Implementation of Imputation Algorithm

Michelle Parker

August 28, 2013

Abstract

The aim of this document

Contents

1 Introduction

The statistical definition of imputation is "a procedure for entering a value for a specific data item where the response is missing or unusable" [1]. In genetics, imputation refers to the process of inferring genotypes that are either missing, have a low quality score or are not directly assayed in sampled individuals. This technique is used increasingly in genome-wide association and whole genome sequencing studies.

Imputation has several uses: to boost power in a study by inferring erroneous, low confidence or sporadic missing data; to refine genotype probabilities; and to infer untyped data by using a reference panel containing a larger set of genetic markers.

In genome-wide association studies (GWAS), a number of genetic variants, usually single-nucleotide polymorphisms (SNPs), are assayed in groups of individuals with and without the disease of interest. The selection of genetic variants used in the study are often genotyped using DNA microarrays; distinct types of microarray are composed of different selections of genetic variants across the genome.

Multi-marker association analysis, used in GWAS, identifies markers that are independently associated to the disease of interest. Imputing sporadic missing data can make it easier to interpret the results of these analyses and can also boost power. There may also be a number of genotype calling errors due low confidence data. Imputation can be used to correct these errors which may help to control false-positive associations. [2].

A reference panel is a collection of samples genotyped at a dense set of markers across the genome. These known haplotypes can be used to impute genetic variants that are not included on the SNP array used in the study. This increases the number of SNPs that can be tested for association and advances fine-mapping studies where the location of the causal variant is identified. This technique can also be used in meta-analysis, where two or more studies conducted on different platforms are combined. Stronger associations may be found with untyped SNPs than those that are genotyped even if the untyped SNP is poorly tagged, because imputation algorithms estimate haplotypes taking into account multiple markers surrounding the missing genotype [3].

The process of imputation is illustrated in Figure 1. Imputation is based on the identification of stretches of haplotypes which are shared between individuals. These regions are known as being identical-by-descent (IBD) and would have originated from identical copies of the same ancestral allele. Regions of IBD are much longer in closely related individuals than in unrelated individuals. In both related and unrelated samples, imputed haplotypes are modelled as mosaics of haplotypes in the reference panel. When there is uncertainty over which haplotype is IBD, many imputation programs take this into account and summarize the information probabilistically. Genotype likelihoods give a more accurate representation of the data and can often lead to more significant results when used in downstream analysis [3].

In low coverage whole genome sequencing, such as in the 1000 Genomes Project [4], imputation is used to assist genotype likelihood estimation and calling. Relatively small amounts of sequence data across many individuals is combined to accurately estimate the genotypes for each sample. In other words, the imputation algorithms use the sample data itself as its own reference panel. Imputation can also be extended to include non-SNP variation such as copy number variation, classical human leukocyte antigen alleles, and short insertions and deletions (indels) [3].

There are many different methods for imputation based on different statistical models, but imputation algorithms can be divided into two main categories: those which take into account all observed genotypes during its computation of a single sample (IMPUTE, MACH, fastPHASE/BIMBAM), and those which only look at a window of markers either side of the missing genotype (BEAGLE, PLINK) [2]. Using all observed genotypes is computationally expensive but in many cases can achieve a more accurate result. Factors which affect imputation accuracy include the size and choice of reference panel; the difference in genetic diversity between the study population and the reference panel; use of tagging methods for SNP array design. In almost all cases, the larger the reference panel, the more accurate the imputation. The algorithm implemented in this report uses a probabalistic graphical model and is described in the BEAGLE paper [5].

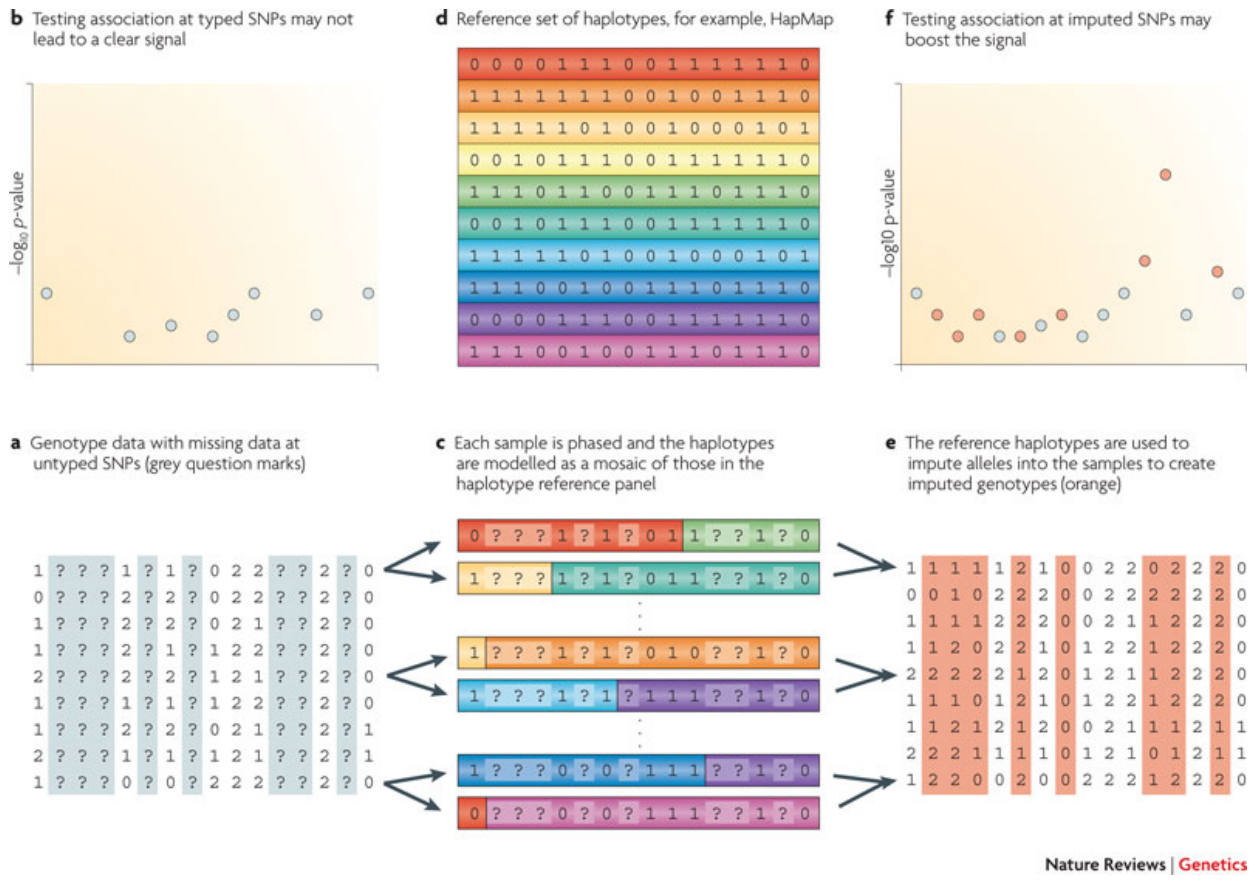

Figure 1: This figure illustrates imputation of untyped SNPs in a sample of unrelated individuals using a reference panel [3].
Samples in the study are genotyped at a small number of sites using a microarray (**a**). A reference panel (**d**) of denser SNPs is used to infer the SNPs that have not been directly genotyped in the samples (**c**). In this process the samples are phased and the untyped SNPs are imputed (**e**).
Testing association on just the SNPs which are directly assayed (**a**) may not lead to a significant result (**b**) but testing on all SNPs after imputation (**e**), stronger associations on inferred SNPs may be found (**f**).

2 Methods

The BEAGLE algorithm has a similar structure to the expectation maximization (EM) algorithm [6]. An initial estimate of haplotype phase is obtained for each individual and then at each iteration, the model is fit and an improved estimate of haplotype phase per individual is sampled. The final output haplotypes are obtained by calculating the most likely haplotype pairs per individual conditional on the model in the final iteration.

The process can therefore be split into 3 sections: model building, where a probabilistic graphical model is used to implicitely cluster haplotypes; sampling, where haplotypes are re-estimated and used as input into the next model build; and a maximization step, where the most likely haplotypes are computed based on the current model.

The haplotype-clustering model which is built defines a Hidden Markov Model (HMM) where a modified forward-backward can be used for the sampling step and the Viterbi algorithm can be used for the final stage. The BEAGLE recommendation is to run 10 iterations of the model building and sampling steps before carrying out the Viterbi algorithm.

2.1 Model Building

The model is represented as a directed acyclic graph. Assuming no missing alleles and a number of samples genotyped at M markers the directed graph has the following properties:

1. The graph is levelled with $M + 1$ levels. Each node of the graph is a member of a level that corresponds to the position in the sequence of markers.
2. At level 1, there is one root node which has no incoming edges. At level $M + 1$ there is one terminal node which has no outgoing edges.
3. All incoming edges to a node at level m have a parent node at level $m - 1$. All outgoing edges from the node at level m have a child node at level $m + 1$.
4. Each edge at level m is marked by a single allele a. Two edges originating from the same parent node can not be labelled with the same

allele.

5. There exists a path from the root node to the terminal node for each haplotype in the samples such that the mth allele of the haplotype is the label of the mth edge of the path.

NOTES When two edges are directed inot the same node, this node represents the union of histories represented by the incoming edges. This represents historical recombination and in terms of the probabilistic model, this represents the Markov property where the sequence of alleles two before the node is independent of the sequence of markers two after the node????? not sure about this reworded in beagle 3 paper Example of conditional probability between markers in beagle 1 paper

A node represents represents a collection of possible allele sequences up to the marker level. An edge represents a cluster of haplotypes which have allele a at position m and are similar enough locally to be grouped together at that position

Node markov property. recombination. sequence of alleles before node
END OF NOTES

Haplotype	Count
1111	21
1112	79
1122	95
1221	116
2111	25
2112	112
2122	152

Figure 2: Summary genotype data from 300 individuals

The model building algorithm will be exemplified using data in Figure 2. The algorithm begins by inserting genotype data from each indivudal into the graph. The data is processed a marker at a time. If the phase is not known, the data is randomly phased. If the marker is missing, the marker is randomly imputed based on population allele frequencies at that

marker. Each path from the root node to the terminal node represents a single haplotype and the results inserting data from Figure 2 into the model format is shown in Figure 3.

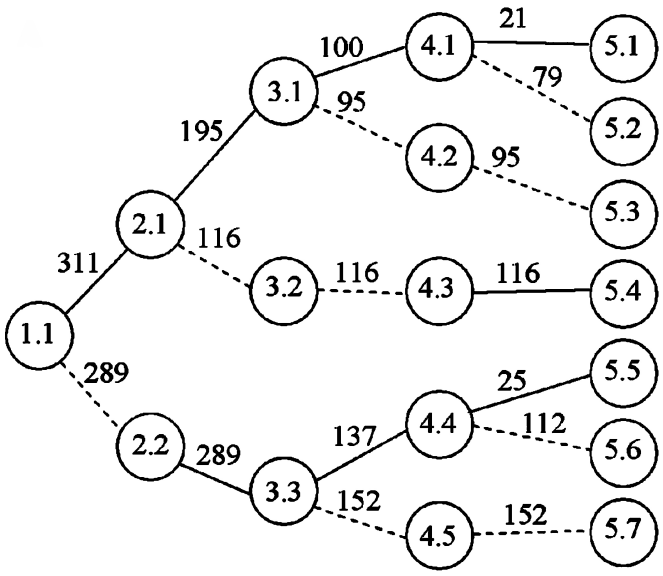

Figure 3: Tree constructed using summary genotype data in Figure 1. Circles are and edges are lines. Solid edge is the allele 1 and dashed edge represents the allele 2. Numbers above the edges represent counts. This graph is directional from left to right. [7]

After this tree has been created, it then needs to be compressed into the graph format described at the beginning of this section. The tree is compressed into a graph format by merging pairs of nodes on each level which are sufficiently similar. The algorithm iterates through each level at a time and merges all pairs of nodes where transition probabilities of all downstream nodes pass a certain threshold. The merging algorithm is illustrated below using Figure 3 as an example. All nodes at the final level are merged to create the single terminal node that represents the cluster of all haplotypes after being processed by the model.

The similarity score between nodes x and y on level $m-1$ is calculated as follows. Let n_x be the haplotype count at node x and n_y be the haplotype count at node y. For allele a_m at level m, let $n_x(a_m)$ be the count of haplotypes which pass through node x and begin with marker a. Similarly, let $n_y(a_m)$ be the count of haplotypes which pass through node y and begin

with marker a. Continuing, $n_x(a_m, a_{m+1})$ and $n_y(a_m, a_{m+1})$ are haplotype counts which pass through nodes x and y respectively and follow with the sequence of alleles a_m, a_{m+1}. The observed conditional probability difference diff_{xy} between node x and y for given sequence of alleles $a_m, a_{m+1}, ..., a_{m+k}$ is

$$\text{diff}_{xy} = \left| \frac{n_x(a_m, a_{m+1}, ..., a_{m+k})}{n_x} - \frac{n_y(a_m, a_{m+1}, ..., a_{m+k})}{n_y} \right|$$

The similarity score for nodes x and y is the maximum diff_{xy} over $k = 0, 1, 2, ..., M - m$ and all possible $a_m, a_{m+1}, ..., a_{m+k}$.

The algorithm iterates through each level in the graph and calculates the similarity score between all pairs of nodes and merges the pair that have the lowest similarity score below the corresponding threshold. The threshold between two nodes x and y is defined as $(n_x^{-1} + n_y^{-1})^{\frac{1}{2}}$.

Consider Figure 3 as an example. The similarity threshold for nodes 2.1 and 2.2 is

$$\left(\frac{1}{311} + \frac{1}{289} \right)^{\frac{1}{2}} = 0.082$$

Beginning the calculation of the similarity score of these two nodes, first calculate $\text{diff}_{2.1,2.2}$ for allele $a_2 = 1$.

$$\text{diff}_{xy} = \left| \frac{195}{311} - \frac{289}{289} \right| = 0.373$$

The similarity score is the maximum score of all diff_{xy} and is therefore at least 0.373. This exceeds the cutoff at 0.082 and therefore these two nodes will not be merged.

At level 3, similarity scores for all pairs are calculated. The threshold for nodes 3.1/3.2 is

$$\left(\frac{1}{195} + \frac{1}{116} \right)^{\frac{1}{2}} = 0.117$$

For $a_3 = 1$,

$$\text{diff}_{xy} = \left| \frac{100}{195} - \frac{0}{116} \right| = 0.513$$

which does exceeds the threshold and therefore nodes can not be merged.

The threshold for nodes 3.2/3.3 is

$$\left(\frac{1}{116} + \frac{1}{289} \right)^{\frac{1}{2}} = 0.110$$

For $a_3 = 1$,

$$\text{diff}_{xy} = \left| \frac{0}{116} - \frac{137}{289} \right| = 0.474$$

which exceeds the threshold and therefore nodes can not be merged.

The threshold for nodes 3.1/3.3 is

$$\left(\frac{1}{195} + \frac{1}{289} \right)^{\frac{1}{2}} = 0.093$$

diff_{xy} is calculated for every suffix combination. For $a_3 = 1$,

$$\text{diff}_{xy} = \left| \frac{100}{195} - \frac{137}{289} \right| = 0.039$$

For $a_3 = 2$,

$$\text{diff}_{xy} = \left| \frac{95}{195} - \frac{152}{289} \right| = 0.039$$

For $a_3 = 1, a_4 = 1$,

$$\text{diff}_{xy} = \left| \frac{21}{195} - \frac{25}{289} \right| = 0.021$$

For $a_3 = 1, a_4 = 2$,

$$\text{diff}_{xy} = \left| \frac{79}{195} - \frac{11}{289} \right| = 0.018$$

For $a_3 = 2, a_4 = 2$,

$$\text{diff}_{xy} = \left| \frac{95}{195} - \frac{152}{289} \right| = 0.039$$

Thus the similarity score for nodes 3.1/3.3 is the maximum of all diff_{xy}.

$$max(0.039, 0.039, 0.021, 0.018, 0.039) = 0.039$$

This score is less than the threshold value as therefore these two nodes can be merged. When two nodes are merged, the counts of all downstream haplotypes are summed. Once the merge has occured, the similarity score would be calculated for all new nodes on the next level and so on. In this example, there are no nodes merged on level 4. All nodes are then merged on the final level. The resultant graph is shown in Figure 4.

2.2 Forward-Backward Algorithm

2.3 Viterbi Algorithm

2.4 Advantages of BEAGLE algorithm

NOTES How beagle scales in number of samples and number of markers Notes: Why the beagle model is good

Does not need any prior information about haplotype frequency distribution. Models which require this type of estimation break down as the number of markers and samples increases because the number of observable haplotypes and the frequencies of these haplotypes become too small to estimate directly. If all feasible haplotypes are considered then this is very computationally expensive. Other models include coalescent models which implies that haplotypes that are similar to the ones we have already seen are more likely to be seen again since changes in haplotypes occur thorugh recombination and mutation. Can produce very accurate results but can be computationally expensive as well Other methods have made use of haplotype blocks but althorugh usefuly it does not adequately explain all the corelation struction between markers because LD can extend beyond block boundaries and can have more complex patterns within blocks Automatically adapts to the amoung of linkage disequilibrium between markers. No need to choose haplotype window size or select tagging markers. Explain what linkage disequilibrium is - correlation structure between markers Using a sliding window approach does not take into account how LD structure varies across the genome or areas where LD does not exist due to recombination hotspots Clusters haplotypes to improve power.

Correlation between markers is a localized phenomenom because LD decays with distance. Needs to be taken into account in phasing algorithms

Markov property paper 3

How beagle works

Localized haplotype-cluster model is an empirical LD model that adapts to the local structure in the daata. Relative to other methods it does well with large data sets. This methods can essentially be thought of as similar to an EM approach where and initial guess athaplotype phase is made, the model is fit, improved estimates of the haplotype phaseis made and the model

is refit etc END OF NOTES

3 Implementation

4 Testing and Results

5 Future work

Pycuda parallel computation

References

[1] Economic Commission for Europe of the United Nations (UNECE), *Glossary of Terms on Statistical Data Editing*, Conference of European Statisticians Methodological Material, Geneva. 2000

[2] Yun Li, Cristen WIller, Serena Sanna & Goncalo Abecasis, *Genotype Imputation*, Annual Review of Genomics and Human Genetics, Volume 10, Issue 1, 2009, Pages 387-406

[3] Jonathan Marchini & Bryan Howie, *Genotype Imputation for Genome-Wide Association Studies*, Nature Reviews Genetics, Volume 11, Issue 7, July 2010, Pages 299-511

[4] 1000 Genomes Project Consortium, *An integrated map of genetic variation from 1,092 human genomes*, Nature, Volume 491, Issue 7422, October 2012, Pages 56-65

[5] Sharon R. Browning & L. Brian Browning, *Rapid and Accurate Haplotype Phasing and MIssing-Data Inference for Whole-Genome Association Studies By Use of Localized Haplotype Clustering*, The American Journal of Human Genetics, Volume 81, Issue 5, November 2007, Pages 1084-1097

[6] Chuong B. Do & Serafim Batzoglou, *What is the expectation maximization algorithm?*, Nature Biotechnology, Volume 26, Issue 8, August 2008, Pages 897-899

[7] Sharon R. Browning, *Multilocus Association Mapping Using Variable-Length Markov Chains*, The American Journal of Human Genetics, Volume 78, Issue 6, June 2006, Pages 903-913