMPLE USAGE

4.1 Population based imputation

The first step in a population imputation scenario is to create a haplotype library from high-density genotyped individuals.

```
AlphaPlantImpute2 -createlib -out library -ped high_density.ped
```

The haplotype library is saved to `library.ped`. Low-density, focal individuals are then imputed using this library.

```
AlphaPlantImpute2 -impute -out imputed -library library.ped -ped low_density.ped
```

Imputed dosages are saved to `imputed.dosages` and imputed genotypes to `imputed.ped`.

4.2 Pedigree based imputation

When the pedigree is known, this information can be used to choose specific haplotypes (from the haplotype library) to be used for imputation. A typical workflow starts with the creation of a haplotype library from high-density, founder genotypes.

```
AlphaPlantImpute2 -createlib -out library -ped high_density_founders.ped
```

The founders for each focal individual are specified with the `-founders` option. Then, the imputation of the focal individuals only uses the corresponding founder haplotypes.

```
AlphaPlantImpute2 -impute -out imputed -library library.ped -ped low_density_
↪offspring.ped -founder founders.txt
```

4.3 Updating a haplotype library

A haplotype library can be updated with new genotypes. The following example updates the library `library.ped` with genotypes from the file `genotypes.ped`. The updated library is saved to `library_new.ped`.

```
AlphaPlantImpute2 -createlib -library library.ped -ped genotypes.ped -out library_new
```

4.4 Converting PLINK .ped files to AlphaGenes format

PLINK .ped files can be converted to AlphaGenes format with the `-decode` option. The coding from alleles in the .ped file to AlphaGenes genotypes is determined by the first .ped file given. At any locus the first allele encountered in the file is coded as 0 and the second as 1. The genotypes written to the AlphaGenes format file is then the sum of the two alleles. The following, converts two .ped files to two AlphaGenes files (`library.genotypes` and `imputed.genotypes`) using the allele coding interpreted from `library.ped`.

```
AlphaPlantImpute2 -decode library.ped imputed.ped
```

CHAPTER

FIVE

INPUT FILE FORMATS

5.1 PLINK .ped files

PLINK .ped files contain genotypes for each individual. AlphaPlantImpute2 reads the individuals' identifiers from the second field (the within family identifier). The allele calls are read from field seven onwards. Fields are assumed to be whitespace delimited. Missing data are represented with 0. All other fields are ignored. The following example shows four individuals genotyped on four markers.

Example:

```
0 id1 0 0 0 0 A A G G 0 0 C C
0 id2 0 0 0 0 A G G T T C C T
0 id3 0 0 0 0 G G T T C C C C
0 id4 0 0 0 0 A A G G T C C C
```

5.2 AlphaGenes genotypes file

AlphaGenes genotypes files contain genotypes for each individual. The first value in each line is the individual's identifier. The remaining values are the genotypes of the individual at each locus, either 0, 1, or 2 (or 9 if missing). Fields are whitespace delimited. The following example shows four individuals genotyped on four markers.

Example:

```
id1 0 2 9 0
id2 1 1 1 1
id3 2 0 2 0
id4 0 2 1 0
```

5.3 Founders file

A founders file is a whitespace delimited file containing the founders for each individual to be imputed. Each line has the individual's identifier followed by a number of identifiers for each of the founders. Any number of founders is accepted. The following example shows four individuals and their founders.

Example:

```
id10 id1 id2
id11 id1 id2
id12 id3 id4 id5
id13 id3 id4 id5
```

CHAPTER

SIX

OUTPUT FILE FORMATS

6.1 PLINK .ped files

Haplotype libraries and imputed genotypes are written as PLINK .ped files when the `-ped` option is given. Haplotype libraries are written as phased alleles - the first alleles in each pair are for one haplotype, the second alleles are for the other haplotype. Imputed genotypes are written in an unphased way such that the order of alleles has no significance.

6.2 AlphaGenes format files

Haplotype libraries and imputed genotypes are written in AlphaGenes genotype format when the `-genotypes` option is given. Imputed genotypes are written in the format as described for input files above. Haplotype libraries are written in a similar format, except with one haplotype per line. The following example shows the two haplotypes for two individuals:

Example:

```
id1 0 1 1 0
id1 0 1 1 0
id2 0 0 0 1
id2 1 1 1 0
```

6.3 Dosage file

The dosage file gives the expected allele dosage for the alternative allele (coded as 1) for each imputed individual. The first value in each line is the individual's identifier. The remaining values are the allele dosages at each locus. If there is a lot of uncertainty in the underlying genotype calls, this file can often prove more accurate than the called genotype values.

Example:

```
id1    0.0503    2.0000    1.7890    0.0001
id2    1.0000    1.0000    1.0000    1.0000
id3    2.0000    0.0000    2.0000    0.0001
id4    0.0000    2.0000    1.0000    0.0000
```

CHAPTER

SEVEN

REFERENCES

Antolín, R., Nettelblad, C., Gorjanc, G., Money, D., Hickey, J.M., 2017. A hybrid method for the imputation of genomic data in livestock populations. Genetics Selection Evolution 49, 30.

Li, Y., Willer, C.J., Ding, J., Scheet, P., Abecasis, G.R., 2010. MaCH: Using Sequence and Genotype Data to Estimate Haplotypes and Unobserved Genotypes. Genet Epidemiol 34, 816–834.

Rabiner, L., 1986. An introduction to hidden Markov models. IEEE ASSP Magazine 4–16.