UPPSALA
UNIVERSITET

IT Licentiate theses
2022-003

Computational statistical methods for genotyping biallelic DNA markers from pooled experiments

Camille Clouard

UPPSALA UNIVERSITY
Information Technology

UPPSALA
UNIVERSITET

Computational statistical methods for genotyping biallelic DNA markers from pooled experiments

Camille Clouard
camille.clouard@it.uu.se

September 2022

Scientific Computing
Information Technology
Uppsala University
Box 337
SE-751 05 Uppsala
Sweden

http://www.it.uu.se/

Dissertation for the degree of Licentiate of Philosophy in **Scientific Computing**

ISSN 1404-5117
Printed by the Department of Information Technology, Uppsala University, Sweden

Abstract

The information conveyed by genetic markers such as Single Nucleotide Polymorphisms (SNPs) has been widely used in biomedical research for studying human diseases, but also increasingly in agriculture by plant and animal breeders for selection purposes. Specific identified markers can act as a genetic signature that is correlated to certain characteristics in a living organism, e.g. a sensitivity to a disease or high-yield traits. Capturing these signatures with sufficient statistical power often requires large volumes of data, with thousands of samples to analyze and possibly millions of genetic markers to screen. Establishing statistical significance for effects from genetic variations is especially delicate when they occur at low frequencies.

The production cost of such marker genotype data is therefore a critical part of the analysis. Despite recent technological advances, the production cost can still be prohibitive and genotype imputation strategies have been developed for addressing this issue. The genotype imputation methods have been widely investigated on human data and to a smaller extent on crop and animal species. In the case where only few reference genomes are available for imputation purposes, such as for non-model organisms, the imputation results can be less accurate. Group testing strategies, also called pooling strategies, can be well-suited for complementing imputation in large populations and decreasing the number of genotyping tests required compared to the single testing of every individual. Pooling is especially efficient for genotyping the low-frequency variants. However, because of the particular nature of genotype data and because of the limitations inherent to the genotype testing techniques, decoding pooled genotypes into unique data resolutions is a challenge. Overall, the decoding problem with pooled genotypes can be described as as an inference problem in Missing Not At Random data with nonmonotone missingness patterns.

Specific inference methods such as variations of the Expectation-Maximization algorithm can be used for resolving the pooled

data into estimates of the genotype probabilities for every individual. However, the non-randomness of the undecoded data impacts the outcomes of the inference process. This impact is propagated to imputation if the inferred genotype probabilities are to be devised as input into classical imputation methods for genotypes. In this work, we propose a study of the specific characteristics of a pooling scheme on genotype data, as well as how it affects the results of imputation methods such as tree-based haplotype clustering or coalescent models.

Acknowledgments

I express my sincere thanks to my supervisor Carl Nettelblad for his guidance as well as the always thorough and sharp discussions. The work presented in thesis has been conducted thanks to the funding of Formas (grant No. 2017-00453) and the computing resources provided by SNIC through Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX).I am also grateful to all my TDB colleagues and friends for the moral support, the scientific and intercultural cultural insights as well as precious advice about writing and teaching. Last, beyond the academic environment at Ångström, I would like to thank all "gypsy" scientists expatriated to Uppsala, who are represented to a large extent in the climbing family. You have preserved me from running insane and you have also helped improving my (climbing) problem solving skills a lot!

List of abbreviations

1KGP:	1000 Genomes Project
AAF:	Alternate Allele Frequency
bp:	base pair
DNA:	DeoxyriboNucleic Acid
EM:	Expectation-Maximization
GBS:	Genotyping By Sequencing
GWAS:	Genome-Wide Association Studies
HMM:	Hidden Markov Model
LD :	Linkage Disequilibrium
MAF:	Minor Allele Freqeuncy
MCMC:	Markov Chain Monte Carlo
ML:	Maximum Likelihood
MML:	Maximum Marginal Likelihood
M(C)AR:	Missing (Completely) At Random
MNAR:	Missing Not At Random
NGS:	Next-Generation Sequencing
NGT:	Nonadaptive Group Testing
NORB:	Nonadaptive Overlapping Repeated Block
ONT:	Oxford Nanopore Technologies
PCR:	Polymerase Chain Reaction
QTL:	Quantitative Trait Loci
SNP:	Single Nucleotide Polymorphism
STD:	Shifted Transversal Design
WGS:	Whole Genome Sequencing

List of Papers

This thesis is based on the following papers

I C. Clouard, K. Ausmees, and C. Nettelblad. A Joint Use of Pooling And Imputation For Genotyping SNPs. 2021. doi: 10.21203/rs.3.rs-1131930/v1. `url:https://doi.org/10.21203/rs.3.rs-1131930/v1`. *Under revision.*

II C. Clouard and C. Nettelblad. Consistency study of a reconstructed genotype probability distribution via clustered bootstrapping in NORB pooling blocks. Technical report No. 2022-005 in IT series, June 2022.

Contents

Chapter 1

DNA sequencing and genotyping in the big data era

1.1 Technologies for sequencing the DNA and genotyping markers

Representation of chromosomes, DNA, and nucleotides

The genetic code of the living organisms is encoded within genetic material which consists of DNA molecules. The visible characteristics, or phenotype, of an organism as well as the biological reactions of the metabolism are mostly the result of the expression of the genetic code. The DNA molecule is double-stranded and has remarkable stability properties when replicated, divided and shared through the sexual reproduction, which is the core assumption for parentage and population genetic studies.

One of these characteristic properties is the complementarity of the two strands, that is the nucleotide (or base) adenine (A) is always paired to thymine (T) while the cytosine (C) is paired to the guanine (G). Therefore, the DNA molecule is usually denoted a sequence of pairs of bases (*bp*). The complementary strands are oriented, by convention the reference strand has a *forward* orientation, which corresponds to the orientation of transcriptions of the genes. The *chromosomes* are sequences of DNA that are stored in a compacted form in the nucleus of the cells. In mammalian species, the chromosomes have a length with an order of magnitude of 10^8 bp. In the case of diploid species as human, each chromosome can be paired to its homolog, which shares the same structural features and the same genes at the same loci (genetic positions). The exact sequence of the DNA of each of

the homologous chromosomes depends however on its parental source that can carry different *alleles* at corresponding loci. The combination of alleles is described as a *genotype*. The processes of mutation and recombination that arise in DNA over generations constitute the basis of genetic diversity. This diversity can be studied by describing and measuring the allelic variations between individuals of the same species. Figure 1.1 shows a simplified representation of a pair of human chromosomes and their sequence of nucleotides.

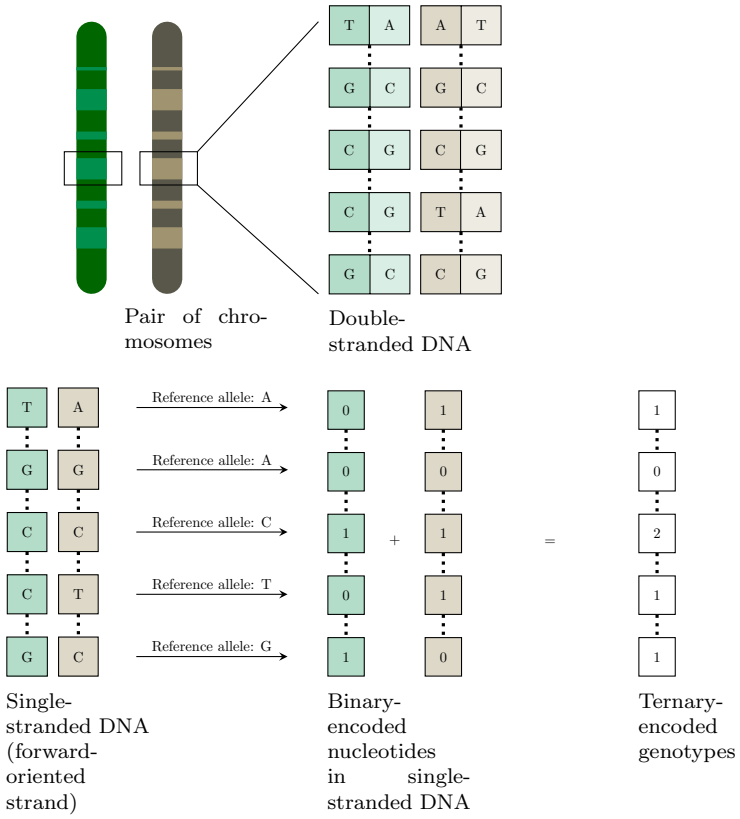


Figure 1.1: **Simplified representation of chromosomes, DNA, and nucleotides for a diploid species.**

Historical evolution of DNA sequencing and successive improvements

DNA sequencing consists in determining the nucleotide sequence of selected parts of the DNA of an organism. The DNA sequencing technologies are

usually classified in successive generations that have been developed since the late 1970's. The first-generation sequencing, originally synonymous to Sanger sequencing, lets the scientists identify the sequences after separating the DNA fragments on polyacrylamide gel [19, 24, 26].

The second-generation DNA sequencing is mostly represented by the Next-Generation Sequencing (NGS) technologies. The NGS machines operate a massive parallelization of the sequencing by using multiplexed schemes for the DNA probes. This type of sequencing has drastically cut the cost of sequencing and facilitated Whole Genome Sequencing (WGS) projects.

Research continues into newer generations of sequencing technologies, with multiple avenues including single molecule sequencing with long reads, real-time sequencing, and nanopore sequencing. To the difference of the two former generations that requires the DNA to be copied e.g. with polymerase chain reaction (PCR) in order to have sufficient quantities, the third-generation sequencing techniques operate without DNA amplification. Therefore, these techniques are expected to further decrease the cost of large-scale DNA sequencing. The third-generation sequencing technologies [54] let the research community hope for achieving ultra-low cost sequencing.

Technologies for genotyping variants of interest in the DNA

Millions of genetic positions in the human DNA have been identified as known positions of genetic variation, which are referred to as variant positions or markers. One category of genetic markers are Single Nucleotide Polymorphisms (SNPs), which means that each of the two chromosomes in a pair might show one of the alleles of a given pair only, depending on the individual. The pair of nucleotides can be any combination from the set A, T, G, C and varies for each locus. The complementary nature of the two DNA strands let the representation of the DNA be simplified as a single strand of nucleotides as shown on Figure 1.1. SNPs are not adjacent to each other in the nucleotidic sequence.

The genotyping technologies are designed to detect which nucleotide is present at these single genetic positions. Each version of the nucleotides is an allele which is identified when calling the genotype Usually, the two alleles of a SNP are arbitrarily typed as *reference* or *alternate* (Figure 1.1). This typing choice is unrelated to the allele frequencies in a population even if it is common to choose the reference allele being the most frequent or the ancestral one. The degree of statistical correlation between the frequencies of the allele arrangements over two SNPs can be expressed as Linkage Disequilibrium (LD). The larger the LD is, the more chances the alleles are inherited together. The resulting series of alleles that derives from the same parent constitute a *haplotype*. From the computational perspective,

SNPs have the advantage that they can be represented as binary entities at the locus, where 0 would denote the reference allele and 1 the alternate allele. Commonly, the genotype of a SNP is represented as a ternary entity representing the total allele count in the locus, with the possible values of $\{0, 1, 2\}$. A locus having twice the same allele, and hence genotype 0 or 2 is said to have a *homozygous* genotype, else (genotype 1) it has a *heterozygous* genotype.

The fluorescent detection of nucleotide on DNA arrays was developped in the late 90's and 2000's. A microarray is a glass plate with one well for each SNP to be genotyped, one individual can therefore be tested for thousands or millions of positions on a single plate. On each chip, there is a collection of short fragments of synthetic DNA probes (less than 100 bp in size) that are complementary to the sequences where the SNP of interest is located. The probes are attached on a chip [24]. After denaturation, amplification and fragmentation of the DNA, the allele discrimination is done by hybridizing nucleotides marked with a fluorescent dye (dNTP*). The SNP alleles are determined based on the color of the fluorescent signal on the chip.

Usually, a sample which is homozygous at the locus of interest returns either a red or a green light signal depending on the allele which is detected. For the heterozygotes, both the reference and the alternate alleles are detected on the chip such that the light signal that is returned combines both red and green fluorescence into a yellow dye. Figure 1.2 [1] illustrate an example of a chip. As for DNA sequencing, the color intensity of the signal measured when reading the array is not linearly proportional to the allelic concentration in the samples. If the fluorescence sensor is correctly calibrated and the DNA sample is not too concentrated, some methods however allow for deducing a quantitative measure of the concentration.

The call rate denotes the proportion of SNPs that can be detected without ambiguity when reading the microarray. Usually, the SNPs with a consequently low call rate are deleted in post-processing before using the genotype data for other analyses such as imputation or Genome-Wide Association Studies (GWAS). Affymetrix and Illumina are today two dominating manufacturers of SNP microarrays.

Each microarray is designed for a selected set of variants. If a reference genome is available, the set of SNPs that are targeted can be positioned with respect to this genome. A reference genome is obtained from a genome assembly experiment which relies on the sequenced data in a cohort of individuals. For instance, in the *1000 Genomes Project* (1KGP), the genomic data was positioned based on the GRCh37 and GRCh38 maps. These maps were created from two different studies of human genome assembly [49, 50]. Traditionally, the SNPs can also be positioned based on an adequate genetic map. What differentiates the sets of SNPs targeted is their density, that is

to say how many SNPs par kbp are genotyped. Nowadays, the manufacturers offer SNP microarrays for variant sets in different species, ranging in size from a few thousands to millions of markers per individual. When genotyping a large population of individuals, there are thousands of millions wells to read and process. In comparison to whole sequence data, the characteristics of the SNP data are their sparse and discontinuous, possibly noisy nature.

Genotyping can also be achieved based on the Next Generation Sequencing (NGS) technologies, in that case it is called Genotyping By Sequencing (GBS). Especially, the NGS technologies have benefited genotyping by considerably reducing the costs since the sequencing can be done in parallel for numerous individuals [25].

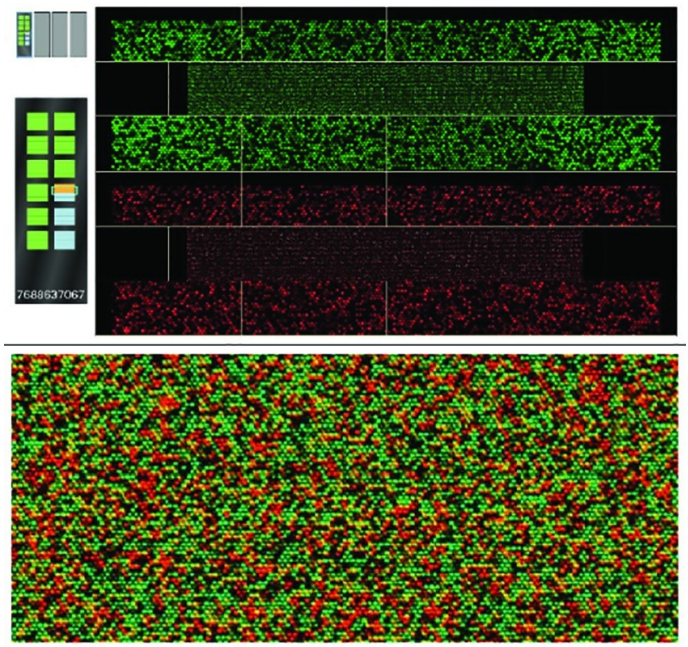


Figure 1.2: **Example of an Illumina BeadChip microarray for SNP genotyping.**

Upper panel: The BeadChip in this example is made of 12 cells on a glass plate and marked with a identifying number at its bottom. Each cell has thousands of micro wells where the DNA probes are fixed. After hybridization, the cells are scanned with red and green fluorometry.

Lower panel: Zoom-in on a cell. The software combines and interprets the results as colors.

1.1.1 Practical applications sequence and genotype data

Mapping phenotypes to genetic profiles

The Human Genome Project was carried out based on sequence data from machines that automated the Sanger technique and led to the sequencing of the first whole human genome in 2003 [19]. The DNA sequencing technologies have had a significant impact on biomedical research as they have enabled large-scale genomic studies by providing a genetic basis for investigating the susceptibility and the heritability to both common and rare human diseases. This kind of research also opens up prospects for future applications as the personal genome project [46] and personalized medicine based on genetic profiles. The biomedical research often makes use of the results of the GWAS. The purpose of these studies is to find statistical relationships between genotypes and phenotypes [5] in a selected population. Often, the phenotype of interest is the status for a disease as diabete. In such case, the results of GWAS enable to relate the risk for developing the disease to a specific genetic profile, which is commonly defined as a polygenic risk score [20].

Beyond the human applications, internationally accessible whole or partial genome sequences data sets are nowadays available for thousands of species. They open the opportunity for comparative genomics studies that aim to capture the molecular mechanisms in organisms [46] and find applications in e.g. environmental or ecological studies.

Agriculture has also benefited from the improvements in sequencing technologies. Genetic data, especially genetic markers as SNPs, serve as support for Marker Assisted Selection in crops [25, 29] and animal farming [2]. Such studies are in particular interested in Quantitative Trait Loci (QTL) and SNPs for genomic selection [13, 25]. Genomic selection aims to use the estimated breeding value for choosing which individuals in an animal or plant population will be used for founding the next generation in a breeding process. The breeding value for the current generation is the predicted relationship between different haplotypes showing specific QTL alleles at some loci that are in LD. The density of markers in the set that is considered affects the estimated breeding value. Overall, the applications using DNA sequence data for domesticated e.g. the mouse *Mus musculus* and wild species e.g. the thale cress *Arabidopsis thaliana* have been less investigated than the human data and there are fewer standard data sets available.

Among the most used and standard genotype maps for human are The HapMap project, the *1000 Genomes Project* and more recent similar ones as the UK Biobank or the SweGen resource [3, 49, 48, 50]. Some of these data sets are created in view of medical applications e.g. the UK Biobank data, which includes rich phenotypic data that is required for calculating

polygenic risk scores [20].

Outlooks and challenges with the sequencing technologies

In conjunction with the improvement of the efficiency of the sequencing technologies, many research fields have increasingly used larger volumes of generated sequence data.

For instance in medical research, the genotyped markers have helped to analyze the heritability and potential genetic causes, or genetic propensity, to develop cancer, diabetes, and heart diseases. The understanding of genetic factors related to diseases opens up prospects towards individualized medicine that is tailored for each patient based on its genetic profile e.g. targeted drug treatments.

Agriculture also benefits from the information that are mined from genomic data. Specifically in breeding and selection programs for crops or livestock [2, 19, 46], the selection of phenotypes with high agricultural value can be accelerated and sharpened based on predictions from the genotype data. In many food crops, phenotypes of interest could be an increased drought or pest resistance, as well as higher yields or improved nutritional value. Genomics-assisted breeding of plant species has for example used GBS technologies [25]. Not only the domesticated organisms but also wildlife monitoring can be boosted with genomic data.

In the case of human data though, privacy concerns have emerged with human genetic testing [30] and the access to personal data e.g. GDPR has reshaped the legal framework.

Regarding the technical aspects, the current sequencing technologies still have limitations for the statistical studies based on the sequenced data. For example, only short reads are produced with NGS technologies [26], which makes the technology inappropriate for genome assembly purposes and for the analysis of genomic variations that could be observed in longer segments. The single positions read from SNP genotyping cannot capture all the structural variants in the genome either, and sometimes fail at capturing the LD [24]. Moreover, the microarray-produced genotype data is also sparse and often noisy [54], which requires specific data (pre)processing as well as adequate statistical treatments. Other single-molecule DNA sequencing such as the Pacific Biosciences (PacBio) sequencing and the Oxford Nanopore Technologies (ONT) are emerging alternatives that produce long-reads with high accuracy [33], and therefore provide better data about the LD information.

1.2 Using group testing for sequencing and genotyping purposes

Despite the progress of sequencing, the production cost of genetic data can still be a concern for projects involving extreme sample counts, especially for non-model organisms, where costs per test tend to be higher. Such issues can for instance arise in agriculture where many animal or plant genomes have not been broadly investigated and sequenced. For these, the data sets currently available may lack sufficient quality, and the creation of new data sets is likely to be more expensive because of the limited concurrence between the retailers. Group testing, or pooling, is relevant in this context of large-scale sequencing and genotyping at low cost.

Beyond the technological effort for parallelizing and automating sequencing and genotyping, pooling represents a supplementary strategy for reducing the cost of large-scale genomic testing [24]. For sequencing purposes, pooling is intended to be used in addition to the NGS technologies in order to further reduce the costs of processing. DNA pooling has been successfully used since the 90's for among others large-scale association studies for human diseases [45] and later for breeding and selection purposes as in rice for pooled genotyping in rice [17] or cattle [2]. However, for low-frequency variants, many individuals need to be genotyped in order to have sufficient statistical power. In the case of pooling for GWAS, often a multivariate regression model is often used for processing the genotyping from pooled samples. These regression models have specific strategies for accommodating with missing or erroneous data.

1.2.1 Categorization of group testing schemes

Principle of pooling

In a broad sense, the general pooling problem consists in identifying a few deviating items in a population in an accurate and efficient way [36]. The samples are mixed together, or pooled, and tested in groups, which reduces the total number of tests performed compared to individual testing. Usually, the tests have a binary outcome e.g. the infection status for a disease. A defective item, for example an infected individual, returns a positive test result, whereas the other items return a negative one. A core assumption underlying pooling is that the test result for a pool is positive as soon as at least one of the items in the pool is positive. For instance, numerous studies have been recently published about practical applications of pooling for population screening and the identification of groups of individuals that were infected with the severe acute respiratory syndrome coronavirus 2 (SARS-

Cov-2) [22, 42]. Whenever the test result for a group was positive, the entire group was considered to be possibly infected, or at least exposed. This type of testing has been mostly used in the US in some schools across the country.

For genotyping purposes, the pooled testing at each SNP consists in calling on a chip the alleles at this locus in a solution where the DNA is mixed from several individuals assigned to the same group. As described previously, the light signal returned by the chip depend on the allelic composition of the DNA mix.

The pooling problem addresses the question of how to form pools in an optimized way. Various objectives can be targeted when solving the optimization problem, such as finding a design that minimizes the number of pools, that limits the pool size, or that accommodates for testing errors [36].

Families and categories of group testing schemes

Two main categories of group testing are common in the literature. The first one is combinatorial group testing which relies on the assumption that the maximum number of defective items in the population to be tested is known and fixed to some integer. The second category is probabilistic group testing where a fixed probability is set for any item to be defective.

When pooling and testing are repeated s times and every new iteration depends on the results of preceding one, the pooling design is said to be s-staged, or adaptive. If the procedure for forming the pools and testing them is specified independently from any other results and for 1 stage only, the pooling design is nonadaptive [36].

Every SNP-chip is manufactured for a predetermined set of SNPs. Hence, using a SNP-chip implies that the genotypes of thousands or millions of SNPs in the targeted set are tested simultaneously. This setting does not allow for adaptive testing of single SNPs or a subset of the SNPs targeted. Therefore, only nonadaptive group testing (NGT) algorithms can be used for SNP genotyping purposes [14, 56].

Strategies for constructing deterministic designs

Various methods that have been investigated for constructing deterministic pooling designs such as pooling-deconvolution, shifted transversal design (STD) and its hypergraph extensions, multiplexed schemes or compressed sensing [14, 16, 23, 24, 36].

In the research we have conducted, we have used the nonadaptive and STD-based approach [51] that constructs overlapping pools and arrange the pools into blocks. The block construction is repeated over the entire popu-

lation such that each sample to be genotyped is assigned to a block. Section 1.3.1 in this chapter provides more details about the terminology of pooling. The designs used for DNA library screening or rare variant frequency estimation are not necessarily overlapping ones [53].

1.2.2 Properties and parameters of deterministic and non-adaptive pooling designs

Definitions and notation

A pooling design defines an algorithm that determines encoding and decoding rules. Assigning the samples to pools corresponds to an encoding step [23]. The pooling problem consists in identifying the deviating items based on the outcomes of the pooled tests. The process of determining the test results for any individual from the pooled outcomes is referred to as the decoding step of the pooling design. For example, both in the STD study [51] and in the DNA Sudoku scheme [55], the decoding step for this design follows pattern-consistency rule.

A NGT design with repeated blocks can be described in a compact way for each block with a design matrix that we denote M. M is a matrix with binary entries of dimensions (T, B), where each of the T rows represents a pool and each of the B columns represents a sample. The entry 1 at coordinates (i, j) indicates that the sample j belongs to the pool i, otherwise the entry is 0. An example of a design matrix, largely based on the STD and the DNA Sudoku, is given in the next section of this chapter.

Let $y = [y_1, \ldots, y_T]$ be the vector indicating the test outcomes for the pools. Likewise, we denote $x = [x_1, \ldots, x_B]$ the vector representing the test outcomes for the individuals. The relationship between the outcomes of the pooled tests in y and the decoded results for every sample represented by x can be modelled as the ceiled result of the multiplication of the design matrix M by the outcome vector y

$$y = \lfloor M \cdot x \rfloor \tag{1.1}$$

Solving the pooling problem consists in finding the vectors x that satisfy 1.1 given the outcomes in the vector y are observed. Typically for this purpose, it is desirable the design matrix M is d-disjunct. This means the design guarantees exact reconstruction of the vector x if there are at most d 'faulty' items in it. d is also called *decoding robustness*.

Performance-critical parameters of the pooling design

Let us define the reduction factor $\rho = \frac{B}{T}$ based on the suggestion in [56] for an under-sampling ratio. Optimizing the pooling design consists in finding a

trade-off between the reduction factor and the decoding robustness. Ideally, both ρ and d are as large as possible. If there are more than d defective samples in the population to be pooled, some items are missing after pooling because they cannot be decoded.

Theoretical studies have explored different methods for constructing designs that are relevant in different contexts. Constructing optimal pooling designs can also lead to time and memory complexity challenges [36]. In our research, we have experimented one design which is detailed in the next section, and studied it only practically for genotyping applications.

1.3 Example of a Nonadaptive Overlapping Repeated Blocks design for SNPs genotyping

This section introduces the simple case of STD that is used in Paper I and II for simulating experiments of pooled genotyping. Given the characteristics of the design, we choose to designate it by the name Nonadaptive Overlapping Repeated Blocks (NORB) pooling design.

1.3.1 NORB parameters and design matrix

A NORB pooling design can be described with the following properties:

- The population to be tested is divided in **blocks** of equal size B [56]. In our experiments, we have chosen $B = 16$.
- That is, if the study population consists of 160 individuals, a block unit is **repeated** 10 times.
- Within each block, we assign the individuals to pools, such that each pool consists of 4 samples and each sample is part of $W = 2$ pools. In other words, there are 2 pools **overlapping** at each sample. Moreover, each of the $T = 8$ pools in the block intersects any of the other pools $\lambda = 1$ time.
- The blocks and the pools are assigned only once for one testing stage, that is the algorithm is **nonadaptive**.

This NORB scheme can be represented by the following design matrix M:

$$
\mathrm{M} = \begin{array}{c} \\ R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \\ R_7 \\ R_8 \end{array}
\begin{array}{c}
\begin{array}{cccccccccccccccc} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \end{array} \\
\left(\begin{array}{cccccccccccccccc}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{array} \right)
\end{array}
$$

With this NORB design, the matrix M is said being d-disjunct if Equation 1.2 involving the maximal value for λ and the minimum one for W is verified:

$$d = \left\lfloor \frac{W_{min} - 1}{\lambda_{max}} \right\rfloor \tag{1.2}$$

The decoding robustness of the design d is defined as the maximum number of carriers of the alternate allele in the block that can be identified with certainty. Our design has a decoding robustness $d = \frac{2-1}{1} = 1$.

In the case of genotype testing, identifying the carriers does not imply that the exact genotype of these items can be resolved as it could be either heterozygous or homozygous for the alternate allele. The reduction factor of the pooling design is $\rho = 2$, which means that half the number of tests are necessary for genotyping the pools compared to doing one test per individual.

1.3.2 Representation of a pooling block

As an alternative to the design matrix, we have used within our studies a more intuitive, graphical representation of pooling blocks. Figure 1.3 shows this block representation as a 4×4 square grid. The $B = 16$ individuals fill the cells of the grid. Each row of the grid consists of 4 samples that belong to the same pool, and likewise for the columns of the grid. That is, there are $T = 4 + 4 = 8$ pools in a block. Each sample intersects one row and one column of the grid, which corresponds to the weight of the design, $W = 2$.

	P_5	P_6	P_7	P_8
P_1	I_1	I_2	I_3	I_4
P_2	I_5	I_6	I_7	I_8
P_3	I_9	I_{10}	I_{11}	I_{12}
P_4	I_{13}	I_{14}	I_{15}	I_{16}

Figure 1.3: **Representation of a NORB pooling block as a square grid.**

1.3.3 Algorithms for encoding and pattern-consistency decoding

While earlier implementations of overlapping pooling schemes [53, 55, 15] were interested in identifying a binary test outcome (carrier of a rare variant or not), our research has focused on a ternary outcome (heterozygote and two kinds of homozygotes).

Algorithm 1 presents pseudocode for determining the pooling outcome of a pool based on the genotype of the samples being part of the pool (encoding step).

Algorithm 1 Genotype encoding with a NORB pooling design

P_{jk} is the genotype of the kth pool at the jth marker
G_{ij} is the genotype of the ith individual at the jth marker
k is the kth pool
for all j **do**
  for all k **do**
    if $\{G_{ij} = 0\}, i \in k$ **then**
      $P_{jk} \leftarrow 0$
    else if $\{G_{ij} = 2\}, i \in k$ **then**
      $P_{jk} \leftarrow 2$
    else
      $P_{jk} \leftarrow 1$
    end if
  end for
end for

As soon as the pools implicated have different genotypes, then the genotype of the individual that is part of these pools cannot be retrieved with certainty. The individual is asserted to be homozygous for the reference allele i.i.f. it participates to at least one reference-homozygous pool, and by symmetry, the sample is decoded as homozygote for the alternate allele i.i.f. at least one of the pools is alternate-homozygote. Similarly to previously, we use the notation $G_{ij} \in \{0, 1, 2, -1\}$, where -1 stands for the case of a missing genotype. The rule for genotypes decoding is described in Algorithm 2. Given the symmetry property for the reference and the alternate alleles, the decoding procedure is similar for the genotypes $\{0, 2\}$.

Algorithm 2 Genotype decoding with a NORB pooling design

P_{ijk} is the genotype of the kth pool at the jth marker in which the individual i participates
G_{ij} is the genotype of the ith individual at the jth marker
k is the kth pool
for all j **do**
  for all i **do**
    if $\{P_{ijk} = 0\}, i \in k$ **then**
      $G_{ij} \leftarrow 0$
    else if $\{P_{ijk} = 2\}, i \in k$ **then**
      $G_{ij} \leftarrow 2$
    else if $\{P_{ijk} \in \{0, 1\}\}, i \in k$ **then**
      $G_{ij} \leftarrow 0$
    else if $\{P_{ijk} \in \{1, 2\}\}, i \in k$ **then**
      $G_{ij} \leftarrow 2$
    else if $\{P_{ijk} = 1\}, i \in k \cap \{P_{ijk} = 0\}, i \notin k$ **then**
      $G_{ij} \leftarrow \{1, 2\}$
    else if $\{P_{ijk} = 1\}, i \in k \cap \{P_{ijk} = 2\}, i \notin k$ **then**
      $G_{ij} \leftarrow \{0, 1\}$
    else
      $G_{ij} \leftarrow -1$
    end if
  end for
end for

If the decoding robustness is exceeded, the rule-based method fails to accurately decode genotypes and fill them in as completely missing (-1 when none of the alleles is assayed) or partially ($\{0, 1\}, \{1, 2\}$ when the presence of one allele is definite but the other allele is indeterminate.

The nested tests in Algorithms 1 and 2 are computationally costly. Depending on the programming language, more efficient alternatives can be

implemented practically, as suggested in [55]. We developed for example a code used in both Paper I and II which performs the decoding and the encoding steps with vector-matrix computations. Encoding in a simulation context as well as decoding for simulated or actual data can be run independently for different genetic markers, making them suitable for parallel execution.

1.4 Overview of the remaining chapters

Chapter 2 details the characteristics of the missing data following NORB pooling. Defining the typology for the missing genotype data determines what methods can be used for inferring the missing genotypes from the combinatorial constraints imposed by the pooling design chosen. We propose a few examples of inference methods that can be implemented for estimating the genotype probabilities in pooled data. Chapter 3 presents two main families of algorithms that can be used for imputing unassayed or missing genotype data. For each family, minimal examples are provided for demonstrating how pooled genotypes impact the imputation model compared to usual data sets. Both Papers I and II are studies implementing a method for estimating the most likely genotypes in pooled data with a NORB design. Paper I is a simulation of a practical application of the use of pooled data followed by genotype imputation. Paper II focuses on the statistical characteristics and the consistency of the data produced by various versions of our method for producing genotype probability estimates during decoding.

Chapter 2

Probabilistic decoding methods of pooled experiments for genotype imputation

This chapter addresses the need of tailored inference methods for estimating genotype data in cases where full decoding is impossible in a NORB pooling design. In that sense, the pooling algorithm can be defined as the missingness mechanism. The specific characteristics of the structure of the missing data in pooled experiments determine which statistical inference methods are suitable, as well as their potential caveats due to the peculiar dependence structure in the data set.

2.1 Structure and characteristics of the missingness in NORB pooled data

2.1.1 Minimal example of a NORB pooling design

For readability, we use hereafter a smaller example than the 4×4 pooling block. The 2×2 minimal example is only intended for illustrative purposes – for one thing its dimensions imply that pooling in this scenario will not reduce the number of tests performed. This pooling design would require $2 + 2$ tests in total for rows and columns, which is equivalent to the number of individuals if they were to be tested separately.

The square block representation for a 2×2 pooling design with the definitions used in Chapter 1 is shown in Figure 2.1.

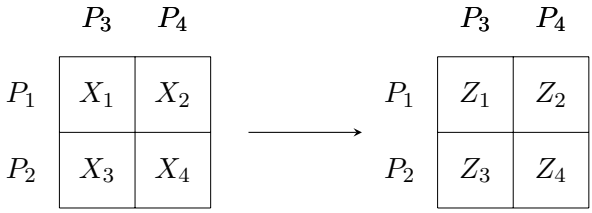


Figure 2.1: X is the true complete genotype data for each individual I. Z is the possibly incomplete data after pooling and decoding.

2.1.2 Graph representation of the pooling algorithm as a missingness mechanism

As described by Mézard et al. [35] for studying missingness in view of missing data inference, Directed Acyclic Graphs (DAGs) can be used for representing a missingness mechanism. In our research, the NORB pooling process can be interpreted as the missingness mechanism. Figure 2.2 shows the DAG representation of the 2×2 NORB pooling design.

2.1.3 Classification of the missingness mechanism

There are three main categories of missingness described in the literature that are defined based on the dependence structure in a data set with missing items. As introduced on Figure 2.2, we use the variable Z for representing the pooled and decoded data which is possibly missing. That is, a realization of Z can generate both observed and unobserved data. The nature of the dependence between the missingness status R_i of any item i in a pooling block and both the other observed and unobserved items in the block let us distinguish the following categories:

- Missing Completely At Random (MCAR): the missingness status is independent of the data, observed or not.
- Missing At Random (MAR): the missingness status depends only on observed data Y
- Missing Not At Random (MNAR): the data is neither MCAR nor MAR.

In this section, we use the 2×2-study case shown in Figure 2.3 for illustrating the dependencies in NORB pooled data. Figure 2.3 shows a heterogeneous example with a mix of all genotypes, where 1 pool (P_4) is tested being homozygous for the reference allele. After decoding, both Z_2

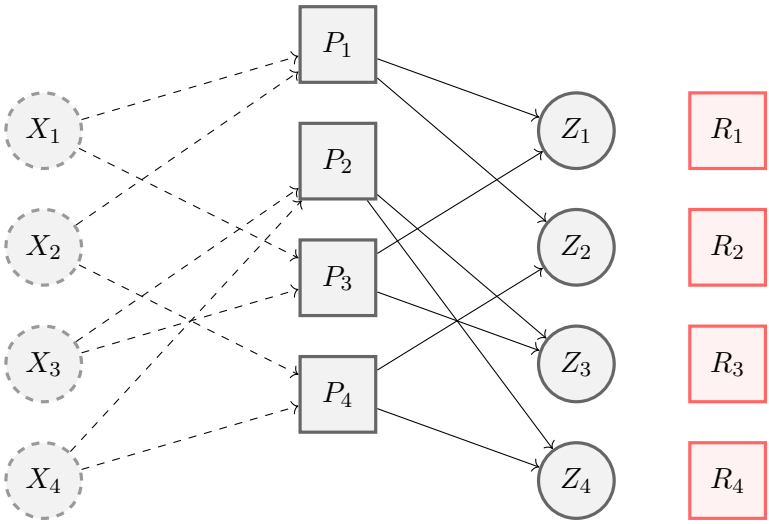


Figure 2.2: DAG representation of a 2×2 NORB pooling design. We use notations similar to the ones proposed by Mézard et al. [35]. The nodes P_i are represented as square nodes since their value corresponds to the direct result of a genotyping test. The variables X_i and Z_i, represented with circle nodes, are individual genotypes in the block. X stands for the true data and the dotted lines represent the fact this data is accessible only in the case of a pooling simulation, otherwise only the data P is known. Z stands for the pooled and decoded data which is possibly missing. R is a variable indicating the missingness status of Z. The edges on the left-hand side of the DAG indicate what samples X_i belong to which pools P_i, this corresponds to the encoding stage of the pooling algorithm. On the right-hand side, the edges connect the pools from which the genotyping results are combined to be decoded into an individual genotype Z_i.

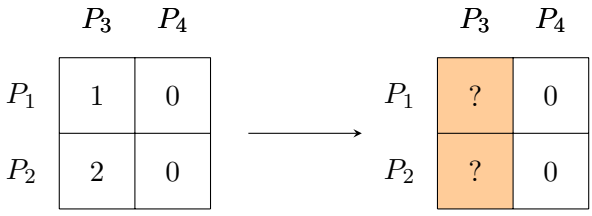


Figure 2.3: The left-most block shows the values for the true data X and the right-most block shows the pooled and decoded values Z.
All three possible genotypes are present among the items X, however by chance the homozygous ones for the reference allele are placed in the same column-pool.
In this example, the pooling pattern is $\psi = ((0, 2, 0), (1, 1, 0))$ as both P_1 and P_2 are tested with genotype 1, as well as P_3 is, and P_4 is homozygous for the reference allele. The individual genotypes Z_1 and Z_3 are missing after pooling.

and Z_4 can therefore be identified as homozygotes for the reference allele, whereas Z_1 and Z_3 are missing.

For any NORB pooling block, we define its pooling pattern

$\psi = (n_{G_{rows}}, n_{G_{columns}})$, where $n_{G_{rows}}$ (resp. $n_{G_{columns}}$) is a triplet of integer values that denote, in this order, the number of row-pools (resp. column-pools) having genotype 0, 1, and 2. For instance, the pooling pattern in Figure 2.3 is $\psi = ((0, 2, 0), (1, 1, 0))$.

Dependency between the missingness status and the observed data

The result of the pooled genotyping test for P_1 impacts the decoded value for both the genotypes Z_1 and Z_2. Therefore, given a particular outcome for Z_2 and the jointly observed result for the pool P_1, some values of Z_1 are inconsistent. For instance, if $Z_2 = 0$ is observed as in Figure 2.3, this constrains $Z_1 \neq 0$. Indeed, if $X_1 = 0 \wedge X_2 = 0$, the pooling algorithm produces the result shown in Figure , that is the observed pooling pattern is $\psi = ((1, 1, 0), (1, 1, 0))$.

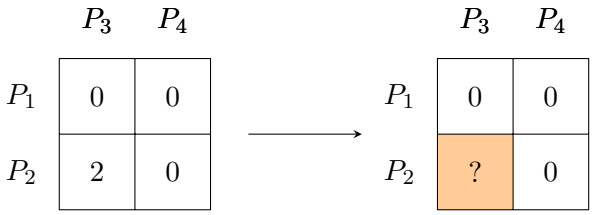


Figure 2.4: The most left block shows the values for the true data X and the most right block shows the pooled and decoded values Z.
Both homozygous genotypes are present among the items X, but no heterozygous ones. By chance, the homozygotes for the reference allele are placed in the same column-pool P_4 and the same row-pool P_1.
In this example, the pooling pattern is $\psi = ((1, 1, 0), (1, 1, 0)) \neq ((0, 2, 0), (1, 1, 0))$ and only the genotype Z_3 is indeterminate.

In other words, the missingness status of Z_1 is conditioned on the observed value of Z_2.

Dependency between the missingness status and the unobserved data

The test result for the pool P_3 affects the decoded values for both Z_1 and Z_3. Hence, the pooling algorithm imposes $Z_3 = 2 \implies Z_1 \neq 2$. Indeed, if $X_1 = 2 \wedge X_3 = 2$, the pooling algorithm produces the result shown in Figure 2.5, that is the observed pooling pattern is $\psi = ((0, 2, 0), (1, 0, 1))$ and the pooling block is fully decoded.

In other words, the missingness status of Z_1 is conditioned on the unobserved value of Z_3.

Figure 2.6 shows the DAG equivalent to Figure 2.3 with the dependencies between the variables highlighted in blue. Through the example of the

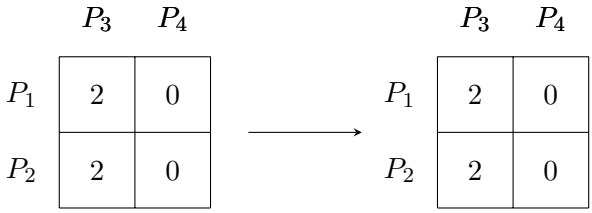


Figure 2.5: The left-most block shows the values for the true data X and the right-most block shows the pooled and decoded values Z.
Only homozygous genotypes 0 and 2 are present among the items X, and by chance the homozygotes for the same allele are placed in the same column-pools P_3 and P_4.
In this example, the pooling pattern is $\psi = ((0,2,0),(1,0,1)) \neq ((0,2,0),(1,1,0))$ and all individual genotypes can be decoded.

relationships between Z_1, Z_2, and Z_3, we reveal that the missingness status of an item in a NORB pooling block depends both on the other observed and unobserved items. In accordance with the definitions given at the beginning of this section, we can therefore characterize the undecoded items in NORB pooling as being missing not at random.

The examples in Figures 2.3, 2.4, and 2.5 illustrate valid and invalid configurations for X given a pooling block pattern. It means that the missing items cannot be reordered in a different order than the one imposed by the pooling design, and therefore follow so called *nonmonotone patterns* [41, 52].

2.2 A tailored inference method for pooled genotype data

We implement custom probabilistic decoding methods for NORB genotype pooling for two main reasons:

- In practical applications to genetics such as in the DNA Sudoku study, primarily a deterministic decoding method was suggested for the scenario of detecting alternate allele carriers within a marker, regardless of zygosity status. For our genotyping purposes, we need to extend the procedure to a decoding method suited for a ternary outcome, namely two opposite heterozygous genotypes and one heterozygous. We have proposed in Paper I and II an EM-based inference method which can be seen as a probabilistic decoding procedure for a NORB poling design.

- The method we propose should be put in the perspective that it is designed to be complemented by a genotype imputation step, as investigated on the *1000 Genomes Project* data set in Paper I. That

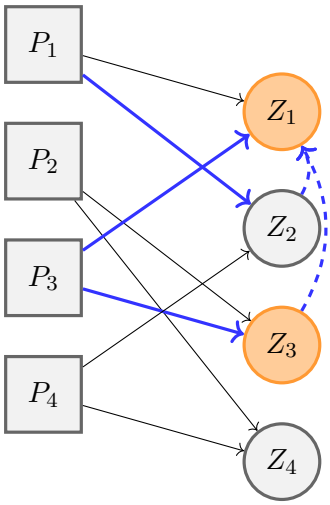


Figure 2.6: Only the decoding step from the pools into individual genotypes is shown, since the behavior during decoding is the focus of this chapter. The variable R is not represented, instead the node Z_i is colored in orange if its value is missing. These choices for the representation are made in order to align the matrix representation of the pooling block and its DAG representation. The plain edges highlighted in blue indicate which items are involved in the decoding algorithm for the missing item Z_1. The dashed blue arrows illustrate that the value of Z_1 is conditioned on both Z_2 and Z_3. The value for Z_1 is obtained from the genotyping result of P_1 and P_3. As X_2 (not represented here) also affect the tests result for P_1, the observed value Z_2 indirectly affects the missingness status of Z_1. Similarly, the value of the missing variable Z_3 indirectly impacts the underlying value of Z_1 as P_3 involves both X_1 and X_3.
This example of pooling block illustrates the MNAR mechanism imposed by NORB pooling as the missingness status of the decoded genotypes depends on observed as well on missing variables participating in the same pools.

means, not only the accuracy of the inferred genotypes from pooling matters, but also to what extent the computed estimates for the genotype probabilities benefit the performance of different imputation methods.

2.2.1 Statistical framework for estimating the missing items in pooled data with a NORB design

As the statistical framework for our research is largely described in Papers I and II, this section presents only briefly some elements of this framework.

Vector notation of the data

Let us model the pooling mechanism as the data mapping t

$$t\colon \mathcal{X} \longrightarrow \mathcal{Z}$$
$$\mathbf{x} \longmapsto \mathbf{z}$$

where $x = (x_1, x_2, ..., x_n)$ and $z = (z_1, z_2, ..., z_n)$ are vectors of genotypes. We are interested in studying possible inversions of the mapping t for estimating the missing values in $\mathbf{z}$ as the most likely genotype probabilities for $\mathbf{x}$.

The true genotype data at any marker for a sample i is represented by a probability simplex $x_i = [p_{0i}, p_{1i}, p_{2i}]^\top$. The three simplex values represent, in order, the probability of the genotype being a homozygote for the reference allele, a heterozygote, and a homozygote for the alternate allele. Similarly, the pooled and decoded genotype data for the same sample i is denoted $z_i = [\tilde{p}_{0i}, \tilde{p}_{1i}, \tilde{p}_{2i}]^\top$.

Formulation of the inference problem in NORB pooled data

The inference problem can be partitioned into a series of maximum likelihood estimation problems conditioned on each missing data pattern, that is to say each pooling block pattern ψ. It is common for likelihood maximization problems for arbitrary complex models to be solved iteratively using Expectation-Maximization (EM) approaches [21].

The empirical likelihood of the complete data in a pooling block is defined as

$$L_\psi(\mathbf{x}) = \prod_{b=1}^{B} x_b \tag{2.1}$$

For any pattern ψ, the objective is to compute point estimates $(\hat{p_{0i}}, \hat{p_{1i}}, \hat{p_{2i}})$ of the genotype probabilities for any item i in the pooling block. The estimation consists in maximizing the empirical likelihood in equation 2.1, that is

$$\forall\, b \in [1, B] \qquad (\hat{p_{0i}}, \hat{p_{1i}}, \hat{p_{2i}}) = \arg\max_{\mathbf{x}} L_\psi(x_b) \tag{2.2}$$

2.2.2 Expectation-Maximization based methods

A detailed description of the EM-based estimation methods we have used can be found in Papers I and II, as well as a few numerical examples.

Therefore, we here only illustrate the general idea for one iteration (m) in Figures 2.7 and 2.8 and highlight some specific features of our EM-based

estimation method for pooled genotype data. Paper II involves several variations of the main steps presented here. On the whole, the various EM versions perform rescaling and marginalization of the expected frequencies of genotypes and/or alleles in different ways.

Expectation step

The expectation step, or E-step, enumerates all data completions of $\mathbf{z}$ for a pooling block having the pattern ψ, as illustrated in Figure 2.7 (I) and 2.8 (II). Some of the enumerated completions might be invalid in the sense that they map to a decoded vector of genotypes which is inconsistent with ψ. Figure 2.7 (II) proposes a few examples of invalid completions for the pooling pattern $\psi = ((2,2,0),(2,2,0))$ in a 4×4 pooling block.

From an algorithmic point of view, the enumeration can be implemented as a dynamic recursion in a ternary tree with n_B levels where each node has a genotype value in $\{0,1,2\}$. The invalid completions correspond to branches in the tree that are pruned, which makes the complexity of the algorithm unpredictable to a certain extent. This dynamic recursive task poses some computational challenges due not only to the size of the search space (the tree might have up to 4.3×10^7 terminating leaves), but also because of the irregular length of the branches depending on the pooling pattern considered. Other strategies can be chosen for the enumeration, such as a Forward-Backward-like algorithm.

At the initial iteration, the genotype probabilities for any item in the block to be decoded can be chosen freely as long as they sum up to 1. For each data completion $\mathbf{x}$ enumerated, its expected proportion $\mathbb{E}[\mathbf{x}|\mathbf{z};\psi]^{(m)}$ is computed as in Equation 2.3.

$$\mathbb{E}[\mathbf{x}|\mathbf{z};\psi]^{(m)} = \frac{Pr(\mathbf{x}|\mathbf{z};\psi)^{(m)}}{\sum_{\mathbf{x}} Pr(\mathbf{x}|\mathbf{z},\psi)^{(m)}} = \frac{Pr(\mathbf{z}|\mathbf{x};\psi)\, Pr(\mathbf{x})^{(m-1)}}{\sum_{\mathbf{x}} Pr(\mathbf{z}|\mathbf{x};\psi)\, Pr(\mathbf{x})^{(m-1)}} \tag{2.3}$$

In the case of invalid data completion, $Pr(\mathbf{z}|\mathbf{x};\psi) = 0$, otherwise $Pr(\mathbf{z}|\mathbf{x};\psi) = 1$.

$Pr(\mathbf{x})^{(m-1)} = \prod_{b=1}^{B} Pr(x_b)^{(m-1)}$ is the probability of $\mathbf{x}$ computed based the individual posterior probabilities at the iteration $(m-1)$.

$\mathbb{E}[\mathbf{x}|\mathbf{z};\psi]$ represent the expected proportion of every valid data completion given that we observe the pattern ψ.

Maximization step

The maximization step, or M-step, calculates for every item in $\mathbf{x}$ the probability of each genotype from the expected frequencies computed at the

E-step:

$$(\tilde{p_0}, \tilde{p_1}, \tilde{p_2})^{(m)} = \frac{\mathbf{x}\,\mathbb{E}[\mathbf{x}|\mathbf{z};\psi]^{(m)}}{\sum_{\mathbf{x}} \mathbf{x}\,\mathbb{E}[\mathbf{x}|\mathbf{z};\psi]^{(m)}} \tag{2.4}$$

where $(\tilde{p_0}, \tilde{p_1}, \tilde{p_2})^{(m)}$ are vectors of estimated genotype frequencies for all samples in the block at iteration m.

Rescaling step

Consecutively to the usual E- and M-step, we implement rescaling operations as follows:

- First, dividing every item in $(\tilde{p_0}, \tilde{p_1}, \tilde{p_2})^{(m)}$ by its individual prior and normalizing the result aims to guarantee the consistency of the method.
- Second, explicitly up-scaling the probabilities for heterozygotes by a factor of 2. This relates to our representation of a single heterozygous state, while there are actually two distinct heterozygote genotypes. Even with a uniform prior, the heterozygous state should thus be twice as common. We refer to this effect as *heterozygotes degeneracy*.
- Once the convergence criterion is met, a final down-scaling of the estimated genotype probabilities is performed. This step is implemented in view of using the genotype probabilities as input in genotype imputation algorithms that internally double the probabilities for the heterozygotes. By down-scaling, we avoid an over-representation of the heterozygous genotypes in imputation.

2.2.3 Quality of the estimates of the genotype probabilities

As seen previously, the NORB pooled data is MNAR, such that the EM-based methods might produce biased estimates [43]. Paper II focuses on the nature and the extent of this bias by investigating the consistency of the EM-reconstructed empirical distribution of the pooled data. We use a divergence criterion for evaluating the consistency of the distribution of the reconstructed data with respect to the true data distribution.

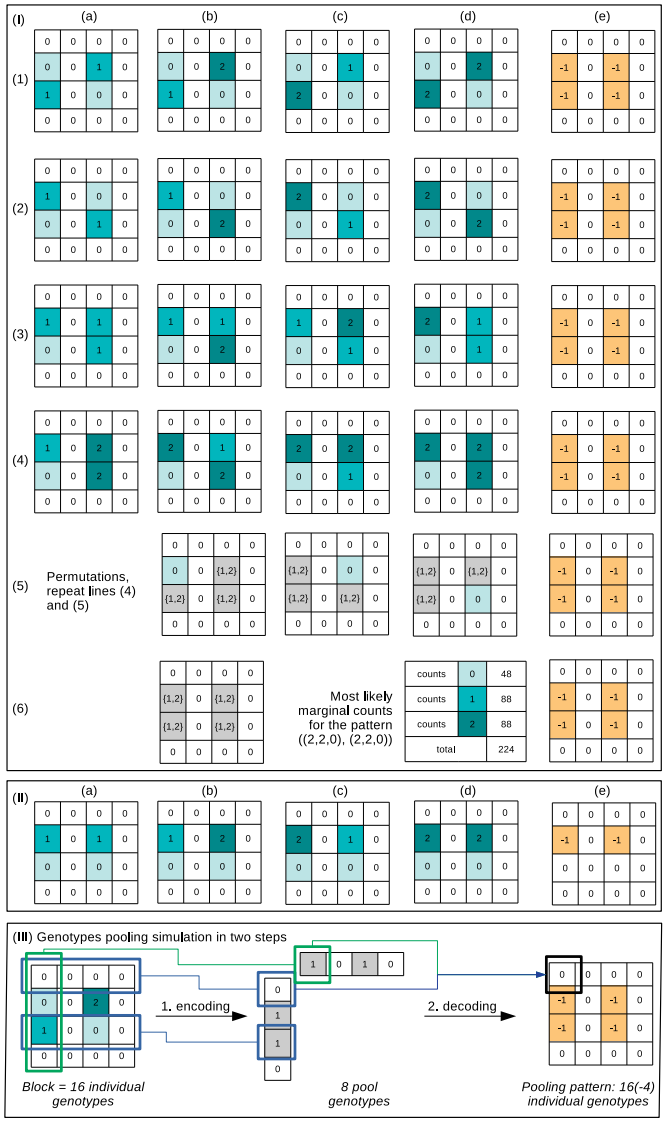


Figure 2.7: **A Maximum-Likelihood-like method for decoding pooling blocks.**

Enumeration example for a 4×4 block with a pattern $\psi = ((2,2,0),(2,2,0))$.
(I) Enumerating the valid layouts compatible with this pattern results in 56 outcomes. Over these combinations, the homozygotes having genotype 0 (resp. the heterozygotes 1 and the opposite homozygotes 2) appears 48 times (resp. 88 and 88), such that the estimated genotypes distribution fitted to the layout is $(0.214, 0.393, 0.393)$. This corresponds to a Maximum Marginal Likelihood estimation.
(II) For a given set of genotypes, some permutations result in a genotype vector that is not compatible with the observed pooling pattern ψ.
(III) Simulating pooling consists in a first encoding step which resolves the genotype of the row- and column-pools: 2 rows have genotype 0, 2 have genotype 1, none has genotype 2, and similarly for the column-pools. The second step decodes the pooled data into individual genotypes. After decoding, the 4 missing items are placed on two distinct pairs of row and column, while the other items are decoded homozygotes (genotype 0).

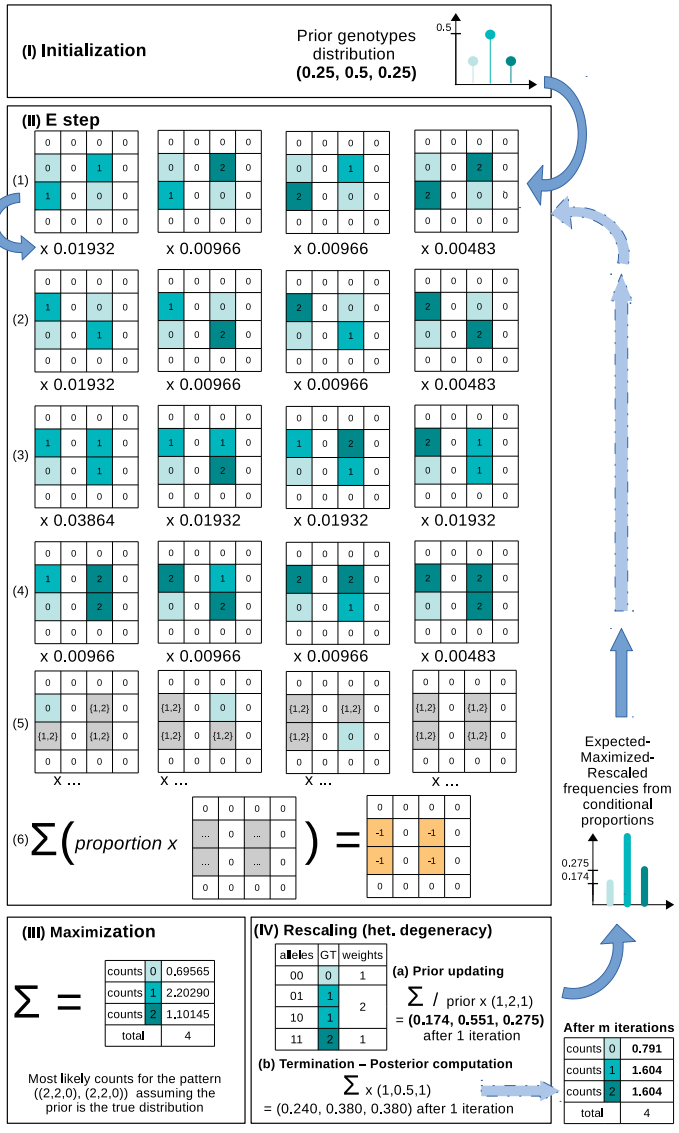


Figure 2.8: **A self-consistent method for estimating ambiguous pooled genotypes with heterozygotes degeneracy.**

The figure shows the same pooling block as in Figure 2.7.
(I) The genotype probabilities for any sample in the pooling block are initialized to a prior value of $(0.25, 0.5, 0.25)$.
(II) The enumeration of the valid data completions is executed in the same way as in Figure 2.7. For each valid completion, the prior genotype probabilities are used to compute the likelihood of the given completion. The likelihood of each completion is later used as a weighing factor.
(III) The most likely genotypes counts are computed based on the likelihood of every valid completion.
The second step decodes the pooled data into individual genotypes. After decoding, the 4 missing items are placed on two distinct pairs of row and column, while the other items are decoded homozygotes (genotype 0).
(IV) Rescaling is applied for accounting for heterozygotes degeneracy and layouts collapsing, as well as a final down-scaling step if the computed estimates are to be used in genotype imputation.

Chapter 3

Statistical genotype imputation for missing markers in large populations

In this chapter, we present statistical computational methods achieving genotype imputation. Imputing the genotype of markers consists in inferring the most likely genotype at these markers when they are missing, based on the known genotype data available for other markers. Data can be missing for different reasons: the DNA material might for example be damaged as with ancient samples, or the genotype data is not reliable after the genotyping technique returned noisy results or of poor quality (e.g. low coverage or low calling rate). Commonly, imputation methods are used for decreasing the cost of large-scale studies based on the genotype data of markers e.g. in GWAS. Given a chosen set of markers of interest in a study population, only part of these markers will be assayed with genotyping techniques. For the remaining part which is unassayed, computational methods are used for imputing the data. The best-performing imputation methods have shown high accuracy, however they usually give less accurate results for rare variants.

In our application, genotype data is missing because of the pooling technique applied. The characteristics of group testing, which were discussed in previous chapters, pose new challenges for current imputation algorithms.

3.1 Introduction

On a general level, the imputation problem can be formulated as resolving ambiguous or unknown genotypes in a study population using probabilistic predictions [28] employing population-wide genetic information. The predictions are derived from different information types available, commonly a

set of densely genotyped and subsequently phased individuals serving as as a reference for estimating the unassayed genotypes of study individuals, and relatedness between the individuals if such data are provided.

We focus on population-based methods, designed for dealing with unrelated individuals. There are also family-based methods including pedigree information in the imputation process, but we have not considered them here. They would be worth investigating in further research work associated to plant breeding, where ancestry and descent information of lines are available over several generations. While other approaches can be found in the literature, we describe only two groups within the population-based methods that have been dominating in the field of genotype imputation [12, 28]. The first group encompasses the coalescence-based models e.g. MaCH and Impute2, as well as an implementation developed locally called *Prophaser* [4], closely modelled on MaCH, but tailored for being used with genotype probabilities as input data. The second group is illustrated by the tree-clustering models e.g. Beagle. Both approaches are iterative and they have been reported [12, 28] among the best performing ones. MaCH and Impute2, as well as Beagle, have been essentially designed for solving the imputation problem in populations where the genotype data is fully missing for all individuals at some markers, that is to say the genotypes are missing completely at random. In Paper I, we showed that *Prophaser* performs well on pooled data whereas the performance of the Beagle model was negatively affected by pooling [18].

3.1.1 Definitions and notations

Let us denote Θ being a set of n_h template haplotypes at n_j loci. Depending on the imputation strategy used, Θ is built upon the reference panel and/or the n_i individuals from the study population. For each of the n_i individuals of the study population, H_{ij} $j \in [1, n_j]$ is a pair of haplotypes at marker j for the i-th individual and we denote $\mathbf{H}_i$ the sequence of n_j haplotypes over all markers. Similarly, $G_{i,j}$ is the genotype (pair of alleles) at marker j for the i-th individual and $\mathbf{G}_i$ the sequence of n_j genotypes. In other words, any study sample i is modelled as a sequence of either haplotype states or genotypes.

3.1.2 Mathematical formulation of the imputation problem

Both population-based models are iterative statistical methods that yield probabilistic predictions of the genotypes for the missing marker data. For each marker imputed at locus j for the individual i, the probabilistic predictions calculated for the genotype can be formulated as in Equation 3.1.

Commonly, this prediction is discretized as the *best-guess genotype* value [28] and formatted as GT. The GT value is the genotype having the highest probability in the predicted probability tuple. The GT format can be a *phased* genotype if each of the alleles is attributed to the parental haplotype (maternal or paternal haplotype), else the genotype is said to be *unphased*.

$$p_{ija} = Pr(G_{ij} = a|\Theta, \mathbf{G}_i),\ a \in \{0, 1, 2\},\ \sum_a p_{ija} = 1 \tag{3.1}$$

That is, the genotype of any marker at locus j for the individual i is conditioned both on the other haplotypes Θ in the population that are used as templates, and on the genotypes observed at the other loci in the sequence $\mathbf{G}_i$.

3.1.3 Hidden Markov Models for modelling haplotypes and sequences of genotypes

Both coalescent and tree-clustering methods implement Hidden Markov Models (HMMs). A graphical representation of a generic HMM for genotype imputation is given in Figure 3.1.

Using a notation consistent with the classical HMM treatment by Rabiner [40], the HMMs used in imputation models can be characterized as follows:

1. The number of states n_h in the model equals the number of template haplotypes, or the number of pairs of template haplotypes. The hidden state i is denoted s_i in Figure 3.1.

2. There are 2 distinct observation symbols per haplotype i at the locus j, one for each allele e.g. $G_{ij} = 0$ or $G_{ij} = 1$ in Figure 3.1. If the hidden state is a pair of haplotypes, there are 2^2 observation symbols for each hidden state, each of them corresponding to a phased genotype.

3. The transition probability distribution from state s_{i1} to s_{i2} is $\mathcal{F} = \{f_{s_{i1},s_{i2}}\}$. $\mathcal{F}$ is either explicitly parametrized with a recombination rate ρ as in coalescent models, or implicitly captured through the counts of haplotype clusters when building the tree in the Beagle model. The transition from one haplotype state to another between two consecutive markers mimics a historical recombination event. It is correlated to the LD between markers.

4. The probability of emitting the symbol a_j from the state s_i is $\mathcal{G} = \{g_{s_i}(a_j)\}$. The observed genotypes model possibly erroneous copies of the haplotypes and hence express mutation events. These events are explicitly parametrized in coalescent models with the mutation rate μ.

5. The initial states are determined or randomly assigned based on the observed haplotypes in the population to impute (see examples in the Sections 3.2.3 and 3.3.2). We denote $\mathcal{S}$ the initial distribution of the states.

A HMM model designed for genotype imputation is typically used for solving three problems for any study sample i [32], which are:

Problem 1 Given a sequence of observations e.g. a sequence of genotypes $\mathbf{G}_i$ and the model parameters $(\mathcal{F}, \mathcal{G}, \mathcal{S})$, the HMM lets one compute the probability of the sequence $Pr(\mathbf{G}_i|\mathcal{F}, \mathcal{G}, \mathcal{S})$. The computation is executed with the Forward-Backward algorithm which sums the probabilities of observing $\mathbf{G}_i$ over all possible sequences of hidden states e.g. haplotypes.

Problem 2 Given a sequence of genotypes $\mathbf{G}_i$ and the model parameters $(\mathcal{F}, \mathcal{G}, \mathcal{S})$, the Viterbi algorithm determines the most likely sequence of haplotypes $\mathbf{H}_i$ from which $\mathbf{G}_i$ derives.

Problem 3 The Baum-Welch algorithm adjusts the model parameters $(\mathcal{F}, \mathcal{G}, \mathcal{S})$ such that $Pr(\mathbf{G}_i|\mathcal{F}, \mathcal{G}, \mathcal{S})$ is maximized.

In coalescent models, the most likely sequence of genotypes for each study individual is computed iteratively by using the three algorithms listed above at each iteration. The Impute2 model however uses fixed model parameters and hence simplifies the **Problem 3**, whereas MaCH reevaluates the model parameters at each iteration. *Prophaser* as used in Paper I executes only one iteration of the MaCH model.

Tree-based clustering, in the form implemented in Beagle, is an empirical model determined by the counts of similar segments found across the template haplotypes.

For both the coalescent and the tree-based models, the hidden states underlying the Markov chain of the HMM are defined by single or aggregated template haplotypes. The strategy for choosing the template haplotypes used as hidden states notably differs between the coalescent and the tree-clustering approaches. An illustrated example is given later in this chapter (Sections 3.2.3 and 3.3.2).

3.2 Coalescent models

3.2.1 The coalescence principle

The coalescent models in this family rely on the so called principle of *coalescence* [27, 32] which asserts that the haplotypes in a homogeneous population tend to be similar. The variations found between the haplotypes

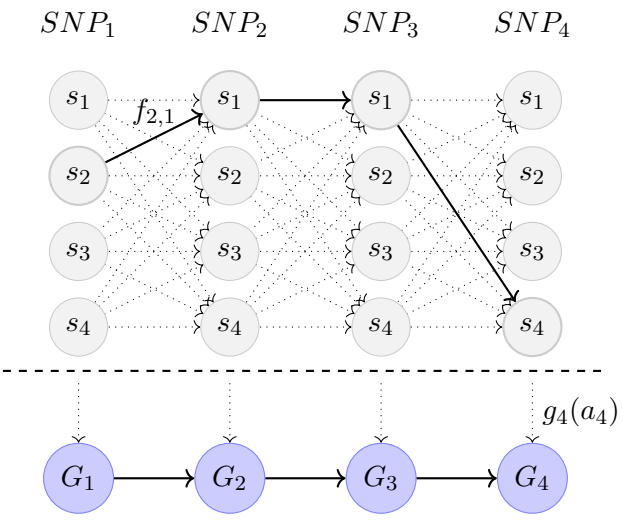


Figure 3.1: **Trellis of the observation sequence (G_1, G_2, G_3, G_4) for an HMM with 4 states.** The thick arrows indicate the most probable transitions. Each state s_i represent a template haplotype. $f_{s_{i1},s_{i2}}$ is the probability to transition from the hidden state s_{i_1} to the hidden state s_{i_2} which depends on the linkage disequilibrium (LD) between the two successive loci. $g_{s_i}(a_j)$ is the probability to emit the symbol a_j from the state s_i. In the case of coalescent models, the recombination and the mutation probabilities are modelled with the explicit parameters ρ and μ. The most likely sequence of states (s_2, s_1, s_1, s_4) can be seen as a mosaic of template haplotypes.

are explained through a combination of the genetic events of recombination and mutation over time. These events are assumed to be rare over small genetic distances and limited time-spans, there are therefore great similarities between haplotypes in different individuals within a population [12, 47]. MaCH and Impute2 exploit the linkage disequilibrium (LD) between markers for capturing the genetic patterns across haplotypes.

3.2.2 Specific aspects of the coalesent models

For each sample i of the study population, the coalescent models computes the probability of the observation sequence $\mathbf{G}_i$ based on Equation 3.2. Impute2 and MaCH proceed by sampling sequences of states through the trellis of haplotypes as in Figure 3.1.

The hidden states underlying the Markov chain of the HMM are the haplotypes (single haplotypes or haplotype pairs depending on the model) which are selected from a set of template haplotypes. The way this set of template haplotypes is constituted varies with the imputation method used.

$$Pr(\mathbf{G}_i|\Theta,\mu,\rho) = \sum_{\mathbf{H}} Pr(\mathbf{G}_i|\mathbf{H},\mu) \cdot Pr(\mathbf{H}|\Theta,\rho) \tag{3.2}$$

The factor $Pr(\mathbf{G}_i|\mathbf{H},\mu)$ models mutation (μ is an explicit parameter or not depending on the model type) along the Markov chain of hidden states, and the factor $Pr(\mathbf{H}|\Theta,\rho)$ models recombination (whether ρ is explicit or not) from a hidden states to its emitted symbol at a marker.

Impute2 uses fixed probabilities of recombination events that are provided in a fine-scale recombination map as LD values between the markers. These values depend on the physical distance between the markers [31, 34, 47]. The distance between the markers is provided in the form of a genetic map that is calculated from a genome assembly.

MaCH reevaluates the recombination and mutation probabilities at each iteration once all the study samples have been processed, based on the last sampled sequences of haplotypes.

Selection of the template haplotypes

Impute2 selects the template haplotypes from the reference panel and the study population based on similarity to the individual being phased ('informed selection' of conditioning states) [27, 28]. MaCH randomly selects a subset from the reference and the study population [32]. The subsetting strategy maximizes the use of available information while limiting the size of the state space in the Markov chain.

Haplotype phasing

Impute2 executes at every iteration two steps that are haplotype phasing and actual genotype imputation [34]. In the HMM used for phasing, the transition probabilities are the probabilities that the hidden state switches between two consecutive assayed markers (observed genotypes). At the first iteration, the haplotypes in the study population are randomly phased and the initial transition probabilities are equal for all hidden states. Phasing the haplotypes of every study sample is executed in the Impute2 model sampling the most likely state path in a Markov Chain Monte Carlo (MCMC) scheme. The resulting path can be seen as a compound of template haplotypes, therefore the expression "mosaic of haplotypes" is frequently employed [28, 32, 38].

The MaCH model performs path sampling as well but proceeds backwards, which is different from the regular Viterbi algorithm. At every locus, MaCH uses the forward probabilities of the possible paths through the templates weighted by the likelihood of the current estimate for sampling an updated sequence of haplotypes. This technique adjusts the likelihood of the sampled sequence locally at each marker without recomputing the likelihood of the entire path. The final sequence of n_j haplotypes sampled for each study individual is used in its turn as one of the templates in the processing of additional individuals.

Genotype imputation

In the Impute2 model, the genotype imputation step reuses the results of the computations in the phasing step for computing the marginal probability of each genotype $0, 1, 2$ for any missing item. The model assumes that the phased haplotypes were sampled from a population that conforms to Hardy-Weinberg Equilibrium (HWE). The genotype probabilities are derived from the allelic probabilities [27] in the entire population.

MaCH does not directly compute the genotype probabilities at each marker, but the path sampling for each individual is performed in a way such that the sequences of haplotypes are edited consistently with the observed genotypes (**Problem 3**). The genotype probabilities at missing markers are deduced after the last iteration from the counts of sampled genotypes over all iterations.

Complexity and computational performance

Impute2 and MaCH form the HMM hidden states by selecting n_h template haplotypes in both the reference and the study population, such there is a constant number n_h^2 hidden states at each of the n_j diploid markers.

Thanks to a memory-saving technique implemented in the forward-backward algorithm, both methods have a memory complexity $\mathcal{O}(\sqrt{n_j})$ for each individual. The time complexity grows linearly as the size of the study population and quadratically with the number of template haplotypes [27]. Nonetheless, several papers point out computational time issues with MaCH [12, 28, 37] when compared to the other methods mentioned. One reason is the reevaluation of the crossover and the mutation rate parameters after each iteration.

By contrast, Beagle operates a dimension reduction of the hidden states space thanks to its clustering approach, which has been shown to be particularly efficient when imputing large data sets. The successive releases for Beagle have improved the software performance in this direction [7, 8, 9, 12, 11].

3.2.3 Minimal examples of phasing and imputation in randomly missing and pooled genotype data

The illustrations presented in this section are based on the example used by Howie and Marchini [28, 34].

The reference panel consists of phased haplotypes from individuals. Each haplotype is a sequence of alleles at the markers of interest, inherited from the mother or the father.

The study sample consists of genotypes with sparse data at the same markers, where the haplotypes are unphased. Let us define two marker sets as follows:

- The set of markers $\mathcal{T}$ which consists of markers for which the genotypes are known in both the reference panel and the study population,
- The set of markers $\mathcal{U}$ which consists of markers for which the genotypes are assayed in the reference panel only and missing in the study population.

Figures 3.2 and 3.4 illustrate the definitions for the haplotypes and the marker sets. We consider the examples of two different study populations at the same loci:

1. A study population where the genotype data is missing fully at random. Whenever a marker is assayed, the genotypes are known for all samples, and conversely when a marker is unassayed, the genotype data is missing for all study samples (M(C)AR data).
2. A study population where the genotype data is missing due to a NORB pooling process. The markers are likely to be missing for only some samples, or entirely missing at common variants (MNAR data).

The HMM employed for phasing uses haplotypes from the reference as panel as well as those currently in $\mathcal{T}$ as templates. For simplicity, our figures will show template haplotypes chosen exclusively from the reference population. Figure 3.3 shows the phased haplotypes of three study samples after one iteration of the phasing-imputation algorithm with M(C)AR data, as well as the resulting imputed genotype for one sample.

If prior genotype probabilities are provided for any missing genotype, they are specified with the factor $Pr(\mathbf{G_i}|H, G, \mu)$ in Equation 3.2. The prior genotype probabilities affect the phasing step and the resulting mosaic of haplotypes. In Paper I, we have investigated how pattern-adaptive estimates of the genotype probabilities in pooled data can improve the accuracy of the phasing step and consequently benefit genotype imputation.

3.3 Tree-based haplotype clusters models

Beagle is the software illustrating tree-based haplotype clustering. It has been developed and improved by Browning and Browning since 2006 [7, 6, 8, 9, 10, 11, 12]. The different versions of Beagle have shown competitive accuracy and computational performance in various settings, including very large data sets. The software has been tested on human [28] as well as on animal and crop species genomic data [39]. Thanks to the clustering approach that reduces the state space dimension, Beagle has been shown being particularly efficient on large data sets and the successive releases have improved the method performance in this direction. Browning and Browning have adopted an alternative approach to coalescence for exploiting sequence variations that feature a given genetic structure in a population.

3.3.1 Specific aspects of the Beagle model

Construction of the template haplotypes

At each iteration of Beagle, the algorithm includes a preliminary model-building step which uses all haplotypes available in the reference panel and the study population. More recent versions of Beagle implement an iterative weighing of the reference vs. the study haplotypes, such that the reference panel affects the model building more at the first iterations [7]. The model-building step consists in fitting a HMM with n_j levels to the observed haplotype data. The levels correspond to an ordered sequence of n_j markers. The resulting model that is built can be described as a Variable-Length Markov Chain (VLMC) where the number of template haplotypes that condition phasing and imputation varies at each marker. This feature is a notable difference to the coalescent models where the number of template

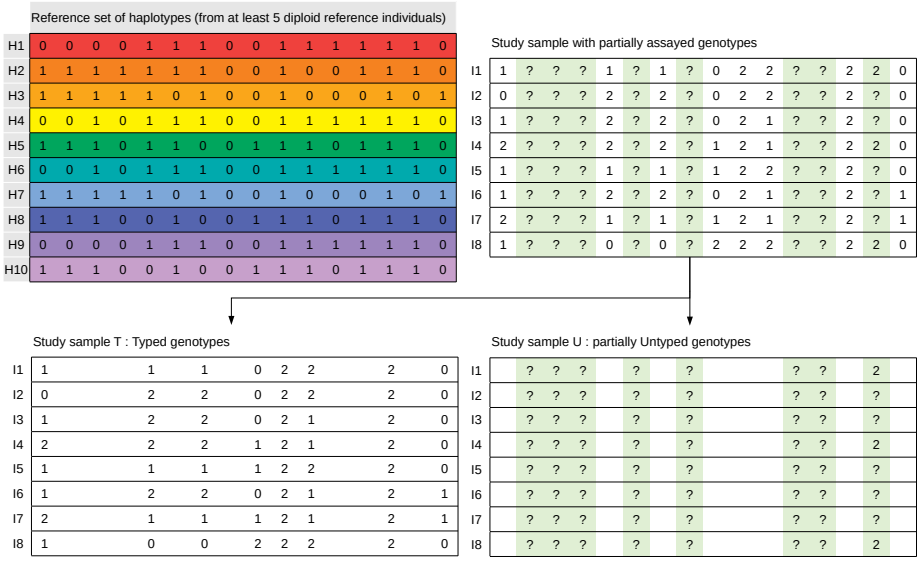


Figure 3.2: **Data sets involved in phasing and imputation for a coalescent model with M(C)AR data.** The data sets consist of a 10 haplotype reference panel and an 8 individual study population. The marker genotypes at 16 loci are represented as integers being the sum of their two alleles for readability. In this particular examples, the template haplotypes come from the reference panel only. The study population is split into a set $\mathcal{T}$ with assayed genotypes and the complementary set $\mathcal{U}$ with unassayed genotypes. With M(C)AR genotype data, the markers are either fully missing for all study individuals, or fully assayed.

haplotypes used is constant along the sequence of markers.

A minimal example is provided in the Section 3.3.2 of this chapter.

At each level j of the tree, the child nodes at level $j+1$ are derived by splitting the observed haplotypes according to their alleles at the current marker. For biallelic markers, any node will have up to two children, depending on what alleles are actually present in the considered haplotypes at marker $j+1$. The tree is extended at each locus such that two loci are connected by an edge.

After processing the last marker at locus n_j, the edges of the tree are weighted by the number of observed haplotypes passing through them. Every template haplotype initially has a unit weight [7]. At the very first

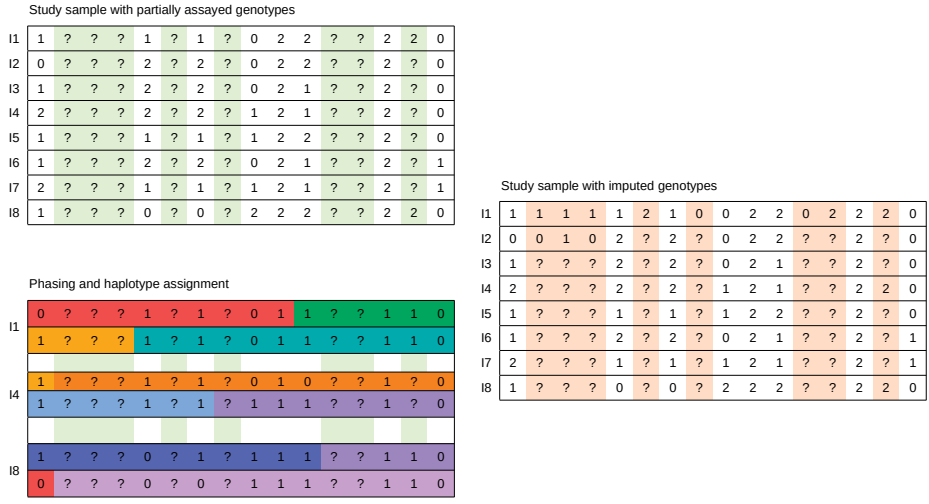


Figure 3.3: **Mosaic of haplotypes from phasing and imputation for a coalescent model with M(C)AR data.** The phasing step computes the most likely pair of mosaic haplotypes for any study sample, based on the template haplotypes (only the reference panel here) and the assayed genotypes. At each locus, the likelihood of every possible pair of haplotypes is computed, which results in $n_h^2 \times n_j = 10^2 \times 16 = 1600$ operations for every study sample. The missing genotypes are imputed as the most likely emitted symbol from the phased haplotypes. The imputed genotypes are represented as the sum of the alleles that are carried by the two haplotypes at the locus e.g. I_1 has genotype $1 = 0 + 1$ for the three first imputed markers since the red segment of haplotype carries the allele 0 at these loci, and the orange haplotype carries the allele 1.

iteration, all haplotypes available in the reference panel and in the study population are used. The nodes are merged at every level of the tree accordingly to a threshold computed from downstream haplotypes frequencies [6, 10, 11].

The merging process results in haplotypes that are clustered based on a frequency criterion of the allele sequences [7]. Node mergers will occur depending on the linkage disequilibrium between successive loci, such that the number of nodes locally increases with the linkage disequilibrium [12].

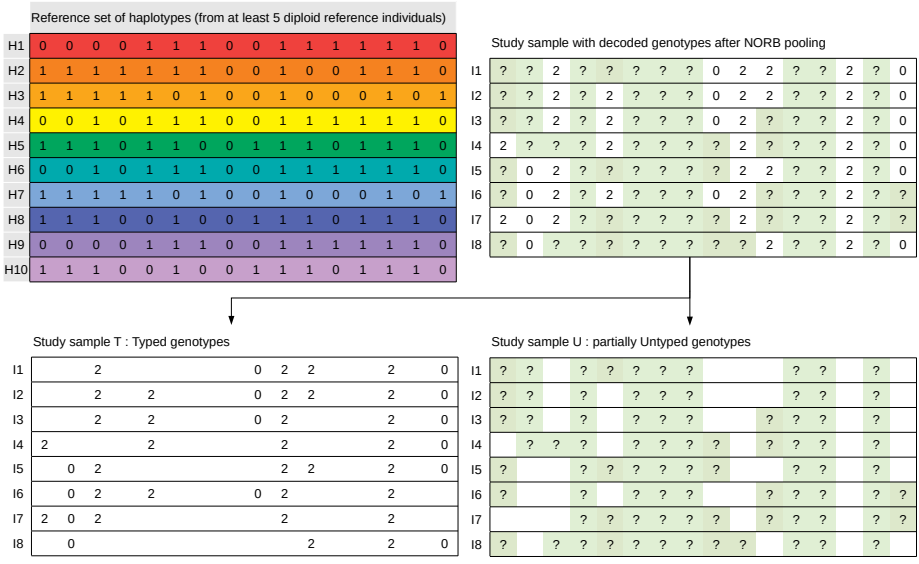


Figure 3.4: **Data sets involved in phasing and imputation for a coalescent model with MNAR data.** The data sets are the reference panel of 10 haplotypes and study population of 8 individuals. The marker genotypes at 16 loci are represented as integers being the sum of their two alleles for readability. In this particular examples, the template haplotypes come from the reference panel only. The study population is decoded from a pooled genotype testing, with the same split as used in Figure 3.2. With MNAR genotype data, some markers can be missing for only a subset of individuals, partially dependent on their actual genotype, something that affects the results of phasing imputation.

Haplotype phasing and genotype imputation

Beagle can perform both phasing and imputation simultaneously, but phasing can also be done beforehand, either with Beagle or with other software e.g. Impute2. For each target individual, phasing is done by sampling the most likely haplotypes with the Viterbi algorithm from the clustered tree, conditioned on the observed genotypes. Missing alleles are randomly imputed according to the observed allele frequencies, which are themselves derived from the haplotype estimates. The newly sampled haplotypes are used as estimates in the next iteration of the algorithm for updating the

haplotype tree. The genotype probabilities at each locus are eventually computed from the last estimated tree. By applying merging, weighting and pruning in the successive trees, Beagle captures the population-specific diversity through the haplotype patterns, without explicitly modeling recombination or mutation events as sources of genetic variation [12].

Complexity and computational performance

The number of template haplotypes obtained with clustering is less than the initial number of haplotypes in the reference panel and the study population. Therefore, the size of the state space of the HMM used for imputation is decreased, which is a key factor of the computational efficiency of Beagle in terms of memory as well as time consumption.

3.3.2 Minimal examples of a leveled HMM from M(C)AR and MNAR data

The reference panel and the study population are identical to the examples previously shown for the coalescent models in order to facilitate the comparisons between these two families of imputation models. Figures 3.5 and 3.6 show the initiation of the model building step in the case of imputation of M(C)AR data, based on the research of Browning and Browning [7, 10]. Figures 3.7 and 3.8 contain the corresponding illustrations for the MNAR case of decoding pooled NORB data.

Genotype data missing fulling at random: M(C)AR data

This example corresponds to the classical imputation scenario studied in Paper I.

After sampling alleles at unknown markers and randomly phasing the genotypes, the reference panel and the study population would correspond to the state shown in Figure 3.5. The tree shown in Figure 3.6 are derived from the counts presented in Table 3.1.

In practice, the number of haplotypes to use should be much larger (several hundred) such that the clustering model has sufficient statistical power, but this number is kept small for the sake of the example.

Genotype data missing not at random: pooled data

This example corresponds to the joint pooling and imputation scenario studied in Paper I.

After sampling alleles at unknown markers and randomly phasing the genotypes, the reference panel and the study population would correspond

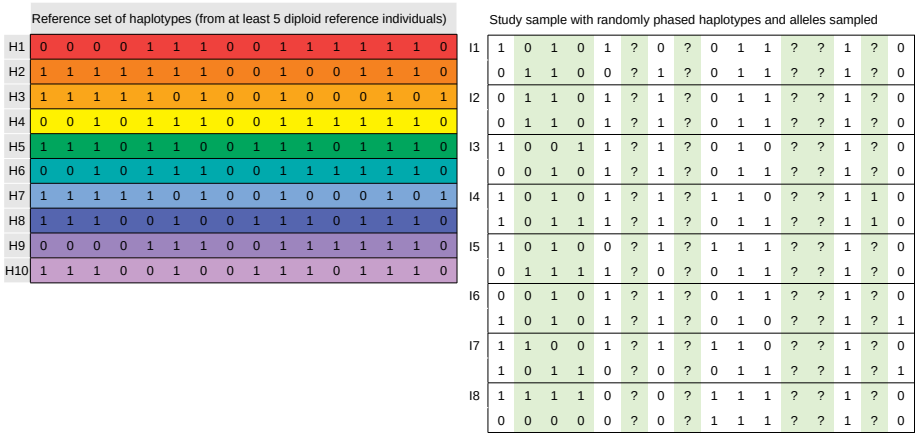


Figure 3.5:

Example of initiation of the VLMC with sparse M(C)AR data.

The unassayed genotypes in the study population to be imputed were randomly phased and the alleles chosen proportionally to the observed allele frequency at each marker. For instance, at the second marker (2nd column of the reference panel and the study population), the genotypes are fully unassayed. The observed frequency of the allele 0 is the one observed in the reference panel only, which is equal to $\frac{4}{10} = 0.4$ (0.6). In the study population, the 16 unknown alleles are randomly assigned in these proportions, that is to say $16 \times 0.4 \sim 6$ haplotypes carry the allele 0.

to the state shown in Figure 3.7. The tree representation is shown in Figure 3.8, based on the counts in Table 3.2.

3.4 Conclusion

In this chapter, the illustrated examples with coalescent models in Section 3.2.3 and the Beagle model in Section 3.3.2 reveal the impact of pooling on haplotype phasing and genotype imputation. Pooling notably modifies the frequencies of observed genotypes from which the template haplotypes are determined, which in its turn affects the sampling operations performed in the HMMs. How much the pooled genotype frequencies differ from the true

Haplotype	Count
0000	3
0001	0
0010	4
0011	0
0100	0
0101	0
0110	3
0111	1
1000	0
1001	1
1010	4
1011	2
1100	1
1101	0
1110	3
1111	4

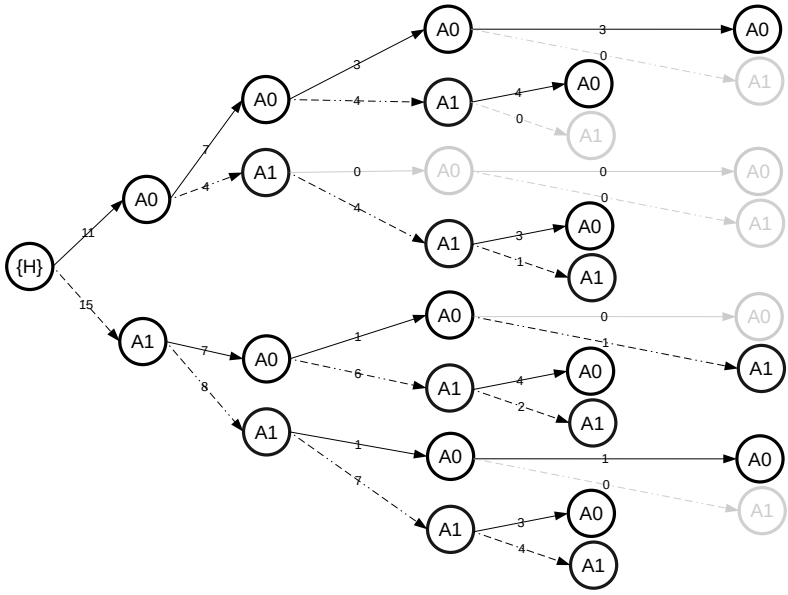
Haplotype counts in M(C)AR data.

Table 3.1: The counts are obtained after completing the missing data based on the observed allele frequencies at each marker.

Haplotype	Count
0000	2
0001	0
0010	3
0011	1
0100	0
0101	0
0110	0
0111	1
1000	0
1001	0
1010	9
1011	3
1100	0
1101	0
1110	4
1111	3

Haplotype counts in MNAR data.

Table 3.2: The counts are obtained after completing the missing data based on the observed allele frequencies at each marker.



Example of VLMC with haplotypes from sparse M(C)AR data.

Figure 3.6: The tree is formed from haplotype counts for the 4 first markers in Figure 3.5. The root of the tree $\{H\}$ is not a marker. $A0$ represent the allele 0 and $A1$ the allele 1. Grey nodes and branches indicate that the allele sequences that were not observed in the available set of haplotypes.

ones depends the allele frequency at the markers. This relationship is not linear but is related to the hypergeometric distribution [14]. If the HWE holds for the genotype frequencies in the population, the pooled genotype frequencies will deviate from this equilibrium, which may deteriorate the imputation accuracy of the models relying on the HWE assumption. In the case of Beagle, the number of haplotypes and the length of the considered marker sequences in the examples are too small for fully demonstrating the effect of pooling on the node merging step. Nevertheless, the trees presented in Figures 3.6 and 3.8 let us notice clear variations in the counts of haplotypes. As a consequence, the resulting VLMC which define the template haplotypes have very different structures and the impact of this modification on the convergence of the imputation algorithm has not been studied to our knowledge. Paper I investigates more thoroughly the consequences of pooling on genotype imputation by comparing the imputation accuracy in two scenarios, one of which corresponding to a M(C)AR case and the other

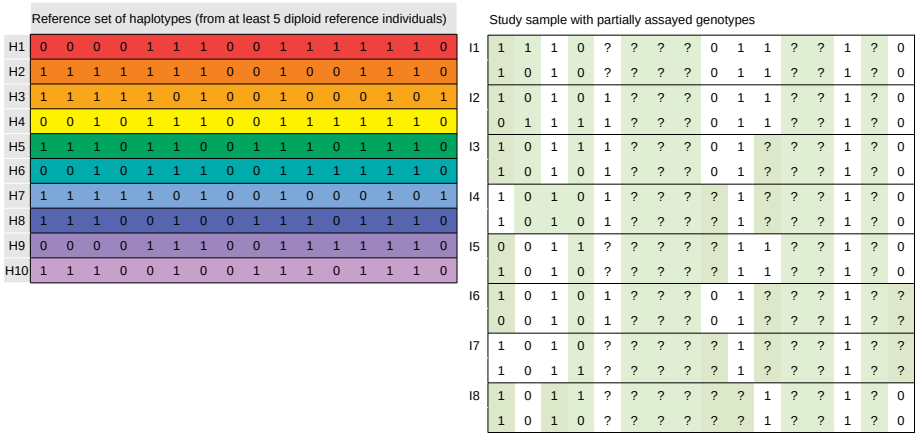


Figure 3.7:

Example of initiation of the VLMC with sparse MNAR data.

The unassayed genotypes in the study population to be imputed were randomly phased and the alleles chosen proportionally to the observed allele frequency at each marker. For instance, at the second marker (2nd column of the reference panel and the study population), the genotypes are partially unassayed. The observed frequency of the allele 0 is the one observed in the reference panel and for 8 haplotypes from the study population, which is equal to $\frac{12}{18} \sim 0.7$. In the study population, the 16 unknown alleles are randomly assigned in these proportions, that is to say $8 \times 0.7 \sim 6$ haplotypes carry the allele 0. As a result, the allelic proportions are notably different relative the ones in Figure 3.6.

one to a MNAR case.

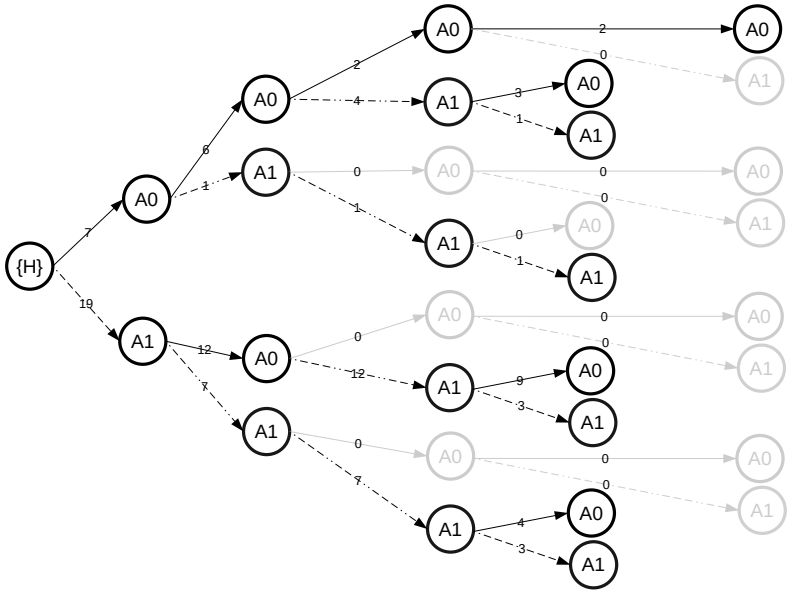


Figure 3.8:

Example of VLMC with haplotypes from sparse MNAR data.

The tree is formed from haplotypes counts at the 4 first markers in Figure 3.7. The root of the tree $\{H\}$ is not a marker. $A0$ represent the allele 0 and $A1$ the allele 1. Grey nodes and branches indicate that the allele sequences that were not observed in the available set of haplotypes. The observed haplotypes include the ones in the pooled study population. Because of the different allelic proportions at each marker, the tree of haplotypes is looking different than the tree form M(C)AR data. Some haplotypes e.g. 0110 are missing compared to the previous example in Figure 3.5, while other ones are over represented e.g. the haplotype 1010. This might have a significant impact on the later node merging step and notably modify the template haplotypes used for imputation, which in turn will affect the accuracy of the imputation results.

Chapter 4

Summary and future work

The work presented in this thesis describes pooling techniques tailored for genotype data of SNPs in the broader context of genotype imputation.

We have introduced the general context of genotype imputation and its relevance in many research fields, as well as the cost issue of large-scale genotyping. We have argued that a pooling technique augmented by imputation could contribute to address this cost challenge. The complementary nature of group testing for identifying items occurring at low frequency especially improves the genotyping accuracy of the rare variants. The rare variants are usually delicate to impute in a population, while determining the genotype of the samples at these markers they can be very valuable in e.g. biomedical association studies or marker-assisted selection in breeding.

We have investigated different strategies of pattern-consistent decoding for the pooled genotypes with the 4×4 NORB design we chose. These strategies are implemented in the *simpool* program that computes the most likely genotype probabilities of any sample in a pooling block. We however showed that the probabilistic decoding step in genotype pooling implies specific challenges that are due on the one hand to the ternary nature of the genotype data, on the other hand to the pooling design itself that introduces structural dependencies between the missing and the non missing genotypes in the pooled data.

Finally, we have demonstrated that the specific structure of the pooled genotype data poses some difficulties for imputation, both with coalescent and clustered tree methods. We have proposed hypotheses about the mechanisms underlying these difficulties, we believe they can be explained by imbalances in the missing data rate which is correlated the allelic frequencies at the markers to be imputed. Genotype imputation from pooled data has nonetheless shown good performance overall with the coalescence-based algorithm *Prophaser*. These promising results open opportunities for ap-

plications in biomedical research as well as in animal and plant breeding.

In order to evaluate the relevance and the efficacy of pooling for genotype imputation, we conducted in Paper I a study comparing two scenarios of genotype imputation in a study population sampled from the *1000 Genomes Project*. The first scenario simulates a situation where the data set to impute consists of markers that are either fully assayed in the study population to be genotyped, or fully missing for all samples. In this usual setup for genotype imputation, the genotype data is missing at random (MAR data). The existing methods for genotype imputation such as the coalescent models of MaCH or Impute2, as well as the Beagle model, have been developed for a usage in this scenario, and they have demonstrated very good accuracy and computational performance.

The second scenario simulated in Paper I implements genotype pooling with a 4×4 NORB design in the study population in a first step, followed by genotype imputation in a second step. The genotype data in this setup is missing not at random and shows nonmonotone missingness patterns (MNAR data), which impacts negatively the accuracy and the performance of the imputation methods. In order to address the particularities of the missing data in the case of pooling, we proposed two new tools. First, a self-consistent iterative algorithm (*simpool*) for inferring the most likely genotype of any missing item in a pooling block, based on the observed patterns of the pools. The probabilistic estimates computed with *simpool* partly overcome the difficulties encountered when imputing decoded data from pooling. Second, an extended coalescent method (*Prophaser*) that is able to make use of the estimates computed by *simpool* for improving the accuracy of imputation with pooled data. On the whole, the results presented in Figure 4 in Paper I demonstrate that the usage of pooling augmented by imputation benefits especially the genotyping of rare variants. While imputation in usual settings performs the best over all markers, we found that the rare variants are however efficiently identified thanks to pooling and imputed more accurately.

As the strategies for inference with MNAR data are more complex than with MAR data, Paper II proposes an investigation of the consistency of the genotype distribution reconstructed with the *simpool* algorithm. We evaluated the distributional consistency based on a divergence criterion. The new insights provided by this study let us improve the original algorithm, such that the versions of *simpool* implemented later output genotype probabilities that show higher consistency with the true genotype distribution. These results should be nonetheless interpreted in the context of genotype imputation in further investigations, that is to what extent the more consistent genotype probability estimates improve imputation accuracy.

Based on the results presented in Paper I, a short preliminary study of

the incorrectly imputed genotypes for a given pooling pattern let us believe that we should investigate to what extent a second iteration of both *simpool* and *Prophaser* on the data set. We will execute this second iteration from the imputed data set in Paper I.

Moreover, as our findings might benefit the plant breeding science, we also plan to carry out an experiment of joint pooling and imputation for genotype data with a crop species such as the *diverse MAGIC Wheat* inbred lines [44].

Bibliography

[1] A. Adler, G. Wiley, and P. Gaffney. Infinium assay for large-scale snp genotyping applications. *J Vis Exp*, 81(e50683), 2013.

[2] P. A. Alexandre, L. R. Porto-Neto, E. Karaman, S. A. Lehnert, and A. Reverter. Pooled genotyping strategies for the rapid construction of genomic reference populations. *Journal of Animal Science*, 97(12):4761–4769, 2019.

[3] A. Ameur et al. Swegen: a whole-genome data resource of genetic variability in a cross-section of the swedish population. *European Journal of Human Genetics*, 25:1253–1260, 2017.

[4] K. Ausmees and C. Nettelblad. Achieving improved accuracy for imputation of ancient DNA. *bioRxiv*, 2022.

[5] W. Bodmer and C. Bonilla. Common and rare variants in multifactorial susceptibility to common diseases. *Nat Genet*, 40:695–701, 2008.

[6] B. L. Browning and S. R. Browning. Efficient multilocus association testing for whole genome association studies using localized haplotype clustering. *Genetic Epidemiology*, 31:365–375, 2007.

[7] B. L. Browning and S. R. Browning. A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *The American Journal of Human Genetics*, 84:210–223, 2009.

[8] B. L. Browning and S. R. Browning. Genotype imputation with millions of reference samples. *The American Journal of Human Genetics*, 98:116–126, 2016.

[9] B. L. Browning, Y. Zhou, and S. R. Browning. A one-penny imputed genome from next-generation reference panels. *The American Journal of Human Genetics*, 103(3):338–348, 2018.

[10] S. R. Browning. Multilocus association mapping using variable-length markov chains. *Am. J. Hum. Genet.*, 78:903–913, 2006.

[11] S. R. Browning and B. L. 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*, 81:1084–1097, 2007.

[12] S. R. Browning and B. L. Browning. Haplotype phasing: existing methods and new developments. *Nature Reviews Genetics*, 12, 2011.

[13] M. P. L. Calus, T. H. E. Meuwissen, A. P. W. de Roos, and R. F. Veerkamp. Accuracy of Genomic Selection Using Different Methods to Define Haplotypes. *Genetics*, 178(1):553–561, 01 2008.

[14] C. Cao, C. Li, Z. Huang, X. Ma, and X. Sun. Identifying rare variants with optimal depth of coverage and cost-effective overlapping pool sequencing. *Genetic Epidemiology*, 37(8):820–830, 2013.

[15] C. Cao, C. Li, and X. Sun. Quantitative group testing-based overlapping pool sequencing to identify rare variant carriers. *BMC Bioinformatics*, 15(195), 2014.

[16] H.-B. Chen and F. Wang. A survey on nonadaptive group testing algorithms through the angle of decoding. *Journal of Combinatorial Optimization*, 15:49–59, 2008.

[17] X. Chi, X. L. anfd M.C.K. Wang, et al. An optimal dna pooling strategy for progressive fine mapping. *Genetica*, 135(267), 2009.

[18] C. Clouard, K. Ausmees, and C. Nettelblad. A joint use of pooling and imputation for genotyping SNPs. *bioRxiv*, 2021.

[19] F. S. Collins, M. Morgan, and A. Patrinos. The human genome project: Lessons from large-scale biology. *Science*, 300(5617):286–290, 2003.

[20] J. A. Collister, X. Liu, and L. Clifton. Calculating polygenic risk scores (prs) in uk biobank: A practical guide for epidemiologists. *Frontiers in Genetics*, 13, 2022.

[21] 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*, 39(1):1–22, 1977.

[22] E. Esteves, A. Mendes, M. Barros, C. Figueiredo, J. Andrade, J. Capelo, et al. Population wide testing pooling strategy for sars-cov-2 detection using saliva. *PLoS ONE*, 17(1), 2022.

[23] H. Gao, F. K. Hwang, M. T. Thai, W. Wu, and T. Znati. Construction of d(h)-disjunct matrix for group testing in hypergraphs. *Journal of Combinatorial Optimization*, 2006.

[24] A. Gomes and B. Korf. Chapter 5 - genetic testing techniques. In N. H. Robin and M. B. Farmer, editors, *Pediatric Cancer Genetics*, pages 47–64. Elsevier, 2018.

[25] J. He, X. Zhao, A. Laroche, Z.-X. Lu, H. Liu, and Z. Li. Genotyping-by-sequencing (gbs), an ultimate marker-assisted selection (mas) tool to accelerate plant breeding. *Frontiers in Plant Science*, 5:484, 2014.

[26] J. M. Heather and B. Chain. The sequence of sequencers: The history of sequencing dna. *Genomics*, 107:1–8, 2016.

[27] B. Howie, P. Donelly, and J. Marchini. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS Genetics*, 5(6), 2009.

[28] B. Howie and J. Marchini. Genotype imputation for genome-wide association studies. *Nature Reviews Genetics*, 11, 2010.

[29] G. Keeble-Gagnère, R. Pasam, K. L. Forrest, D. Wong, H. Robinson, J. Godoy, A. Rattey, D. Moody, D. Mullan, T. Walmsley, H. D. Daetwyler, J. Tibbits, and M. J. Hayden. Novel design of imputation-enabled snp arrays for breeding and research applications supporting multi-species hybridization. *Frontiers in Plant Science*, 12, 2021.

[30] A. Kho, L. Rasmussen, J. Connolly, et al. Practical challenges in integrating genomic data into the electronic health record. *Genet Med*, 15:772–778, 2013.

[31] N. Li and M. Stephens. Modeling linkage disequilibrium and identifying recombination hotspotsusing single-nucleotide polymorphism data. *Genetics*, 165(4):2213–2233, 2003.

[32] Y. Li, C. J. Wille, J. Ding, P. Scheet, and G. R. Abecasis. Mach: Using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genetic Epidemiology*, 34(8):816–834, 2010.

[33] G. Logsdon, M. Vollger, and E. Eichler. Long-read human genome sequencing and its applications. *Nat Rev Genet*, 21:597–614, 2020.

[34] J. Marchini, B. Howie, S. Myers, G. McVean, and P. Donnelly. A new multipoint method for genome-wide association studies by imputation of genotypes. *Nature Genetics*, 39:906–913, 2007.

[35] M. Mézard, M. Tarzia, and C. Toninelli. Group testing with random pools: Phase transitions and optimal strategy. *J Stat Phys*, 131:783–801, 2008.

[36] H. Q. Ngo and D.-Z. Du. A survey on combinatorial group testing algorithms with applications to dna library screening. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 49–59, 2000.

[37] M. Nothnagel, D. Ellinghaus, S. Schreiber, M. Krawczak, and A. Frankeg. A comprehensive evaluation of snp genotype imputation. *Human Genetics*, 125:163–171, 2009.

[38] T. Pook, M. Mayer, J. Geibel, S. Weigend, D. Cavero, C. Schoen, and H. Simianer. Improving imputation quality in beagle for crop and livestock data. *Genes Genomes Genetics*, 98:116–126, 2019.

[39] E. Porcu, S. Sanna, C. Fuchsberger, and L. G. Fritsche. Genotype imputation in genome-wide association studies. *Current Protocols in Human Genetics*, 1.25.1, 2015.

[40] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[41] D. B. Rubin. Inference and missing data. *Biometrika*, 63:581–592, 1976.

[42] N. Salcedo, A. Harmon, and B. B. Herrera. Pooling of samples for sars-cov-2 detection using a rapid antigen test. *Frontiers in Tropical Diseases*, 2, 2021.

[43] J. L. Scafer and J. W. Graham. Missing data: Our view of the state of the art. *Psychological Methods*, 7(2):147–177, 2002.

[44] M. F. Scott, N. Fradgley, A. R. Bentley, T. Brabbs, F. Corke, K. A. Gardner, R. Horsnell, P. Howell, O. Ladejobi, I. J. Mackay, R. Mott, and J. Cockram. Limited haplotype diversity underlies polygenic trait architecture across 70 years of wheat breeding. *bioRxiv*, 2020.

[45] P. Sham, J. Bader, I. Craig, et al. Dna pooling: a tool for large-scale association studies. *Nat Rev Genet*, 3:862–871, 2002.

[46] J. Shendure and al. Advanced sequencing technologies: Methods and goals. *Nature Reviews Genetics*, 5:335–344, 2004.

[47] M. Stephens, N. J. Smith, and P. Donnelly. A new statistical method for haplotype reconstruction from population data. *The American Journal of Human Genetics*, 68:978–989, 2001.

[48] C. Sudlow, J. Gallacher, N. Allen, V. Beral, P. Burton, J. Danesh, et al. Uk biobank: An open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS Medicine*, 12(3), 2015.

[49] P. Sudmant, T. Rausch, E. Gardner, et al. An integrated map of structural variation in 2,504 human genomes. *Nature*, 526:75–81, 2015.

[50] Y. J. Sung, L. Wang, T. Rankinen, C. Bouchard, and D. Rao. Performance of genotype imputations using data from the 1000 genomes project. *Human Heredity*, 73:18–25, 2012.

[51] N. Thierry-Mieg. A new pooling strategy for high-throughput screening: the shifted transversal design. *BMC Bioinformatics*, 7(28), 2006.

[52] S. van Buuren. *Flexible Imputation of Missing Data (2nd ed.)*. 2018.

[53] J. Wang et al. Investigation of rare and low-frequency variants using high-throughput sequencing with pooled dna samples. *Nature Scientific Reports*, 6(33256), September 2016.

[54] K.-C. Wong. Letter to the editor: Big data challenges in genome informatics. *Biophysical Reviews*, 11:51–54, 2019.

[55] A. G. Y. Erlich, K. Chang et al. Dna sudoku—harnessing high-throughput sequencing for multiplexed specimen analysis. *Genome Research*, 19:1243–1253, 2009.

[56] P. Zhang, F. Krzakala, M. Mezard, and L. Zdeborova. Non-adaptive pooling strategies for detection of rare faulty items. *Lecture Notes in Computer Science and Workshop on Algorithms and Data Structures 2005: Algorithms and Data Structures*, 2013.

Paper I

METHODOLOGY

A joint use of pooling and imputation for genotyping SNPs

Camille Clouard[1*], Kristiina Ausmees[1] and Carl Nettelblad[1]

Abstract

Background: Despite continuing technological advances, the cost for large-scale genotyping of a high number of samples can be prohibitive. The purpose of this study is to design a cost-saving strategy for SNP genotyping. We suggest making use of pooling, a group testing technique, to drop the amount of SNP arrays needed. We believe that this will be of the greatest importance for non-model organisms with more limited resources in terms of cost-efficient large-scale chips and high-quality reference genomes, such as application in wildlife monitoring, plant and animal breeding, but it is in essence species-agnostic.

The proposed approach consists in grouping and mixing individual DNA samples into pools before testing these pools on bead-chips, such that the number of pools is less than the number of individual samples. We present a statistical estimation algorithm, based on the pooling outcomes, for inferring marker-wise the most likely genotype of every sample in each pool.

Finally, we input these estimated genotypes into existing imputation algorithms. We compare the imputation performance from pooled data with the Beagle algorithm, and a local likelihood-aware phasing algorithm closely modeled on MaCH that we implemented.

Results: We conduct simulations based on human data from the *1000 Genomes Project*, to aid comparison with other imputation studies. Based on the simulated data, we find that pooling impacts the genotype frequencies of the directly identifiable markers, without imputation. We also demonstrate how a combinatorial estimation of the genotype probabilities from the pooling design can improve the prediction performance of imputation models. Our algorithm achieves 93% concordance in predicting unassayed markers from pooled data, thus it outperforms the Beagle imputation model which reaches 80% concordance. We observe that the pooling design gives higher concordance for the rare variants than traditional low-density to high-density imputation commonly used for cost-effective genotyping of large cohorts.

Conclusions: We present promising results for combining a pooling scheme for SNP genotyping with computational genotype imputation on human data. These results could find potential applications in any context where the genotyping costs form a limiting factor on the study size, such as in marker-assisted selection in plant breeding.

Background

Genotyping DNA markers at high density

Biological and medical research e.g. association studies or traits mapping have been interested in Single Nucleotide Polymorphisms (SNPs) genotypes because of their numerous advantages as genetic markers [1]. Among the various tools performing SNP genotyping, the genotyping chips technology (bead-chips) is well-suited for processing many variants at a time.

In association studies, SNPs are used to differentiate subpopulations or individuals from one another when they can be clustered into informative patterns of genetic variation within a sample. Tens or hundreds of thousands of SNPs are often required for achieving relevant, informative, and significant associations or mapping [2]. Despite their abundance, many of the SNPs carrying variation patterns of relevance can be categorized as (extremely) rare variants, e.g. variants with a population frequency less than 1%. Consequently, a large cohort of individuals should be processed to detect these variations and their effects. Computational approaches based on appropriate algorithms offer solutions for increasing both the amount of genotyped markers and the study population size at a reasonable cost. The computational solutions represent a midway to the dilemma of choosing between genotyping a large population at low-density only, or obtaining high-density genotypes sets but for a restricted number of individuals.

*Correspondence: camille.clouard@it.uu.se

[1], Division of Scientific Computing, Department of Information Technology, Uppsala University, Läderhyggsvägen 2, 75105 Uppsala, Sweden

Full list of author information is available at the end of the article

A common method to reduce the genotyping cost is to genotype a low-density (LD) set of markers in a study population and to infer a high-density (HD) one. The inference process, which we refer to as classical imputation, is based on a reference population that is assumed to be similar to the study one, and where the genotypes of all markers are known. Imputation methods have demonstrated high accuracy for inferring unassayed genotypes in a population. Nonetheless, several studies found imputation usually performs less well for the rare variants relatively to the common ones [3–7].

Saving genotyping costs with combinatorial group testing techniques

Pooling is a group testing technique that aims to identify defective samples in a population with the fewest tests possible. Its usage for genetic screening or compressed genotyping was suggested in the 1990s [8]. Numerous studies have proposed the use of pooling for tackling the cost issue for DNA processing [9–11], for instance when conducting DNA variant detection tasks on 96-well PCR-plates. Pooling turns out to be particularly efficient when dealing with the detection of rare variants, as other applications in association studies also show with human [9], animal, and crop data [12, 13]. In this context, the carriers of rare variants are seen as the "defective" items. The applications of DNA pooling in association studies has been mostly used for estimating allelic or haplotype frequencies that are derived from the pooled genotype frequencies. Several papers proposed statistical models that incorporate error-correction mechanisms for taking into account

the noisy genotype data from pools. In some cases, the statistic used for testing the allelic association is corrected with the variance of the estimates in the case and the control populations [14–17]. In other cases, the models relies on linear regression models for handling the genotyping errors when estimating the allelic or haplotypic frequencies [18]. More recently, genotype pooling in cattle has been suggested as an avenue for more efficient breeding value estimates in large populations [19].

We propose to implement a similar pooling strategy in order to reduce the cost of SNP genotyping, without sacrificing the power to detect carriers of low-MAF (minor allele frequency) variants or shrinking the study population size. In practice, this is accomplished by pooling samples before them being tested on the SNP chips, with each sample being included in multiple pools. The individual genotypes are then reconstructed based on the test results from the pools. Our study does not target to estimate the overall allelic frequencies at markers, it rather aims to find a large-scale and moderate-cost genotyping method that focuses on the accuracy of every individual genotype estimated.

Various combinatorial group testing schemes have been explored in the literature. These schemes, also called pooling designs or algorithms, can be split into two families, the sequential and the non-adaptive. In the first case, groups (or pools) are consecutively built from the data and tested in several steps whereas in the latter, all groups are constructed and tested at once simultaneously. Since we test all markers on the SNP chip simultaneously in our pooling design, only non-adaptive group testing (NGT) algorithms are suitable for our study [2, 20].

For uniquely identifying and keeping track of every individual contribution to the pool, the designs with overlapping pools were found to be effective and accurate [2, 21–23]. Among the strategies that have been studied for assigning the individuals into overlapping pools, we found mentioned in the literature the DNA Sudoku approach [9] and the Shifted Transversal Design (STD) [24, 25]. Both present a deterministic algorithm for recovering the individual test results from the pools. We have also noted other approaches as compressed sensing [2, 24, 26] which are particularly suitable for processing the rare variants and incorporate probabilities in the decoding step. Our design is a simple case of STD which partitions the samples to be pooled into repeated blocks, where each block corresponds to a pair of layers [20]. Given the characteristics of the pooling design we implement in this study, we designate it by Nonadaptive Overlapping Repeated Block (NORB) design.

When attempting to decode individual genotypes from the pools, some ambiguity may arise, resulting in missing genotype data for some individuals and markers [2, 9]. This drawback is particularly strong when the defective and the non-defective items are in comparable proportions in the population. In our setting where defectives correspond to minor allele carriers at SNPs, this situation is likely to be encountered with the common variants. As suggested by He et al. [23], a likelihood framework can be used for formulating the pooling problem as an extension to the combina-

torial methods. The authors found that the likelihood framework and its flexibility is especially suitable for applications that target the accurate genotyping of a population. In this study, we propose to first estimate the likely distribution for each incomplete pooling outcome, and then do a full imputation of all missing genotypes in the data set using more traditional genotype imputation methods.

Improving pooled genotyping results with imputation methods

Genotype imputation refers to computational approaches for inferring genotypes based on incomplete or uncertain observational data in a population. Many well-performing algorithms for imputation use Hidden Markov Models (HMM) [3, 27] that exploit haplotype-frequency variations and linkage disequilibrium. Other statistical methods such as SNP-tagging based approaches can be found but are not as accurate.
Imputation has been widely used on human genetic data [27–29], but also on plant or animal DNA more recently [30, 31]. To consider pooling and imputation together has been suggested for improving the decoding process performance when genotyping rare variants [10].

On a general level, the imputation problem can be formulated as resolving ambiguous or unknown genotypes with predictions by aggregating population-wide genetic information [3]. Besides the reference population, some imputation methods can incorporate the relatedness between the study individuals, if such data are provided.

We focused on population-based imputation methods, designed for dealing with unrelated individuals. An extensive investigation of the performance-critical parameters that drive imputation is out of the scope of this study, as well as the family-based methods which include pedigree information in the computations. Due to the very common case of very large populations with significant cost constraints in important applications such as animal and plant breeding, we believe that pedigree-aware imputation methods could form an excellent fit with pooling in that context.

Within the population-based methods, two main approaches have been dominating for a long time, namely the tree-based haplotypes clusters and the coalescent models [3, 32]. More recent approaches tend to build on these, but they locally subsample the references based on index searches. We have not included those in this study, since the decoding of pools renders complex patterns of genotype probabilities.

Both population-based models are statistical methods that yield probabilistic predictions for the missing genotypes. They implement HMM based on template haplotypes, but with some differences. In coalescent models, the probabilistic estimation of the genotypes at unassayed markers is computed from a stochastic expectation-maximization (EM) method. Tree-based clustering, implemented in the Beagle software, is an empirical model determined by the counts of similar segments found across the template haplotypes. For both the coalescent and the tree-based models, the hidden states underlying the Markov chain of the HMM are defined by single or aggregated template haplo-

types. The way this set of template haplotypes is constituted varies with the imputation method used. The transition from one haplotypic state to another between two consecutive markers mimics a historical recombination event, while the emitted symbols of the HMM are the genotypes, which are modeled as possibly erroneous copies of the hidden pair of haplotypes and hence express mutation events. Depending on the approach, recombination and mutation phenomena are either explicitly parametrized, or captured implicitly.

Among the coalescent models, MACH and IMPUTE2 have been found to perform the best in different studies [27, 29, 33, 34]. We implemented a similar method based on [35] and we refer to this algorithm as *Prophaser* [36] in this paper. To the difference of the common practice in MACH and IMPUTE2, *Prophaser* uses all the available template haplotypes as hidden states in the HMM. All aforementioned methods and software run one HMM for each study individual, and yield probabilistic estimates of the missing genotypes.

IMPUTE2 and MACH form the HMM hidden states by selecting h template haplotypes in both the reference and the study population, such there is a constant number h^2 hidden states at each of the j diploid markers. Hence, these methods have a complexity $\mathcal{O}(jh^2)$ in time for each study individual [37], and the time complexity grows linearly as the size of the study population. Despite the use of a memory-saving technique recomputing parts of the forward-backward table on the fly, turning the memory complexity to $\mathcal{O}(\sqrt{j}h^2)$, several papers point out computational efficiency issues with MACH [3, 27, 32] when compared to the other methods mentioned. By contrast, Beagle operates a dimension reduction of the hidden states space thanks to its clustering approach, which has been shown to be particularly efficient when imputing large data sets. The successive releases have improved the software performance in this direction [32, 38–41]. In this study, we use Beagle as a comparison baseline for imputation.

Scope of the study

In this paper, we present a new cost-effective genotyping approach based on the joint use of a pooling strategy followed by imputation processing. We analyze how a pooling procedure, applied on a large data set, impacts what we can conclude about the underlying distribution of genotype frequencies in the study population.

We also evaluate how conventional imputation methods perform when given such a pooled data set which has an unusual and characteristic genotype distribution. Specifically, we investigate if refining the specification of ambiguous genotypes based on the combinatorial outcomes can improve imputation performance. The proposed specific pooling scheme is not unique, however it proves to be a reasonable starting point for evaluating the promise of such designs. Furthermore, we focus solely on the computational aspects of determining genotypes. In practice, proper schemes for performing pooling and SNP genotype quality control would be needed. The resilience of imputation methods to patterns of fully missing markers or fully random genotyping noise is well-known and therefore also not a focus of this study.

Methods

Genotyping scenarios

In order to first evaluate how bead-chip genotype data respond to pooling treatment and second, how imputations methods perform on pooled data, we designed the following simulation experiment. We build two marker sets with genotype data from a human population at low respectively high density (LD resp. HD data sets) by extracting only those markers from the *1000 Genomes Project* (1KGP) data set that are present in one lower-density and one higher-density Illumina bead-chip in common use. We then compare the performance of two approaches for genotyping markers at high-density. The first approach serves as a baseline and simulates a usual study case where part of the markers are genotyped at low density in a target population, and the rest of the markers are imputed based on a high-density reference panel. The second approach evaluates genotyping markers at a high density from pools of individuals and then using imputation for those individual genotypes that are not fully decodable from the pooling.

Data sets and data preparation

We use data from the well-studied reference resource made available by the 1KGP, more specifically phase 3 v5 [21, 29, 42–44], providing genotype data over 2504 unrelated human individuals across 26 subpopulations analyzed worldwide [45].

We select markers from chromosome 20 that has been studied in several previous papers [5, 41, 46]. This chromosome spans approximately 63 million DNA base pairs [42]. Within the 1KGP in the phase 3 version released 2015, $1,739,315$ variants are genotyped as biallelic SNPs, out of which $1,589,038$ (91.4%) have a minor allele frequency (MAF) less than 5%. These are called rare or low-frequency variants [37, 47].

After selecting the biallelic SNPs, we retain markers that are common to both the 1KGP chromosome 20 data set and analyzed on the Illumina bead-chip products *Infinium OmniExpress-24 Kit* and *Infinium Omni2.5 - 8 Kit*. Intersecting the markers from the Illumina arrays and the markers genotyped in the 1KGP for the chromosome 20 yields two overlapping experimental maps. The map derived from the OmniExpress bead-chip consists of $17,791$ biallelic markers, out of which $17,015$ markers are shared with the map derived from the Omni2.5 bead-chip which lists in total $52,697$ markers (see Figure 2a). With respective densities of 1 SNP per 3.5 kb and 1 SNP per 1.19 kb, we hence obtain low-density (LD) and high-density (HD) marker sets [38].

For simulating imputation, the 2504 unrelated human samples are randomly split into two populations, regardless of their subpopulation. The first one is the reference panel (PNL) with 2264 individuals, the latter is the study population (STU) with 240 individuals, thus observing proportion PNL:STU-sizes of ca. $10:1$ as in [3]. For the classical imputation scenario simulation, we delete in the STU population genotype data for the markers only present in the HD data set and keep fully genotyped at LD the $17,015$ markers common to both maps. In the pooling scenario, we keep all the $52,697$ HD in STU and simulate pooled genotypes as described hereafter. In PNL, we keep the genotype

data for all LD and HD markers for both scenarios. Figure 1 gives an overview of the experimental steps carried out in both scenarios.

Figure 2 illustrates the composition of the different data sets composition before imputation. In both scenarios, after imputation, the study population is eventually fully genotyped at HD markers.

Group testing design for simulating pooled genotyping from microarrays data

The study population is further processed with pooling simulation, which yields missing genotypes spread in the data.

Based on the DNA Sudoku study [9], we define critical parameters for optimizing the design which are the number of individuals per block, the number of intersecting pools per block holding each pair of samples, and the number of pools that hold any given sample. These parameters and the pooling algorithm can be mathematically formulated as a binary $k \times m$ matrix M with k rows representing pools and m columns representing samples. M is called the design matrix of the scheme.

NORB parameters and design matrix We choose $n_B = 16$ samples for the block size with pools of degree 4, a samples' weight equal to 2, and a pool intersection value equal to 1. Hence, we get a number of pools per block equal to 8. The reduction factor ρ is 2, or equivalently the number of individuals is twice the number of pools within a block.

Square representation of a block We introduce a graphical representation of a pooling block with genotypes at a given SNP, according to the chosen parameters. As described by Ngo and Du in their taxonomy of nonadaptive pooling designs [25], a simple transversal design can be represented as a grid. The rows and columns $\{P_t\}_{1 \leq t \leq T}$ are the pools, and $\{G_i\}_{1 \leq i \leq n_B} \in \{-1, 0, 1, 2\}$ the individuals' genotypes which is, in order, interpreted as 'missing genotype', 'homozygous for the reference allele', 'heterozygous', 'homozygous for the alternate allele'.

	P_5	P_6	P_7	P_8
P_1	G_1	G_2	G_3	G_4
P_2	G_5	G_6	G_7	G_8
P_3	G_9	G_{10}	G_{11}	G_{12}
P_4	G_{13}	G_{14}	G_{15}	G_{16}

Pooling is simulated on the genotypes in the study population (STU data set) for the imputation scenario 2 (pooled HD data). STU was created in view of having a size which is a multiple of the block size chosen, i.e. STU has a size $B_{stu} * n_B = 15 * 16$, where B_{stu} is the number of pooling blocks formed from the study population. At every SNP, we implemented the pooling simulation as described hereafter.

Encoding and decoding rules With the design we have selected for our experiment, simulating pooling on items involves an encoding step followed by a decoding step. Two examples of genotype pooling simulation are shown in Figure 3a and Figure 3b.

First, the encoding step simulates the genotype outcome for a pool from the combination of the individual

genotypes in it. SNP chip genotyping detects which alleles are present in the sample at each SNP (0 for the reference allele or 1 for the alternate allele) on the chip. That means, in the simulation of the pooling encoding step, a pool has genotype 0 (respectively 2) if and only if all samples forming the pool are homogeneous and have homozygous genotype 0 (resp. 2). Any other intermediate combination of a pool from samples having heterogeneous genotypes 0, 1, or 2 results in a heterozygous pool with genotype 1.

In the second step, decoding the individual genotypes from their intersecting pools is done while assuming there was no genotyping error. In our design, every sample is at the intersection of two pools. If both pools have genotype 0 (or 2), the sample has genotype 0 (or 2). Also, since a pool has a homozygous genotype if and only if all contributing samples have the homozygous genotype, this implies that any individual at the intersection of a homozygous pool and a heterozygous one must be homozygous. In the case of a pooling block with exactly one carrier of the alternate allele (Figure 3a), if exactly two pools have a heterozygous genotype 1 (pools P_3 and P_5 in Figure 3a), we deduce the individual at their intersection has the alternate (or reference) allele, but we cannot state if two copies of this allele are carried (genotype 2, or 0 in the symmetrical case where the reference allele is the minor one) or only one (genotype 1). In this case, ambiguity arises at decoding, in other words, genotype data is reported as missing. To fully assess the probable state of the genotypes of each sample in a pooling block, not only the pools where a sample is included have to be considered but also the full block. We propose to make use of the constraints imposed by the outcome for each pool to estimate the genotype distribution for any undecoded sample. This includes the distribution between heterozygote and homozygote for decoded carriers.

Figure 3c and Figure 3d show some results we obtain after simulating pooling and imputation at two markers for $4 \times 16 = 64$ samples in the study population: Figure 3c is an example for a common variant and Figure 3d illustrates the case of a rarer variant. In practice, genotyping pools of samples on microarrays requires computational processing of the decoding step only.

Estimation of the genotype probabilities from combinatorial information

At the block level, the pooling scheme implies possible and impossible latent genotypes for a given sample. For example, a decoded block comprising twelve REF-homozygous and four missing genotypes as in Figure 3b imposes the constraint at least two out of the four samples are minor allele carriers (i.e. genotype in $\{1, 2\}$), whereas the other missing samples can have any genotype in $\{0, 1, 2\}$. Consequently, within these four unknown sample states, the probability of encountering actual homozygous-REF is lower than in a case where the missingness pattern of genotypes is independent of the actual genotype value, as is typically the case in imputation from low to higher density. By proceeding in a similar way for any observable pooling block, we propose to explicitly model the expected distribution of each incompletely decoded genotype.

Genotype representations

In this paper, beyond the G representation introduced previously, we use the genotype probabilities (GP) format, which expresses any genotype as a probability simplex over the three zygosity categories. G and GP are equivalent representations, for example if all genotype states are uniformly equally likely to be observed, this results in a genotype probability $GP = (0.33, 0.33, 0.33)$ (i.e. $G = -1$). A determined genotype has one of the following probabilities: $GP = (0, 0, 1)$, $(0, 1, 0)$, or $(1, 0, 0)$ (i.e. $G = 2$, $G = 1$, or $G = 0$).

Statistical formulation of the genotype decoding problem

We introduce hereafter the notations and definitions which frame the pooling procedure as a statistical inference problem in missing data. In this framework, we later present an algorithm for estimating the most likely genotype at any missing entry conditioned on the configuration of the pooling block. Our strategy proceeds by enumerating genotype combinations for the missing data that are consistent with the data observed from the pooling blocks, and uses that enumeration to compute an estimate of the latent genotype frequencies.

Model distribution for the genotypes Let the genotype G be a random variable with three outcomes 0, 1, and 2. The genotype probabilities π are expressed as

$$\pi = (p_0, p_1, p_2) \tag{1}$$

where (p_0, p_1, p_2) are the probabilities for the genotype 0, 1, and 2 at a given variant for a given sample. Therefore, we model the complete (not pooled) genotype data within a pooling block as an array $\mathbf{x}$ of size 16×3 ($n_B = 16$) where each data point x_i is a probability simplex $[p_{0i}, p_{1i}, p_{2i}]$. Each probability simplex is an indicator vector, since the genotype is fully known.

$$\mathbf{x} = (x_1, x_2, \ldots, x_{16}) \tag{2}$$

$$\forall i \in [1, 16] \qquad x_i = \begin{bmatrix} p_{0i} \\ p_{1i} \\ p_{2i} \end{bmatrix} \tag{3}$$

Since the samples are randomly assigned to pooling blocks, the genotype probabilities x_i are independent from each other.

Furthermore, we denote $\mathbf{z}$ the prior probabilities for genotypes that follow pooling and pool decoding. $\mathbf{z}$ is another list of probabilities, where some genotypes are fully decoded, some are fully unrecoverable, and some indicate carrier status, without being able to distinguish between a heterozygous genotype or a homozygous one as on Figure 3a. The pooled genotypes are represented by

$$\mathbf{z} = (z_1, z_2, \ldots, z_{16}), \tag{4}$$

$$\forall i \in [1, 16] \qquad z_i = \begin{bmatrix} \tilde{p}_{0i} \\ \tilde{p}_{1i} \\ \tilde{p}_{2i} \end{bmatrix} \tag{5}$$

The data z_i for each cell of a pooling block is modelled with the simplex of genotype probabilities $(\tilde{p}_{0i}, \tilde{p}_{1i}, \tilde{p}_{2i})$.

Mapping of the data space We denote *layout* the data for the full genotypes $\mathbf{x}$, which is represented as a list of genotype probabilities for each individual in the block. We denote t the function transforming $\mathbf{x}$ into $\mathbf{z}$. Since there are several complete layouts $\mathbf{x}$ that could give the same result $\mathbf{z}$ after pooling, t is a many-to-one mapping

$$t\colon \mathcal{X} \longrightarrow \mathcal{Z} \tag{6}$$

$$\mathbf{x} \longmapsto \mathbf{z} \tag{7}$$

where $\mathcal{X}$ is the space of complete observations, and $\mathcal{Z}$ is the space of decoded pooling blocks.

Given the priors z_i for any sample, the problem to solve is to estimate a posterior probability distribution $\hat{\pi}_i = (\hat{p}_{0i}, \hat{p}_{1i}, \hat{p}_{2i})$ for the three genotypes $\{0, 1, 2\}$ in any individual, i.e. recovering a probability distribution from which the true genotype x_i can be said to be sampled, as a probabilistic inversion of t.

Inherently to the NORB design chosen, the assortment of observable $\mathbf{z}$ is finite and constrained. Moreover, any individual genotype z_i depends on the genotypes of the pools intersecting it, but also on all other pools in the block. Therefore, any sample z_i in the full set of probabilities $\mathbf{z}$ representing the pooling block can be parametrized by the pool configuration and the possible intersections.

Valid layouts in block patterns Let ψ be the pooling block pattern described as $\psi = (n_{G_{rows}}, n_{G_{columns}})$, where $n_{G_{rows}}$ (resp. $n_{G_{columns}}$) are the counts of row-pools (resp. column-pools) with encoded genotypes $(0, 1, 2)$. For example, on Figure 3a, the 8 pools can be described with the block pattern $\psi = ((3, 1, 0), (3, 1, 0))$ since there are 3 row-pools having genotype 0, 1 having genotype 1, none having genotype 2, and the same for the column-pools. On Figure 3b, the pooling pattern is $\psi = ((2, 2, 0), (2, 2, 0))$.

We denote $\mathcal{Z}_\psi$ the space of decoded pooling blocks showing the pattern ψ, and correspondingly $\mathcal{X}_\psi$ the space of the set of valid layouts for ψ. A layout is said to be valid with respect to the pattern ψ if applying pooling simulation to $\mathbf{x}$ lets us observe ψ from $\mathbf{z}$. In other words, the valid layouts are

$$\mathcal{X}_\psi = \{t_\psi(\mathbf{x}) \in \mathcal{Z}_\psi : \mathbf{x}\}. \tag{8}$$

The Additional file shows examples of valid and invalid layouts for the same observed pooling pattern.

Parametrizing the data mapping Let $(r, c) \in \{0, 1, 2\}^2$ be the genotype pair of two intersecting pools, such that any z_i is conditioned on (r, c) . We note that if $(r, c) = (1, 1)$, the decoding of the intersected individual genotype z_i is indeterminate. In other cases, the intersected genotype is fully recoverable as with $(0, 1)$ (resulting in $z_i = [1, 0, 0]^\top$). The pair $(r, c) = (0, 2)$ is not consistent with any genotype, therefore it is never observed.

Based on these notations, we seek to approximate the most likely genotype probabilities $\{\hat{\pi}_i\}$ in missing data that are consistent with x_i by using inversion sampling of the priors z_i with respect to t_ψ. That is to say,

$$Pr(x_i|\psi; r, c) = t_\psi^{-1}\big(Pr(z_i|\psi; r, c)\big). \quad (9)$$

Computing the estimate of the posterior for the missing outcomes as $\hat{\pi} := \overline{\hat{\pi}_i}$ in a pooling block with pattern ψ by inverse transform sampling is a numerical problem that can be solved as a maximum likelihood estimation (MLE) based on the enumeration of all valid layouts.

Maximum Likelihood type II estimates

We propose to partition $\mathcal{Z}$ into $\{\mathcal{Z}_\psi\}_{\psi\in\Psi}$. This enables to marginalize the likelihood over ψ, r, c and lets the problem be solved as a series of separate probability simplex MLE problems in each sample subspace $\mathcal{Z}_\psi$. The marginal likelihood is sometimes found as type II-likelihood (ML-II) and its maximization (MMLE) as empirical Bayes method. We present as supplementary information a method for computing $\hat{\pi}$ by maximizing the marginal likelihood of any observed pattern ψ and deriving genotype posterior probabilities estimates (see Additional file). The MMLE example is also well-suited for introducing how we conduct a systematic and comprehensive enumeration of the valid layouts for a given pattern ψ.

Self-consistent estimations

Motivation and general mechanism As a natural extension to the MMLE in presence of incomplete data [48], we implemented a method for estimating the unknown genotypes probabilities inspired by the EM algorithm. The following procedure is applied for each set of parameters ψ, r, c.

We initiate the prior estimate of any entry in the block to $z_i = [0.25, 0.5, 0.25]^\top$. This choice is based on the assumption that, without information about their frequencies, both alleles at a marker are expected to be equally likely carried.

The algorithm iteratively updates $\tilde{\pi} := \overline{\tilde{z}_i}$ by alternating between computing the likelihood of the valid layouts using the prior estimate (E step) and deriving the posterior estimate from the frequencies of the genotypes aggregated across the data completions (M step). The M step can incorporate a rescaling operation of the proportions of genotypes that we designate as heterozygotes degeneracy resampling. Eventually, the E and M steps produce a self-consistent estimate $\hat{\pi}$ [49] (see Additional file for a calculation example).

Heterozygote degeneracy arises from the internal representation we use for the genotypes under the pooling process. Indeed, the two heterozygous states carrying the phased alleles pairs $(0, 1)$ or $(1, 0)$ are collapsed into a single heterozygous genotype $GP = (0, 1, 0)$ (or equivalently $G = 1$). In a way analogous to for example the particles paths in particles filter models, we define this collapsing as heterozygous degeneracy. For instance, a layout involving 4 heterozygous genotypes should be subdivided into 2^4 micro layouts

combining alleles pairs $(0,1)$ and $(1,0)$. More generally, the heterozygous degeneracy has order 2^{n_1}, where n_1 is the number of items having genotype 1 in the layout. In practice, enumerating these micro layouts would increase the computation time a lot. Instead, we include the higher probability for heterozygotes internally in the model, taking the degeneracy into account when normalizing, and again when producing the final likelihoods to be used in the imputation process, where a uniform distribution is the expected structure for data without any informative prior.

Equations of the optimization problem We proceed in a way identical to MMLE for enumerating all possible completions for the n_m unknown genotypes. At each iteration m, The E step calculates first the marginal likelihood of every layout by sampling its genotypes from $\tilde{\pi}^{(m-1)}|\psi$. The mixing proportion $\mathbb{E}[\mathbf{x}|\mathbf{z},\tilde{\pi},\psi]^{(m)}$ of each layout is computed from all aggregated likelihoods and for any $\mathbf{z} \in \mathcal{Z}_\psi$. A breakdown of the formula for $\mathbb{E}[\mathbf{x}|\mathbf{z},\tilde{\pi},\psi]^{(m)}$ is provided in the Additional file.

The M step recomputes the genotype frequencies $(\tilde{p}_0,\tilde{p}_1,\tilde{p}_2)$ by applying MLE to the likelihoods calculated at the E step.

$$\tilde{p_k}^{(m)} = \frac{\sum\limits_{\mathbf{x}\subset\mathcal{X}} n_k\,\mathbb{E}[\mathbf{x}|\mathbf{z},\tilde{\pi},\psi]^{(m)}}{\sum\limits_{k}\sum\limits_{\mathbf{x}\subset\mathcal{X}} n_k\,\mathbb{E}[\mathbf{x}|\mathbf{z};\tilde{\pi},\psi]^{(m)}}, \tag{10}$$

$$k \in \{0,1,2\} \tag{11}$$

where n_k is the counts of genotype k observed in the layout $\mathbf{x}$.

Since we do not compute the distribution of the genotype frequencies from the allelic dosage, we suggest a resampling step after the M step that artificially accounts for the heterozygous degeneracy. Hence, we introduce arbitrary weights $w = (w_0,w_1,w_2) = (1,2,1)$ for rescaling $(\tilde{p}_0,\tilde{p}_1,\tilde{p}_2)$. If we do not account for the heterozygote degeneracy, we pick these weights as $w = (1,1,1)$.

$$\tilde{p}_k^{(m)\prime} = \frac{w_k\,\tilde{p}_k^{(m)}}{\tilde{p}_k^{(m-1)}},\ k \in \{0,1,2\} \tag{12}$$

$$\tilde{p}_k^{(m)\prime\prime} = \frac{\tilde{p}_k^{(m)\prime}}{\sum\limits_k \tilde{p}_k^{(m)\prime}} \tag{13}$$

$$\tilde{\pi}^{(m)} = (\tilde{p}_0^{(m)\prime\prime},\tilde{p}_1^{(m)\prime\prime},\tilde{p}_2^{(m)\prime\prime}). \tag{14}$$

At the last iteration, when the algorithm has converged, the final estimate of $\tilde{\pi}$ is computed from a modified version of rescaling, where we compensate for the artificial upscaling used in the previous steps

$$\hat{p_k}^{(m)}|\psi = \frac{(1/w_k)\,\tilde{p}_k^{(m)}}{\sum\limits_k (1/w_k)\,\tilde{p}_k^{(m)}},\ k \in \{0,1,2\} \tag{15}$$

$$\hat{\pi}|\psi = (\hat{p_0}^{(m)},\hat{p_1}^{(m)},\hat{p_2}^{(m)}) \tag{16}$$

Such self-consistent iterative methods provide local distribution estimates for the undecoded genotypes at the pooling block level, based on information from the pooling design. They are independent of the overall MAF in the population because of the choice we made for the prior, and do not take into account the genetic

variations specific to the population and its structural traits.

Imputation for retrieving missing genotypes

For each sample in the study population, we use the aforementioned estimated genotype probabilities $\tilde{\pi}|\psi, r, c$ as prior beliefs θ_G in imputation. Figure 2 summarizes the experimental settings for both this scenario and the classical one. We compare the imputation performance on pooled SNP genotype data of two population-based algorithms, representing each the haplotype clustering approach and the coalescence principle.

A haplotype clustering method: Beagle

In this work, Beagle is used in its 4.0 version and with the recommended default parameters. This software version is the best performing release having the features needed for this study. Beagle 5.0 is available but this version does not support logged-GP (GL) data type as input.

We use the HapMap GRCh37 genetic map suggested by Beagle developers and consistent with the genome assembly underlying the version of the 1KGP data used [38]. In practice though, we have not noticed clear deterioration when conducting imputation on pooled data without providing any genetic map.

For the classical imputation scenario, we beforehand verify equivalent results and performance are obtained both if Beagle is run on genotypes in a GT format or GL format. In the first case, unassayed HD markers were set to ./. and in the latter, to $(-0.481, -0.481, -0.481)$. As advised in the documentation, we imputed the entire STU population in the same batch.

In the pooling scenario, we used the same reference panel, but we deliberately chose to run Beagle sample-wise for avoiding the very specific genetic structure of pooled data being used as template haplotypes. Preliminary testing showed a clear deterioration in results if this was not done.

A coalescence-based method for haplotype phasing and imputation: Prophaser

The original version of MACH did not support GL as input study data, in contrast to IMPUTE2. The main motivation for writing the *Prophaser* [36] code was to implement this feature with full control of e.g. cutoff thresholds for close-to-zero probabilities. The reference panel is read from GT data.

Prophaser phases and imputes unassayed markers sample-wise and independently from the rest of STU. Whereas MACH and IMPUTE2 include strategies for selecting a subset of reference samples for computational efficiency reasons, we decided to consistently use the full reference panel as templates in a single iteration estimation. Hence, *Prophaser* uses all reference haplotypes as templates.

Evaluation of the experimental design

We quantified the performance of the two genotyping scenarios with the concordance rate and cross-entropy. In both cases, the original data from 1KGP in the study population were used as the ground truth, and the predicted data were the imputed genotypes in the same study population.

Concordance The most widely used imputation quality metric is the genotype concordance measure which counts the number of matches between the true and the best-guess imputed genotypes. A homozygous genotype imputed as heterozygote (or conversely) is counted as a half mismatch, and a homozygote imputed to its opposite homozygote as a full mismatch. Concordance sometimes appears as its complementary formulation with the discordance rate [3]. Several publications refer to the concordance rate directly as the genotype accuracy rate [39] or as imputation accuracy [32], whilst the discordance rate is designated as the imputation error rate [33, 38].

Cross-entropy In the studies presenting the successive Beagle software versions, the accuracy in the sense of the concordance does not quantify how similar the imputed genotypes are to the true ones. This has already been pointed out by e.g. Nothnagel et al. [27]. As an example, we can consider the two following cases: *(a)* a true genotype $G = 1$ being imputed with $GP = (0.56, 0.42, 0.02)$, and *(b)* a genotype $G = 1$ being imputed with $GP = (0.7, 0.28, 0.02)$. Using the best-guess genotype definition, both genotypes will be imputed as $G = 0$ and hence a discordance of one point, but the prediction *(a)* is "weaker" since it has a lower best-guess likelihood ($0.56 < 0.7$). In that sense, the prediction *(a)* should be considered as less significant than the *(b)* one even if both are wrong. Therefore, we introduce the cross-entropy metrics χ as a divergence measure of the predicted genotype distribution. The cross-entropy we propose is defined as in equation 17 at the j-th marker for N individuals imputed.

$$\chi_j = \frac{\sum_{i=1}^{N} \left(- \sum_{g=0}^{2} Pr(G_{ij} = g)\ log(\mathcal{L}_{ijg}) \right)}{N} \qquad (17)$$

where $\mathcal{L}_{ijg}$ is the genotype likelihood (or posterior imputed genotype probability) for the genotype state g at the j-th marker for the i-th individual. For low-probability genotypes, we used a cut-off of $log(10^{-5})$ if the genotype probability was less than 10^{-5}.

Computational tools

Due to their computational costs, imputation algorithms were run on compute servers. The computing resources were provided by SNIC through Uppsala Multidisciplinary Center for Advanced Computational Science. This infrastructure provides nodes (compute servers) of two 10-core Xeon E5-2630 V4 or two 8-core Xeon E5-2660 processors running at 2.2 GHz, with 128 to 512 GB memory.

Results

Genotype distribution before imputation

The LD and HD marker sets built for the experiment both contain SNPs in the whole allelic frequency range but the markers are unevenly distributed over this range. Table 1 provides further details about the uneven distribution. We aim to analyze the uncommon variants at a finer scale and visualize their joint response to pooling and imputation. Therefore, the bins chosen are tighter towards

the least MAF values and the boundaries set to $[0.0, 0.02, 0.04, 0.06, 0.1, 0.2, 0.4, 0.5]$ for the intervals.

The most rare variants ($MAF < 2\%$) represent a substantial share of the studied SNPs with 520 markers in the LD dataset and 12775 in the HD dataset. One should note that even denser chips, or the full marker set of called SNPs in the 1KGP dataset, are even more extreme in this regard.

Table 2 shows the proportion of assayed and determined genotypes before imputation in the LDHD scenario and in the pooled HD scenario.

Already at the preimputation stage, the pooling mechanism proves to be particularly efficient for capturing the most rare variants ($MAF < 2\%$) with 98.1% determined genotypes before imputation. In the LDHD scenario, only 0.41% of the genotypes are assayed in the most rare variants before imputation. In total, there are 67.7% unassayed genotypes before imputation in the LDHD scenario and 44% in the pooled HD scenario. The proportions of known genotypes however varies depending on the MAF.

Whilst the proportion of known genotypes seems to augment as the MAF increases in the LDHD scenario, a negative correlation between the known data rate and the MAF is noticed in the pooling case. Indeed, the proportion of fully decoded genotypes is less than 10% for MAF exceeding 30%. Such markers are common variants. Since both alleles have roughly the same frequency in the population, heterozygotes and mixed genotypes within pools will be far more common as on Figure 3b, or with even more carriers of the minor allele in the block. To summarize, there is a significant correlation between true genotypes and the probability of the genotype being decoded, and that correlation is further dependent on the MAF of the marker. The proportions of known genotypes before imputation per MAF-bin in the LDHD scenario is actually fixed by the choice made for the LD map. In other words, changing the LD map will modify the distribution of known markers. In the pooled HD scenario, the proportions mostly depend on the MAF of every marker and the HD map chosen has a limited impact on the distributions of known markers per MAF-bin.

The distribution of heterozygous and homozygous genotypes obtained in each MAF-bin from both data deletion (LDHD scenario) and pooling simulation (pooled HD scenario) are presented on Figure 4. To the difference of the LDHD data set, the pooled HD one let some markers being half-genotyped in that sense one out of the two alleles can be determined before imputation. For example in the markers having a MAF less than 2%, in addition to the large share of exactly determined genotypes ($GT = M/M$), most of the indeterminate genotypes are yet half-known ($GT = ./m$). The pooling process never fully decodes the true heterozygous genotypes, hence the proportion of unassayed genotypes will be large in common markers. Only the homozygous genotypes can be determined from pooling with our design. For the LDHD scenario, the heterozygous genotypes that are naturally present in the study population at the markers on the LD map are observed in the preimputation data set. These observations highlight the very different compositions of the LDHD and the pooled HD data sets before impu-

tation. On the whole, the distribution of the observed and assayed genotypes in the population is unevenly affected by pooling and depends on the MAF.

Genotyping accuracy after imputation

Table 2 also shows the proportion of genotypes that are imputed exactly to the true one. Table 3 provides a closer insight into the imputation performance of Beagle and *Prophaser* in terms of exact matches for the genotypes at undecoded markers only in the pooled HD scenario.

Figure 5 presents the genotyping accuracy for imputed markers in both the LDHD and the pooled HD scenarios. The concordance and cross-entropy metrics are presented for comparison. Preliminary experiments (unpublished results) showed that the strategy of using pooling patterns-adapted GL values instead of uninformed ones improves the imputation accuracy.

In the LDHD scenario, Beagle shows as expected very good performance with an average concordance of 98.5% and low entropy (0.05). The performance is stable across the MAF range on average, though there is a larger variation in accuracy for more common variants. In the pooled HD scenario, while the overall proportion of missing data is lower, Beagle's performance drops substantially (79.6% concordance on average and a cross-entropy score of 3.43). The wide envelope for the cross-entropy also indicates that the amplitude of prediction errors on the marker level varies widely in the pooled HD scenario. The haplotype-clustering model seems to struggle with the unusual genetic structure of pooled data.

Prophaser achieves higher accuracy than Beagle in the LDHD scenario, showing nearly 99% average concordance and 0.04 cross-entropy score. As for Beagle, the concordance is stable but more spread for higher MAF (less accurate). In the pooled HD scenario, *Prophaser* clearly outperforms Beagle for imputing the undecoded genotypes by maintaining an average concordance of 92.6% and a cross-entropy score of 0.31. The quantile envelopes for both metrics demonstrate that *Prophaser* gives stable performance for most markers, while the results for Beagle show a much greater variation. It is naturally important not only that the average concordance or entropy is good, but that any single imputed marker of possible importance is trustworthy. Despite the weaker performance on the pooled HD data compared to the LDHD scenario, *Prophaser* proves the ability to use the uncertain decoded genotypes from pooling for successful imputation.

Table 2 gives a detailed view of the number and proportions per MAF bin of exact genotypes, both in the LDHD and in the pooled HD data sets, before and after imputation. It reveals the benefit that is obtained from gneotyping pooled samples for the variants having a MAF less than 2%. *Prophaser* indeed succeeds in raising the proportion of exactly matched genotypes after imputation by 0.3%. This gain is not negligible given the very low frequency of the variations in such markers.

Computational performance

For Beagle, the compute server (node) was two 10-core processors running at 2.2 GHz with 128 GB mem-

ory. For *Prophaser* the node resources were two 8-core processors running at 2.2 GHz, with 128 GB memory. Computation times per study sample were about 7 minutes for Beagle respectively 6 hours 40 minutes for *Prophaser*, and the memory requirements for each sample consumed about 2.2 GiB (resp. 35 GiB) of memory. In the classical scenario, it is even possible to run Beagle on all study samples together in about 20 minutes using ca. 12 GiB memory and to get the same accuracy results. Hence, accordingly to the results found in other studies, Beagle demonstrates an excellent computational efficiency in imputing large data sets. *Prophaser* is on the contrary computationally very expensive, as mentioned to be a drawback in the literature with similar algorithms. However, we have not yet optimized the performance of our implementation.

Discussion

As we could expect, pooling enables efficient identification of carriers of rare variants within the population, but yields high missing data rates for more common variants. Several studies have indeed shown that the distribution of the undecoded items is hypergeometrical and correlated to the minor allele frequency [2, 11]. In the case of low-MAF SNPs, the pools are mostly homogeneous and homozygous, or contain at most one rare variant carrier as on Figure 3a. Blocks as on Figure 3b are unlikely to be observed for these SNPs. Indeed, with respect to HWE in a random mating population, rare variant carriers would almost exclusively be heterozygotes. The pooling design used in this study guarantees a theoretical perfect decodability of the samples genotype if at most one sample in the block is carrying the minor allele ($d_0 = 1$, calculated as in the DNA Sudoku [9]). The results presented in Table 2 comply with the theoretical limiting decoding power. The upper bound for MAF with high certainty of decodability is calculated as $\delta_{MAF} = \frac{d_0 \times G_1}{2 \times n_B} = \frac{1 \times 1}{2 \times 16} \approx 3.1\%$. Our results for the pooled HD scenario show that the number of known markers before imputation drops when the MAF is larger than 2%, and decreases even more when the MAF is greater then 4%. SNPs having a MAF below this boundary of 3.1% are expected to be nearly fully assayed in the study population or decoded as rare variant carriers, such that pooling provides a useful complementary process to imputation for achieving accurate genotyping of rare variants that are usually more difficult to impute. Other pooling designs can be explored for increasing the decoding power. With a given pooling design, hybrid procedures consisting of imputation from a fully a assayed LD set and a pooled HD set are further alternatives to consider. Similarly to the representation [22] suggested for evaluating the pooling design performance for clone-based haplotyping, we think that quantifying the genotyping effort in relation to the decoding rate and to the MAF as a performance ratio of pooled genotyping could be a future criterion for choosing a pooling design depending on the markers data set and its characteristics. Considering the very good performance of imputation in a LDHD scenario and the complementary nature of a pooled scenario that excel at capturing the rare variants, one could also imagine a more sparse pool-

ing scheme, such as a 5x5 design, with a dense chip, augmented by full LD testing of some or all individuals. This would give the imputation process a clear scaffold to start out from, together with very accurate information for carriers of rare variants. It also opens perspectives for genotyping on even denser chips targeting very rare variants ($MAF < 0.02$) without large increase in laboratory costs.

We have presented algorithms that locally adapt the genotype frequencies to every pooling block, but we believe further research could be conducted for improving the GL estimates. In our context, the resulting probabilities after decoding should be evaluated in terms of to what extent they improve the imputation results.

Imputation on pooled data yielded notably different performance depending on the imputation method family used. The clustering model as implemented in Beagle seems to suffer from the pooled structure in the data. We think the clusters built collapse together haplotypes that are substantially different, but can have superficial similarities after the decoding of pooled data. This fact also results in the decoded population looking systematically different from the reference population. We showed with the *Prophaser* algorithm that the coalescence assumption supports an imputation model that delivers high accuracy in pooled genotype reconstruction, at a computational cost. This is consistent with other studies [29, 50] that have found the coalescent methods to be robust towards unknown genetic population structures. From the perspective of the method, the systematic bias introduced by the decoding is similar to unknown population structure. By using all the reference haplotypes from the panel during imputation, *Prophaser* might overcome the pitfall of sensitivity to deviant genetic structure as mentioned in [3]. As a result, allele frequencies assessed in the study population are no longer consistent with the effective frequencies differences expressing genetic variation found in the reference panel. While the reason presented in that paper is chip quality, we face similar biased structural heterogeneity issues with pooled data.

This initial investigation of the performance of pooling and imputation as a combined way to recover genotypes is purely based on simulations, in the absence of genotyping errors. In quality control data from chip manufacturers, detection power for alleles can be found on a per-SNP level. Actual detection performance could be influenced by the amount of DNA contributed from various samples within a pool. Our intention is to continue to explore our approach on actual assays, in partnerships where cost-effective genotyping on a massive scale is a real concern.

It should be noted that our probabilistic decoding method could be modified to account for genotyping errors, and that it will be crucial to consider the overall effect of errors in decoding individual SNPs and how those errors in turn affect the ability of the imputation methods to properly reconstruct the haplotype mosaic, since it is the accuracy of that mosaic of reference haplotypes that in turn will influence imputation performance.

Conclusions

The findings of this study suggest that pooling can be jointly used with imputation methods for achieving accurate SNPs at high density while reducing the actual number of genotyping procedures done on microarrays. However, the atypical structure introduced by pooling in the genotype data requires specific attention and processing for ensuring the best imputation performance possible.

Overall, pooling impacts the allelic and genotypic distributions, and introduces a specific structure in the genetic data which does not reflect their natural distribution. We have described a statistical framework that formalizes pooling as a mathematical transformation of the genotype data, and we have proposed in this framework an algorithm for estimating the latent values of undecoded genotypes. Lastly, thanks to a simulation on real human data, we have shown that a coalescence-based imputation method performs well on pooled data, and that informing imputation with estimates of the latent missing genotypes improves the prediction accuracy. We also presented an implementation (*Prophaser*) of this imputation method for pooled genotype data. Overall, this study provides a first prototype for the computational aspect of a SNP genotyping strategy at a reduced cost by halving the number of microarrays needed compared to a full sample-wise genotyping.

List of abbreviations

1KGP: 1000 Genomes Project

AAF: Alternate Allele Frequency

EM: Expectation-Maximization

GL: Genotype Likelihood

GP: Genotype Probability

GT: True Genotype

HD: High Density

HMM: Hidden Markov Models

LD: Low Density

MAF: Minor Allele Frequency

MLE: Maximum Likelihood Estimation

ML-II: type II Likelihood, Marginal Likelihood

MMLE: Maximum Marginal Likelihood Estimation

NGT: Nonadaptive Group Testing

NORB: Nonadaptive Overlapping Repeated Block

PNL: reference panel

SNP: Single Nucleotide Polymorphism

STD: Shifted Transversal Design

STU: study population

Declarations

Ethics approval and consent to participate

The publicly available 1000 Genomes dataset was approved by the 1000 Genomes Samples group, the ELSI subgroup, and the P3G-IPAC consortium, as stated on https://www.internationalgenome.org/sample_collection_principles/.

Consent for publication

Not applicable.

Availability of data and materials

The dataset(s) supporting the conclusions of this article (are) available in the data subdirectory of *genotypooler* repository, https://github.com/camcl/genotypooler/data.

These datasets are created from publicly available 1000 Genomes dataset [45].

Prophaser code can be found at https://github.com/kausmees/prophaser.

Competing interests

The authors declare that they have no competing interests.

Funding

Research project funded by Formas, The Swedish government research council for sustainable development. Grant nr 2017-00453. Cost-effective genotyping in plant and animal breeding using computational analysis of pooled samples.

Author's contributions

CN conceived the study. KA developed the initial version of *Prophaser* and provided advice on its use. CC developed the pipeline for simulating pooling and designed the experiment in collaboration with CN. CC conducted all analysis and drafted the manuscript. All authors edited the manuscript and contributed to the conclusions.

Acknowledgements

The computing resources were provided by SNIC through Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) under Projects SNIC 2019/8-216 and 2020/5-455.

References

1. Fernández, M.E., Goszczynski, D.E., Lirón, J.P., Villegas-Castagnasso, E.E., Carino, M.H., Rogberg-Muñoz, M.V.R.A., Posik, D.M., Peral-García, P., Giovambattista, G.: Comparison of the effectiveness of microsatellites and snp panels for genetic identification, traceability and assessment of parentage in an inbred angus herd. Genetics and Molecular Biology **36**(2), 185–191 (2013)
2. Cao, C., Li, C., Huang, Z., Ma, X., Sun, X.: Identifying rare variants with optimal depth of coverage and cost-effective overlapping pool sequencing. Genetic Epidemiology **37**(8), 820–830 (2013)
3. Howie, B., Marchini, J.: Genotype imputation for genome-wide association studies. Nature Reviews Genetics **11** (2010)
4. Sung, Y.J., Gu, C.C., Tiwari, H.K., Arnett, D.K., Broeckel, U., Rao, D.C.: Genotype imputation for african americans using data from hapmap phase ii versus 1000 genomes projects. Genetic Epidemiology **36**(5), 508–516 (2012)
5. Chanda, P., Li, N.Y.M., *et al.*: Haplotype variation and genotype imputation in african populations. Human Genetics **57**, 411–421 (2012)
6. Saad, M., Wijsman, E.M.: Combining family- and population-based imputation data for association analysis of rare and common variants in large pedigrees. Genetic Epidemiology **38**(7), 579–590 (2014)
7. Mitt, M., Kals, M., Pärn, K., Gabriel, S.B., Lander, E.S., Palotie, A., Ripatti, S., Morris, A.P., Metspalu, A., Esko, T., Mägi, R., Palta, P.: Improved imputation accuracy of rare and low-frequency variants using population-specific high-coverage wgs-based imputation reference panel. European Journal of Human Genetics **25**, 869–876 (2017)
8. Macula, A.J.: Error-correcting nonadaptive group testing with de-disjunct matrices. Discrete Applied Mathematics **80**, 217–222 (1997)
9. Y. Erlich, A.G. K. Chang, *et al.*: Dna sudoku—harnessing high-throughput sequencing for multiplexed specimen analysis. Genome Research **19**, 1243–1253 (2009)
10. Hormozdiari, F., et al.: Efficient genotyping of individuals using overlapping pool sequencing and imputation. 2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), 1023–1027 (2012)
11. Cao, C., Li, C., Sun, X.: Quantitative group testing-based overlapping pool sequencing to identify rare variant carriers. BMC Bioinformatics **15**(195) (2014)
12. Lonardi, S., et al.: Combinatorial pooling enables selective sequencing of the barley gene space. PLoS Computational Biology **9**(4) (2013)
13. Technow, F., Gerke, J.: Parent-progeny imputation from pooled samples for cost-efficient genotyping in plant breeding. PLoS ONE **12**(12) (2017)
14. Cao, C., Sun, X.: Accurate estimation of haplotype frequency from pooled sequencing data and cost-effective identification of rare haplotype carriers by overlapping pool sequencing. Bioinformatics **31**(4) (2015)
15. Zhao, Y., Wang, S.: Optimal dna pooling-based two-stage designs in case-control association studies. Human Heredity **67**(1), 46–56 (2008)
16. Ji, F., Finch, S.J., Haynes, C., Mendell, N.R., Gordon, D.: Incorporation of genetic model parameters for cost-effective designs of genetic association studies using dna pooling. BMC Genomics **8**(238) (2007)
17. Sham, P., Bader, J., Craig, I., *et al.*: Dna pooling: a tool for large-scale association studies. Nat Rev Genet **3**, 862–871 (2002)
18. Iliadis, A., Anastassiou, D., Wang, X.: Fast and accurate haplotype frequency estimation for large haplotype vectors from pooled dna data. BMC Genomics **13**(94) (2012)
19. Alexandre, P.A., Porto-Neto, L.R., Karaman, E., Lehnert, S.A., Reverter, A.: Pooled genotyping strategies for the rapid construction of genomic reference populations. Journal of Animal Science **97**(12), 4761–4769 (2019)
20. Zhang, P., Krzakala, F., Mezard, M., Zdeborova, L.: Non-adaptive pooling strategies for detection of rare faulty items. Lecture Notes in Computer Science and Workshop on Algorithms and Data Structures 2005: Algorithms and Data Structures (2013)
21. Prabhu, S., Pe'er, I.: Overlapping pools for high-throughput targeted resequencing. Genome Research **19**, 12541261 (2009)

22. Li, C., Cao, C., Tu, J., Sun, X.: An accurate clone-based haplotyping method by overlapping pool sequencing. Nucleic Acids Research **44**(12) (2016)
23. He, D., et al.: Genotyping common and rare variation using overlapping pool sequencing. BMC Bioinformatics **12**(6) (2011)
24. Thierry-Mieg, N.: A new pooling strategy for high-throughput screening: the shifted transversal design. BMC Bioinformatics **7**(28) (2006)
25. Ngo, H.Q., Du, D.-Z.: A survey on combinatorial group testing algorithms with applications to dna library screening. DIMACS Series in Discrete Mathematics and Theoretical Computer Science **55** (2000). doi:10.1090/dimacs/055/13
26. Chen, H.-B., Wang, F.K.: A survey on nonadaptive group testing algorithms through the angle of decoding. Journal of Combinatorial Optimization **15**, 49–59 (2008)
27. Nothnagel, M., Ellinghaus, D., Schreiber, S., Krawczak, M., Frankeg, A.: A comprehensive evaluation of snp genotype imputation. Human Genetics **125**, 163–171 (2009)
28. Pei, Y.-F., Li, J., Zhang, L., Papasian, C.J., Deng, H.-W.: Analyses and comparison of accuracy of different genotype imputation methods. PLoS ONE **3**(10) (2008)
29. Sung, Y.J., Wang, L., Rankinen, T., Bouchard, C., Rao, D.C.: Performance of genotype imputations using data from the 1000 genomes project. Human Heredity **73**, 18–25 (2012)
30. Pook, T., Mayer, M., Geibel, J., Weigend, S., Cavero, D., Schoen, C.C., Simianer, H.: Improving imputation quality in beagle for crop and livestock data. Genes Genomes Genetics **98**, 116–126 (2019)
31. Nyine, M., Wang, S., Kiani, K., Jordan, K., Liu, S., Byrne, P., Haley, S., Baenziger, S., Chao, S., Bowden, R., Akhunov, E.: Genotype imputation in winter wheat using first-generation haplotype map snps improves genome-wide association mapping and genomic prediction of traits. Genes Genomes Genetics **9**, 125–133 (2019)
32. Browning, S.R., Browning, B.L.: Haplotype phasing: existing methods and new developments. Nature Reviews Genetics **12** (2011)
33. Browning, S.R.: Missing data imputation and haplotype phase inference for genome-wide association studies. The American Journal of Human Genetics **124**(5), 439–450 (2008)
34. Zhao1, Z., Timofeev, N., Hartley, S.W., Chui, D.H., Fucharoen, S., Perls, T.T., Steinberg, M.H., Baldwin, C.T., Sebastiani, P.: Imputation of missing genotypes: an empirical evaluation of impute. BMC Genetics **9**(85) (2008)
35. Li, Y., Wille, C.J., Ding, J., Scheet, P., Abecasis, G.R.: Mach: Using sequence and genotype data to estimate haplotypes and unobserved genotypes. Genetic Epidemiology **34**(8), 816–834 (2010)
36. Ausmees, K., Nettelblad, C.: Achieving improved accuracy for imputation of ancient DNA. bioRxiv (2022). doi:10.1101/2022.04.26.489533. https://www.biorxiv.org/content/early/2022/04/27/2022.04.26.489533.full.pdf
37. Howie, B., Donelly, P., Marchini, J.: A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. PLoS Genetics **5**(6) (2009)
38. Browning, S.R., Browning, B.L.: 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 **81**, 1084–1097 (2007)
39. Browning, B.L., Browning, S.R.: A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. The American Journal of Human Genetics **84**, 210–223 (2009)
40. Browning, B.L., Browning, S.R.: Genotype imputation with millions of reference samples. The American Journal of Human Genetics **98**, 116–126 (2016)
41. Browning, B.L., Zhou, Y., Browning, S.R.: A one-penny imputed genome from next-generation reference panels. The American Journal of Human Genetics **103**(3), 338–348 (2018)
42. Deloukas, P., Matthews, L., Ashurst, J.: The dna sequence and comparative analysis of human chromosome 20. Nature **414**, 865–871 (2001)
43. Conrad, D.F., Jakobsson, M., Coop, G., Wen, X., Wall, J.D., Rosenberg, N.A., Pritchard, J.K.: A worldwide survey of haplotype variation and linkage disequilibrium in the human genome. Nature Genetics **38**(11), 1251–81 (2006)
44. Spiliopoulou, A., Colombo, M., Orchard, P., Agakov, F., McKeigue, P.: Geneimp: Fast imputation to large reference panels using genotype likelihoods from ultralow coverage sequencing. Genetics **206**, 91–104 (2017)
45. Sudmant, P., Rausch, T., Gardner, E., *et al.*: An integrated map of structural variation in 2,504 human genomes. Nature **526**, 75–81 (2015)
46. Howie, B., Marchini, J., Stephens, M.: Genotype imputation with thousands of genomes. Genes Genomes Genetics **1** (2011)
47. Marchini, J., Howie, B., Myers, S., McVean, G., Donnelly, P.: A new multipoint method for genome-wide association studies by imputation of genotypes. Nature Genetics **39**, 906–913 (2007)
48. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from

incomplete data via the em algorithm. Journal of the Royal Statistical Society **39**(1), 1–22 (1977)

49. Tarpey, T., Flury, B.: Self-consistency: A fundamental concept in statistics. Statistical Science **11**(3), 229–243 (1996)
50. Huang, L., Jakobsson, M., Pemberton, T.J., Ibrahim, M., Nyambo, T., Omar, S., Pritchard, J.K., Tishkoff, S.A., Rosenberg, N.A.: Haplotype variation and genotype imputation in african populations. Genetic Epidemiology **35**(8), 766–780 (2011)

Figures

Tables

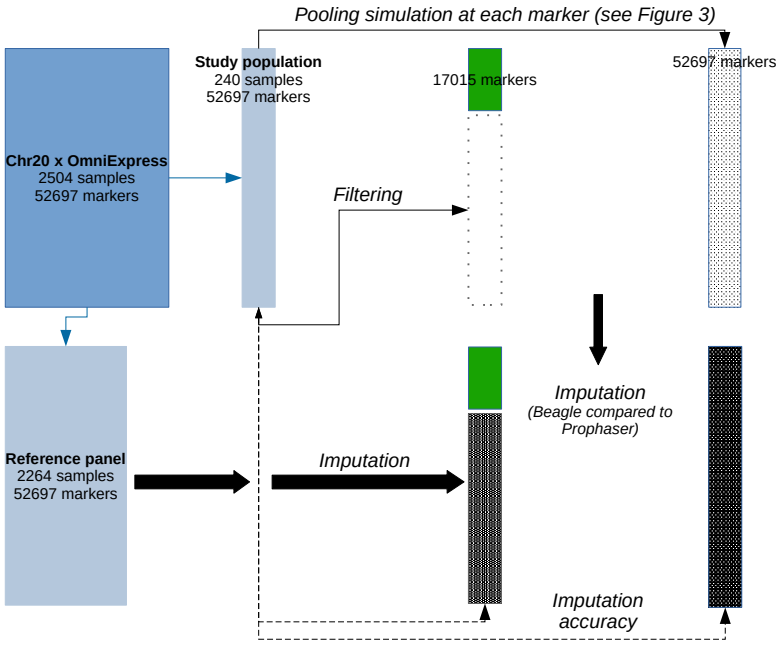
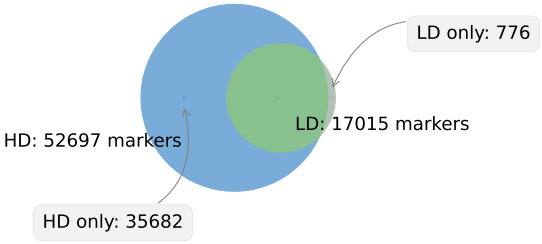


Figure 1: Experimental steps for creating the data sets in the pooling and classical imputation scenarios

The original data set "Chr20 x OmniExpress" consists of the genotype data of 2,504 samples at 52,697 SNPs. The set of markers is created by intersecting the variants present on both bead-chips from the Illumina manufacturer and the data for the chromosome 20 in the 1KGP.
The original data set is randomly split into a reference panel and a study population. In the LDHD scenario, all markers in the HD data set that are not present in the LD data set are filtered out in the study population. In the pooled HD scenario, the study samples are first assigned to blocks and pools, second the pools are genotyped at all markers in the HD data set, and last the genotype of each sample is decoded from the pools at every marker. See Figure 3 for an example of the simulation steps in 1 block at 1 marker. The imputation step is performed in both scenarios from the reference panel, with Beagle on the one hand and *Prophaser* on the other hand. The genotyping accuracy in each scenario is computed by comparing the imputed genotypes with the true ones in the original data set for the study population.

(a) Venn diagram of the makers composition on the Illumina bead-chips Omni2.5 - 8 (HD) and Omniexpress - 24 (LD)



(b) Populations data sets used for comparison in two imputation scenarii

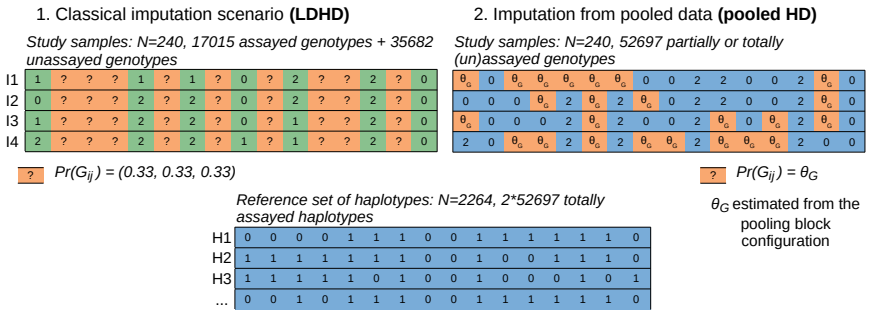


Figure 2: Markers data sets used for the study population in the pooling and classical imputation scenarios

Figure 2a: LD and HD markers data sets from intersecting Illumina bead-chips x 1KGP chromosome 20.
Figure 2b: Missing genotypes repartition and values in a classical imputation scenario (1.), and in an imputation scenario from pooled data (2.) where the genotypes probabilities θ_G are estimated from the configurations of the pooling blocks.

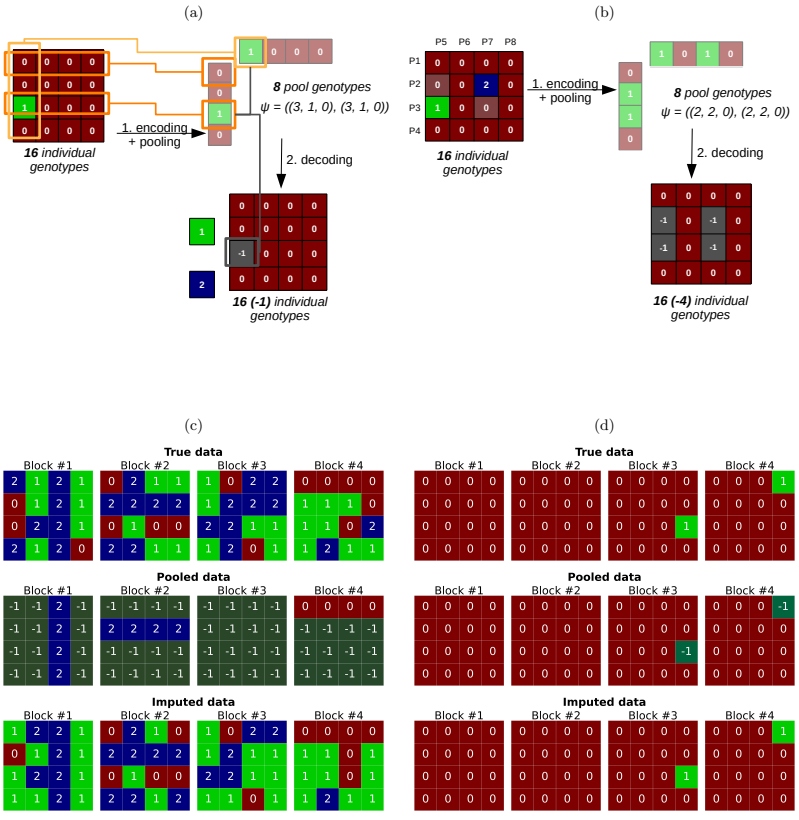
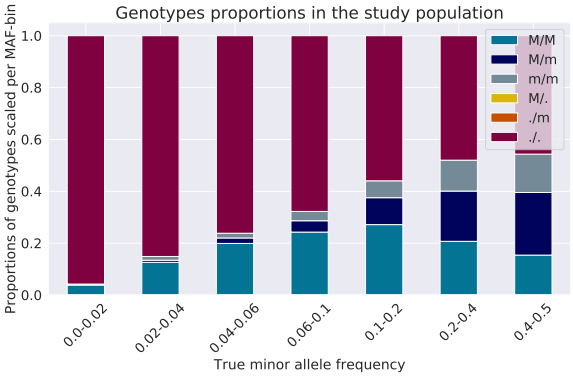


Figure 3: Examples of genotype pooling simulation at the block level

Subfigure 3a: Configuration with 1 sample carrying the minor allele. This carrier is identified after pooling, but not if it has a heterozygous (1) or a minor homozygous (2) genotype.
Subfigure 3b: Configuration with 2 samples carrying the minor allele. At least 2 of the 4 samples highlighted in grey are minor allele carriers, but the genotypes of these 4 samples are indeterminate.
The first step is encoding and pooling. Encoding assigns every sample to a pool and defines its pool coordinates. For instance in subfigure 3a, the sample at the top-left corner of the matrix has coordinates $(1, 5)$. Pooling computes the genotype of a pool as if its would tested on a SNP-chip. Pool 5 (P5, most left) has genotype 1: both alleles 0 and 1 are detected among the samples. Pool 1 has genotype 0 because only the allele 0 is detected. The decoding step infers the pooled genotype of each sample from the genotypes of its coordinates. The genotype can be -1 i.e. indeterminate when both coordinates have genotype 1, or fully determined else. In subfigure 3a, the sample with coordinates $(3, 5)$ carries the alternate allele, but there can be 1 or 2 copies of it.
ψ is the observed pooling pattern that results from grouped genotyping, given as the number of row- and column-pools having the genotypes $(0, 1, 2)$. In the example 3a, there are 3 row-pools having genotype 0, 1 row-pool having genotype 1 and 0 having genotype 2, likewise for the column-pools.

Subfigures 3c and 3d: Simulation example of genotype pooling and imputation outcomes for markers from the 1KGP data (chromosome 20). The genotypes are represented as unphased GT. From top to bottom: true genotype data, pooled genotypes, imputed genotypes.
Subfigure 3c: SNP 20:264365, $MAF = 0.4625$.
Subfigure 3d: SNP 20:62915126, $MAF = 0.00625$.

(a) Decoded and missing genotypes in LDHD data



(b) Decoded and missing genotypes in pooled HD data

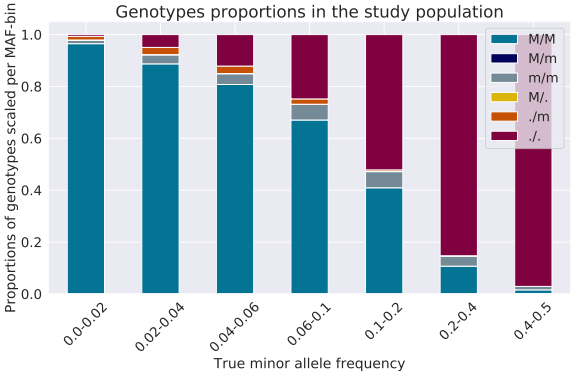


Figure 4: Decoded and missing genotypes in data for both imputation scenarios

The minor and major alleles are denoted m and M. For simplicity, the simulated decoded genotypes from pooling are represented in GT format. We remind adaptive GL are provided later in the experiment for running imputation on data informed with the pooling outcomes. Half-decoded (GT = $M/.$ or $./m$) and not decoded (GT = $./.$) genotypes are considered as missing data. The relative genotypes proportions are scaled in $[0, 1]$ within each bin.
Subfigure 4a: The markers only in the LD data set are fully assayed, all other markers have been deleted.
Subfigure 4b: True heterozygous genotypes (dark blue) are never fully decoded, whereas the rare variants are almost all fully decoded or at least one of the two alleles is determined.

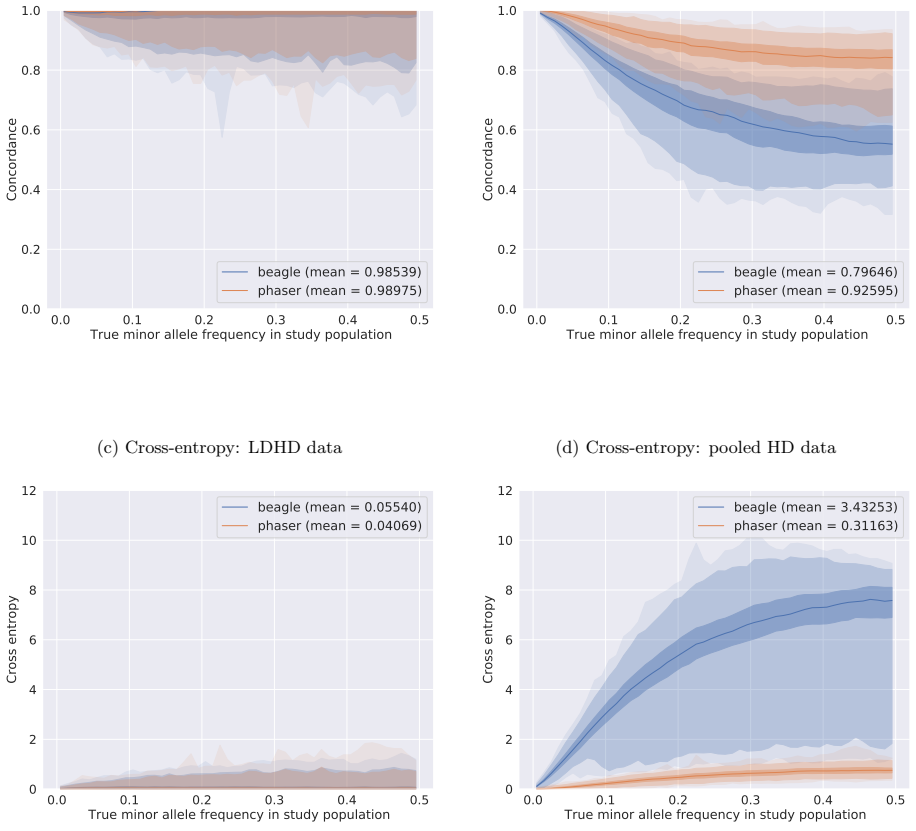


Figure 5: Genotypes imputation accuracy in a classical and a pooled scenario

Subfigures 5a and 5b: concordance (based on best-guess genotype)
Subfigures 5c and 5d: cross-entropy (based on posterior genotypes probability) metrics. All markers from the HD map have been used for computing the metrics (52,697 markers).
Beagle (labeled as "beagle") performance is in blue, and Prophaser (labeled as "phaser") in orange.
The central line is the median and the shadowed areas delimit the percentiles 0.0, 0.01, 0.25, 0.75, 0.99, 1.0. The x-axis was built from 0.05-long MAF bins within which each marker concordance score was computed as the mean score of the 500 previous and 500 next markers sorted per ascending MAF.

MAF	0.00-0.02	0.02-0.04	0.04-0.06	0.06-0.10	0.10-0.20	0.20-0.40	0.40-0.50	Total
LD map (counts)	520	779	673	1537	3969	6561	2976	17015
HD map (counts)	12775	5235	2823	4766	9009	12613	5476	52697
LD map (%)	0.987	1.478	1.277	2.917	7.532	12.450	5.647	32.288
HD map (%)	24.242	9.934	5.357	9.044	17.096	23.935	10.392	100

Table 1: Markers counts and proportions on the LD and the HD maps per MAF bin

The counts are given in the two first rows of the table, the proportions in the two last ones. The proportions are given relatively to the total number of SNPs on the HD map. The HD map is on the whole 3 times denser than the LD map but the density is not uniformly increased over the MAF bins. Almost 25% of the markers on the HD map are very rare variants ($MAF < 0.02$), that is 25 times denser than on the LD map where they represent less than 1% of the markers.

MAF		0.00-0.02	0.02-0.04	0.04-0.06	0.06-0.10	0.10-0.20	0.20-0.40	0.40-0.50
Scenario: LD + HD								
Number before imputation		520.000	779.000	673.000	1537.000	3969.000	6561.000	2976.000
Number after imputation	Beagle	12699.362	5167.613	2776.687	4673.658	8804.892	12301.371	5337.921
	Phaser	12727.142	5193.438	2793.221	4705.346	8870.104	12396.258	5379.408
Proportion before imputation		0.041	0.149	0.238	0.322	0.441	0.520	0.543
Proportion after imputation	Beagle	0.994	0.987	0.984	0.981	0.977	0.975	0.975
	Phaser	**0.996**	0.992	0.989	0.987	0.985	0.983	0.982
Scenario: pooled HD								
Number before imputation		12534.608	4826.542	2396.671	3481.896	4249.592	1853.529	159.575
Number after imputation	Beagle	12565.650	4892.246	2478.292	3778.296	5637.525	5407.479	1941.162
	Phaser	12755.854	5184.621	2758.079	4532.467	7964.742	9858.467	4012.725
Proportion before imputation		0.981	0.922	0.849	0.731	0.472	0.147	0.029
Proportion after imputation	Beagle	0.984	0.935	0.878	0.793	0.626	0.429	0.354
	Phaser	**0.999**	0.990	0.977	0.951	0.884	0.782	0.733

Table 2: Exact genotypes in markers per data MAF bin

The number of markers is given as the average over all samples in the study population per bin. The proportion of markers is given relatively to the number of markers per bin. To the difference of concordance, only full matches with the true genotype are counted, not half-matches.
For the LD + HD scenario, the number of exact genotypes before imputation is equal to the number of variants on the LD map.
For the pooled HD scenario, the number of exact genotypes before imputation is equal to the average number of genotypes that are fully determined after pooling simulation.
Simulating pooling followed by imputation with Phaser yields a gain in accuracy for the very rare variants ($MAF < 0.02$) which are almost all exactly genotyped. This gain is not negligible given the low occurence of these variations.

Additional Files

Additional file — Estimating genotype probabilities in pooled blocks with marginal likelihoods, self-consistency and heterozygotes degeneracy

This file provides further details about the self-consistent procedure, based on the Expectation-Maximization method, that we implemented for computing genotype probabilities at undecoded items in pooled blocks.

MAF	0.00-0.02	0.02-0.04	0.04-0.06	0.06-0.10	0.10-0.20	0.20-0.40	0.40-0.50
Phaser	**0.932700**	0.886214	0.849634	0.820339	0.783430	0.745528	0.724745
Beagle	0.124773	0.156686	0.187206	0.227121	0.287044	0.329487	0.334919

Table 3: Proportion of exact genotypes after imputation for indeterminate data in the pooled HD scenario per data MAF bin

This table focuses on the genotypes that are indeterminate after the pooling simulation. The proportion is calculated for these markers only and relatively to the number of markers in the bin.
For the very rare variants ($MAF < 0.02$), the indeterminate genotypes are the rare allele carriers. Phaser succeeds in imputing exactly most of them from the provided prior genotype probabilities estimates.

Paper II

Consistency study of a reconstructed genotype probability distribution via clustered bootstrapping in NORB pooling blocks

Camille Clouard[1] and Carl Nettelblad[1]

[1]Division of Scientific Computing, Department of Information Technology, Uppsala University

Abstract

For applications with biallelic genetic markers, group testing techniques, synonymous to pooling techniques, are usually applied for decreasing the cost of large-scale testing as e.g. when detecting carriers of rare genetic variants. In some configurations, the results of the grouped tests cannot be decoded and the pooled items are missing. Inference of these missing items can be performed with specific statistical methods that are for example related to the Expectation-Maximization algorithm. Pooling has also been applied for determining the genotype of markers in large populations. The particularity of full genotype data for diploid organisms in the context of group testing are the ternary outcomes (two homozygous genotypes and one heterozygous), as well as the distribution of these three outcomes in a population, which is often ruled by the Hardy-Weinberg Equilibrium and depends on the allele frequency in such situation. When using a nonoverlapping repeated block pooling design, the missing items are only observed in particular arrangements. Overall, a data set of pooled genotypes can be described as an inference problem in Missing Not At Random data with nonmonotone missingness patterns. This study presents a preliminary investigation of the consistency of various iterative methods estimating the most likely genotype probabilities of the missing items in pooled data. We use the Kullback-Leibler divergence and the L2 distance between the genotype distribution computed from our estimates and a simulated empirical distribution as a measure of the distributional consistency.

Background

Purposes of group testing

Pooling is a group testing technique addressing how to confidently identify a category of items, called 'defectives', in a population, with as few tests as possible. Group testing has found numerous applications with DNA data for e.g. the purpose of large-scale sequencing or genotyping at reduced cost.

A pooling algorithm for genetic data

In an other study [1], we have explored the usage of a Non Overlapping Repeated Block (NORB) design for simulation pooling on genotype data, similar to the design suggested by Erlich et al. [2]. Figure 1(a) presents the principle of such a pooling experiment.

The NORB procedure divides the population into B equally sized blocks of n_B individuals. In the encoding step of pooling, every block systematically defines how many pools are formed from the n_B items and the mapping of individual items to different pools. The genotype of a pool is determined by the alleles that are detected among the pool members at the testing step. In the decoding step, the algorithm attempts to retrieve the genotype of any item based on the genotypes of the intersecting pools. In some cases, the decoding fails to confidently identify the genotype of an item and returns it as missing.

Erlich et al. [2] originally presented a NORB algorithm for decoding the genotypes into a binary response, that is, whether any genotype is a carrier of a rare variant or not. In our research, we extend the proposed decoding to a more general case of a ternary outcome, determining if the genotype is homozygous for the reference allele, heterozygous, or homozygous for the alternate allele. We suggest for this purpose an algorithm based on the Expectation-Maximization (EM) method that models all possible pooling configurations and computes the most likely genotype of every item involved in each configuration. The items that cannot not be confidently identified are assigned to a genotype which we call 'adaptive'. Such a genotype is a local consistent block-wise estimate of the genotype probabilities (GP) for these specific items. Figure 1(b) shows one example of our block-adaptive decoding algorithm, where a block configuration is identified by

its pooling pattern ψ.

In this study, we investigate the consistency of our adaptive estimates compared to the pre-pooling genotype values.

Probabilistic formulation of the NORB pooling problem

Data sets

Representation of the genotype data

We model the genotype data at any marker for a sample i as a probability simplex $[p_{0i}, p_{1i}, p_{2i}]^\top$, which stand, in this order, for the probability of the genotype being a homozygote for the reference allele, a heterozygote, and a homozygote for the alternate allele.

True genotype data

The pre-pooling data set, or true data set, consists of n genotypes at each genetic position. Each data point is a genotype x which is fully known, that is to say $[p_{0i}, p_{1i}, p_{2i}]^\top$ is one of the three simplex in

$$\mathcal{X} = \left\{ \begin{bmatrix}1\\0\\0\end{bmatrix} \begin{bmatrix}0\\1\\0\end{bmatrix} \begin{bmatrix}0\\0\\1\end{bmatrix} \right\},$$

At any position, the n individuals in the population are i.i.d. data points which sampled at frequencies $\theta = [\theta_0, \theta_1, \theta_2]^\top$. They form an empirical distribution π_n

$$\mathbf{x} \sim \pi_n(x) \tag{1}$$

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \qquad \forall i \in [1, n] \;\; x_i = \begin{bmatrix} p_{0i} \\ p_{1i} \\ p_{2i} \end{bmatrix} \tag{2}$$

Under the assumption the HWE holds at each marker, the population-wide frequencies of the three genotypes at any marker are directly related to the alternate allele frequency (AAF) that we denote f. Therefore, the model of equation 1 can be reduced to a distribution which only depends on the variable f

$$\mathbf{x} \sim \pi_n(x; f) \tag{3}$$

We note $\hat{\theta}(f) = \mathbb{E}_n[\mathbf{x}; f]$ the empirical risk minimizer ERM estimating the mean of $\pi_n(x; f)$. Assuming HWE let the ERM be expressed as a single-variable parameter, that is

$$\theta(f) = \begin{bmatrix} (1-f)^2 \\ 2f(1-f) \\ f^2 \end{bmatrix} \tag{4}$$

As f is a continuous quantity, we discretize it for convenience in the simulation as the delimiting values of 21 equally-sized bins in the range $[0.0, 1.0]$. For each value of f, we simulate $n = 160$ genotypes (10 pooling blocks of 16 samples) for 10 genetic positions, that is a number $m = 200$ simulated markers.

The true genotypes $\mathbf{x}$ are assigned to $B = 100$ independent pooling blocks of $n_B = 16$ samples

$$\mathbf{x}_B = (x_1, x_2, \ldots, x_{16}), \tag{5}$$

and these blocks are used for simulating NORB pooling and decoding as the examples shown on Figure 1.

Pooled decoded genotype data

Let us describe pooling as a transformation t that maps the complete data $\mathbf{x}$ to the incomplete data $\mathbf{z}$ as follows

$$t \colon \mathcal{X} \longrightarrow \mathcal{Z} \tag{6}$$

$$\mathbf{x} \longmapsto \mathbf{z} \tag{7}$$

The vector $\mathbf{z}$ consists of n genotypes resulting from simulating pooling and decoding on the true data $\mathbf{x}$

$$\mathbf{z} = (z_1, z_2, \ldots, z_n) \qquad \forall i \in [1, n] \;\; z_i = \begin{bmatrix} \tilde{p}_{0i} \\ \tilde{p}_{1i} \\ \tilde{p}_{2i} \end{bmatrix} \tag{8}$$

and, correspondingly to equation 5, the pooled data within a block b are denoted

$$\mathbf{z}_B = (z_1, z_2, \ldots, z_{16}), \tag{9}$$

Depending on the pooling block configuration, the decoding is successful (or unambiguous) if the genotype z_i is a simplex as the ones in $\mathcal{X}$ (white items on Figure 1). In this case, z_i is said to be determined. If the decoding is ambiguous, the genotype is said to be indeterminate and it is considered as missing (orange items on Figure 1).

We introduce V the vector of indices in $\mathbf{z}$ for which the data is fully observed, and correspondingly $\mathbf{y} = \{y_k\}$, $k \in V$ the vector of observed genotypes i.e, determined after decoding. Conversely we use $\overline{V}$ to denote the vector of indices in $\mathbf{z}$ for which the data is unobserved, and $\mathbf{u} = \{u_k\}$, $k \in \overline{V}$ the vector of indeterminately decoded genotypes.

We are interested in studying the mappings t for any value of f. However, the pooling decoding process generating $\mathbf{z}$ cannot be formulated in a closed-form expression. Therefore, we model $\mathbf{z}$ as a sample from an unknown distribution

$$\mathbf{z} \sim \tilde{\pi}_n(z; f) \tag{10}$$

We consider that the distribution $\tilde{\pi}_n(f)$ has an empirical mean $\phi_n(f)$.

Characteristics of the missing data for the undecoded items in pooling blocks

The missing data $\mathbf{u}$ can be categorized as Missing Not At Random (MNAR) data [3], since it inherently depends on the other genotypes observed in each pooling block, as well as on the unobserved AAF at the given genetic position. Because of the NORB setting used, the missingness patterns in the pooled decoded data are by design nonmonotone.

Piece-wise estimates of the genotype probabilities in MNAR data with nonmonotone missing patterns

In another study (unpublished research), we propose a method for computing the most likely probability of each of the unobserved items $\mathbf{u}$ by inverse transform sampling.

The finite set of possible nomonotone missingness patterns can be categorized into subsets of block patterns ψ. All patterns with the same block pattern are just permutations of that pattern as illustrated on Figure 2.

The proposed method exhaustively enumerates all block patterns. For each pooling block having the pattern ψ, the probability of any genotype in $\mathbf{z}_B$ is conditioned on ψ. The variable f is marginalized and depending on the algorithm version implemented, any missing item in $\mathbf{u}$ is substituted with a fixed prior probability that can be initialized to any simplex. The missing data estimation problem over all patterns is solved piece-wise as a series of either Maximum Marginal Likelihood Estimation (MMLE) or EM [4].

Our method produces self-consistent estimates of the most likely genotype probabilities for any item in $\mathbf{u}$.

Using the computed estimates in place of any missing genotype in $\mathbf{u}$, we reconstruct a fully observable vector $\underline{\mathbf{z}}$ as if the pooled genotypes would be sampled from a distribution

$$\underline{\mathbf{z}} \sim \hat{\pi}_n(z) \tag{11}$$

The different versions we have implemented and tested correspond to variations of an EM inference method:

0. The reconstructed distribution $\underline{\mathbf{z}}$ corresponds to a a naive uninformed completion of the data, where any item in $\mathbf{u}$ is set to $(1/3, 1/3, 1/3)$. That is, all genotypes are equally likely, as they would be in the case of a "neutral" HWE and $f = 0.5$ at any marker.

1. The GP are sampled based on the expected allele frequency f in the entire block, that is from a binomial distribution with parameters f and $32 = 16 \times 2$ as each genotype is a pair of alleles. The expected allele frequency is initialized to $f = 0.5$ and then deduced at each iteration from the priors for the genotypes e.g. $f = 0.5\, Pr(G = 1) + Pr(G = 2)$. The posterior estimates for the GP are calculated with an iterative adjustment of the fixed priors. At each iteration, the posterior GP are divided by the prior and normalized in order to ensure the self-consistency of the algorithm. Moreover, since the heterozygotes estimates are degenerated, the posterior GP are rescaled by reweighing each genotype probability in the simplex and normalized in order to compensate for the degeneracy.

2. Similar to 1., but each of the 33 possible allele count outcomes in the block has an individual iteratively fitted probability. The lowest count of alleles is the case of a pooling block where all items have a genotype $G = 0$. Conversely, if all items have a genotype $G = 2$, the allele count sums up to 32. Therefore, there are in total $33 = 32 + 1$ possible allele count outcomes in a pooling block. At every iteration, the alleles for every individual in the block are sampled from the allelic binomial distribution (reference or alternate allele), and the GP posterior is deduced from the allelic frequencies before rescaling it with the GP prior.

3. Similar to above, but the allelic proportions are used as such and not as binomial parameters.

4. Similar to 2., but the alleles are sampled geometrically. The posterior genotype frequencies are directly used as priors at next iteration, without rescaling them with the former prior. On the whole, this process is very close to an EM algorithm.

We approximate the mean of the reconstructed distribution distribution with the empirical risk minimizer $\hat{\phi}_n = \mathbb{E}_n[\underline{\mathbf{z}}]$.

In this study, we evaluate the quality of the reconstructed empirical distributions $\hat{\pi}_n$ from the various approaches presented above, with respect to the simulated

empirical distribution π_n from which they were generated. We conduct a preliminary study of the quality of the reconstructed data based on two consistency criteria.

Clustered bootstrap sampling for pooled genotype data with a NORB design

Statistics for studying the consistency of empirical distributions

We use the following statistics to do a preliminary study of the consistency of the reconstructed empirical distribution $\hat{\pi}$:

- The L2 norm $\hat{\delta}_n = ||\hat{\pi}_n(z) - \pi_n(x)||_2$ which has been suggested for testing goodness-of-fit for densities in e.g. [5].
- The Kullback-Leibler divergence $\hat{\nu}_n = D_{KL}(\pi_n, \hat{\pi}_n)$ as suggested in e.g. [6], defined for the genotype data at one marker as

$$D_{KL} = \frac{1}{n} \sum_{i=1}^{n} \sum_{g=0}^{2} -p_{g,i} \ log \left(\frac{p_{g,i}}{\hat{p}_{g,i}} \right) \tag{12}$$

If the pooled reconstructed data $\underline{\mathbf{z}}$ are consistent with $\mathbf{x}$, we expect $\hat{\delta}_n \approx 0$ and $\hat{\nu}_n \approx 0$.

Statistics computed on every marker having frequency f in each f-bin.

Pooled genotype data reconstruction in the case of infinite sample size

The simulated genotype data and the reconstructed data from point-wise estimates have finite sample size n. We assume the distribution $\hat{\pi}_n$ is consistent with the distribution π_n. In the case of infinite sample size when $n \to \infty$, we expect the behavior

$$\hat{\pi}_n(z) \longrightarrow \pi^*(z) \tag{13}$$

For addressing the variability issue for the estimated statistics with a finite sample size n, we use a bootstrap resampling method to compute confidence intervals (CI) for both statistics $\hat{\nu}_n$ and $\hat{\delta}_n$.

Motivations for using clustered bootstrap sampling

Because of the NORB design chosen, the dependencies between the samples in the pooled genotype data vectors $\mathbf{z}$ and $\underline{\mathbf{z}}$ are particular. Every block is independent from the $B-1$ other ones but within a pooling block, the samples are no longer i.i.d.

$$\forall k \in [1, B] \ \ \forall j \in [1, n_B] \ \ z_j^k \not\perp \{z_{-j}^k\} \tag{14}$$

where z_{-j}^k is any sample but the j-th one in the k-th block.

Construction of the clustered bootstrap samples

Assimilating a pooling block to a cluster of data, we implement a specific bootstrap method for clustered data, largely based on the two-stage bootstrap described in [7]. However, if our block data are exchangeable (the order of the blocks does not matter) as in the two-stage bootstrap, the data within a block are not.

Let us form K bootstrap samples from the data $\underline{\mathbf{z}}$ by randomly choosing with replacement C clusters in the B blocks

$$\forall k \in [1, K] \ \ Z_k^* = \{Z_{k,1}^*, Z_{k,2}^*, \ldots, Z_{k,C}^*\} \tag{15}$$

In each bootstrap sample, we randomly sample a single data point per block such that the equation (15) becomes

$$\forall k \in [1, K] \ \ Z_k^* = \{\underline{z}_{k,1}^*, \underline{z}_{k,2}^*, \ldots, \underline{z}_{k,C}^*\} \tag{16}$$

For each bootstrap sample from $\underline{\mathbf{z}}$, we pick the same block and sample indices in the pre-pooling data $\mathbf{x}$

$$\forall k \in [1, K] \ \ X_k^* = \{x_{k,1}^*, x_{k,2}^*, \ldots, x_{k,C}^*\} \tag{17}$$

We note the mean of the k-th bootstrap sample as

$$\bar{Z}_{k\cdot}^* = C^{-1} \sum_{c=1}^{C} \underline{z}_{k,c}^*, \tag{18}$$

similarly for $\bar{X}_{k\cdot}^*$, such that $K \times C = N$. The bootstrap estimators of the Kullback-Leibler divergence and the L2-norm are formed as

$$\forall k \in [1, K] \ \ \hat{\nu}_{N,k} = D(\bar{X}_{k\cdot}^*, \bar{Z}_{k\cdot}^*) \tag{19}$$

$$\hat{\delta}_{N,k} = ||\bar{X}_{k\cdot}^* - \bar{Z}_{k\cdot}^*||_2 \tag{20}$$

$\hat{\nu}_N$ has a bootstrap estimated variance of

$$\hat{\mathbb{V}}_N = K^{-1} \sum_{k=1}^{K} \left(\hat{\nu}_{N,k} - K^{-1} \sum_{k=1}^{K} \hat{\nu}_{N,k} \right)^2 \tag{21}$$

In pratice, we choose $K = \lfloor 0.8B \rfloor$. The $1 - \alpha$ CI for the bootstrap statistics is hence defined as

$$\bar{T}_\alpha^N = \left\{ \nu : |\nu - \hat{\nu}_N| \leq \sqrt{\hat{\mathbb{V}}_N \, q_\alpha} \right\}, \tag{22}$$

similarly for $\hat{\delta}_N$.

Results

We do not claim to do any hypothesis test about the consistency, we are rather interested in preliminary results that let us visualize the dissimilarity between the simulated and reconstructed empirical distributions. The results presented are to be considered in the perspective of genotype imputation. Most methods achieving genotype imputation essentially consist in a HMM-based inference of the missing genotype data in a population and for a given set of markers. They generally assume that the genotypes at a marker in the population are at HWE. Genotype imputation produces posterior genotype probability estimates for each individual at each marker. The imputation algorithms internally double the prior genotype probability for the heterozygotes, so we need to rescale the *simpool* estimates by doubling the heterozygotes and normalizing every probability simplex in order to render how the reconstructed distribution would be used in the imputation method. Therefore, the results presented compare statistics calculated from the rescaled reconstructed distribution.

We use a ternary plot for representing a a genotype probability simplex. Ternary plots, synonymously de Finetti diagrams, are a standard representation of 3-dimensional data [8, 9, 10]. This representation provides a first intuitive visualization of the distributions as well as the distances between the different genotype estimates. The ternary plots presented are produced with a specific Python package [11].

Figure 3 shows an example of annotated ternary plot for the estimates computed in a pooling block of pattern $((2,2,0),(2,2,0))$. Table 1 gives the coordinates of every data point projected on the ternary axes on Figure 3 in order to facilitate the interpretation of the ternary plot.

Figure 4 shows the empirical means of the rescaled data in each AAF-bin on a ternary plot. The 'true' line represents the distribution from which the data $\mathbf{x}$ is sampled. The heterozygotes are under-represented in all reconstructed distributions, which indicates that the *simpool* algorithm tends to favor the inference of homozygous genotypes. For example, in place of two missing items, two opposite homozygotes are more likely than two heterozygotes. The closest reconstruction of the pooled distribution is achieved with the version 4 of *simpool*.

The L2 distance measures shown on Figure 5 present the same characteristics as the D_{KL} measures. Since the reference and alternate alleles for biallelic markers have symmetrical properties, the allelic frequency is commonly presented as Minor Allele Frequency (MAF) rater than AAF. The minor allele is either the alternate or the reference one depending on its frequency. The L2 distance is a commonly used metrics that reveals how far the data points forming the empirical 'true' and the empirical reconstructed distribution are. If the metrics is equal to 0, the data points have identical coordinates. This is for example the case if $MAF = 0$ on Figure 5, that is to say the population studied is purely homozygous for the major allele at the marker considered. Given the NORB pooling design used in this experiment, all genotypes are decoded as homozygous for a pure homozygous population. Therefore the decoded data are identical to the true one and the L2 norm is null. However, we are more interested in studying the distributional consistency between the 'true' and the reconstructed distributions than the distance between single points. Indeed, since the GP estimates are to be used as prior probabilities for genotype imputation, we need to consider the dissimilarity between the distribution from this perspective. In genotype imputation in a population and especially at the phasing step, the posterior genotype probabilities computed for the indeterminate markers depend on the Linkage Disequilibrium (LD) between the markers. The LD renders the probability that the genotypes of a sequence of markers in parent individuals are inherited together by the offspring. This metrics is correlated to the physical distance between the markers in the DNA but the relationship is not linear, such that the L2 distance is not the most well-suited metrics for apprehending how the prior genotype probabilities might impact the imputation. The concept of D_{KL} is more relevant for studying the distributional consistency and quantifying the information loss between probabilities, therefore we prefer to focus on describing the divergence results of the bootstrap resampling.

The divergence D_{KL} between π_n and $\hat{\pi}_n$ across range of the MAF values is shown on Figure 6. It presents the same characteristics as the confidence intervals for the L2 distance. D_{KL} quantifies as a single measure the dissimilarity between the reconstructed distribution and the 'true' distribution it was pooled from. Overall, all CI-envelopes reveal a correlation between D_{KL} and the MAF. As for the L2 distance, the minimum is observed if the data is purely homozygous ($MAF = 0$) since all items are fully decoded to homozygotes. The least divergence is also achieved if both alleles are in equal proportions ($MAF = 0.5$). Around $MAF = 0.5$, all pooled genotypes are very likely to be missing and this results in nearly uniform estimates ($\tilde{\pi} \sim (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$) regardless of the version of *simpool* that is used. After rescaling, the

estimates are almost equal to $(0.25, 0.5, 0.25)$ which are the HWE proportions used for generating the 'true' data set when $MAF = 0.5$. The situation is very similar to the case of fully missing data with equally likely genotypes for each of the unassayed markers, which is on the whole the assumption made by most imputation methods. The divergence of the reconstructed distributions reaches a maximum around 0.05 for $MAF = 0.2$, except from the reconstruction with the version 4 of *simpool* for which the maximum is shifted to $MAF = 0.3$. When $MAF = 0.2$, the homozygotes for the major allele are dominating in the true data, whereas $MAF = 0.3$ coincides with the frequency at which the heterozygotes are the most frequent in a population at HWE.

When designing *simpool*, we expected our estimates to be closer to the true distribution than the default case of uniform data completion (version 0). Figure 6 however shows that the estimates from the versions $1, 2, 3$ are almost identical to the naive version 0. In other words, the computed estimates do not add much information about the most likely genotype. As it is already suggested by Figure 4, the reconstructed distribution is the most consistent with the 'true' data when using estimates computed with the version 4 of *simpool*. This reconstruction is also the most accurate, as the narrow curve envelope indicates. The version 4 of *simpool* was implemented while conducting this study as we noticed that the earlier versions $1, 2, 3$ were not satisfying. The version 4 intends to improve the consistency of the reconstructed pooled distribution. While D_{KL} still correlates to the MAF, it is significantly lower (at most 0.012) than with the previous versions (up to 0.055) and is almost null for $MAF = 0.5$. The divergence measures reveals that we have succeeded in capturing better the the 'true' distribution when reconstructing the pooled data.

Conclusions

Many studies have proposed powerful algorithms for decoding binary outcomes from pooled data and NORB is one pooling design example that has been investigated. When the test outcomes are ternary $(G = 0, 1, 2)$ as for genotyping biallelic markers, the DNA Sudoku method described in [2] is not robust enough for decoding the pooled genotypes. We implement in [1] various EM-based estimation methods specifically tailored for reconstructing the incomplete genotype data from a NORB pooling design.

The findings of the present study should be put in the context of the genotype imputation that we are interested in with our research [1]. Indeed, it is essential that the GP estimates forming the reconstructed distribution favor the downstream imputation of the correct genotype. In this perspective, the consistency between the reconstructed distribution and the 'true' one is more relevant than the physical closeness of the point-wise GP estimates. Therefore, the quality metrics should not only focus on the divergence from the true data but also reward the information gain they bring to the pooled data.

In this paper, we made a preliminary analysis of the consistency of these various GP reconstruction methods in with a divergence and a distance measure. In order to account for the limitations of the numerical representation of the genotypes in *simpool*, we have introduced the concept of heterozygotes degeneracy. However, the first versions of *simpool* did not appear to be satisfying and we therefore explored variations with the explicit intent to minimize the values of the KL divergence. The later versions of *simpool* were implemented as we started investigating the consistency of the reconstructed distribution. Thanks to a geometrical sampling of the alleles at each iteration, the improved *simpool* algorithm manages to capture better the allelic distribution at the level of the pooling block, as well as the derived genotype frequencies. The version 4 of *simpool* uses the same initial prior probabilities for each missing item regardless of the pooling pattern in a block. Another strategy, possibly improving the consistency of the reconstructed pooled distribution, could choose the initial allelic priors depending on the pooling pattern observed.

It is difficult to assess from D_{KL} only which of the heterozygous or homozygous genotypes contribute the most to the divergence. A further analysis of the divergence in relation to the heterozygosity rate might enlight new improvements that could be made in the GP estimation method. Moreover, a broader investigation of the information gain brought by our adaptive GP estimates to imputation would be suitable, especially compared to a naive uninformed completion. We suggest for this purpose to study the imputation results we obtained in an earlier paper [1] with the results we would obtain for the same pipeline but replacing the reconstructed estimates with values of later *simpool* versions e.g. version 4 as they have the highest consistency.

References

[1] C. Clouard, K. Ausmees, and C. Nettelblad. *A Joint Use of Pooling And Imputation For Genotyping SNPs*. 2021. DOI: 10.21203/rs.3.rs-1131930/v1. URL: https://doi.org/10.21203/rs.3.rs-1131930/v1.

[2] A. Gordon et al. Y. Erlich K. Chang. “DNA Sudoku—harnessing high-throughput sequencing for multiplexed specimen analysis”. In: *Genome Research* 19 (2009), pp. 1243–1253.

[3] D. B. Rubin. “Inference and missing data”. In: *Biometrika* 63 (1976), 581–592.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum Likelihood from Incomplete Data Via the EM Algorithm”. In: *Journal of the Royal Statistical Society* 39.1 (1977), pp. 1–22.

[5] M. H. Neumann and E. Paparoditis. “On bootstrapping L2-type statistics in density testing”. In: *Statistics & Probability Letters* 50.2 (2000), pp. 137–147. ISSN: 0167-7152. DOI: https://doi.org/10.1016/S0167-7152(00)00091-2. URL: https://www.sciencedirect.com/science/article/pii/S0167715200000912.

[6] A. Lindholm et al. “Data Consistency Approach to Model Validation”. In: *IEEE Access* 7 (2019), 59788–59796. DOI: 10.1109/ACCESS.2019.2915109.

[7] C. A. Field and A. H. Welsh. “Maximum Likelihood from Incomplete Data Via the EM Algorithm”. In: *Journal of the Royal Statistical Society* 69.Part 3 (2007), pp. 369–390.

[8] C. Cannings and A. W. Edwards. “Natural selection and the de Finetti diagram”. In: *Ann Hum Gen* 31 (1968), 421–428. DOI: https://doi.org/10.1111/j.1469-1809.1968.tb00575.x.

[9] Richard J. Howarth. “Sources for a history of the ternary diagram”. In: *The British Journal for the History of Science* 29.3 (1996), 337–356. DOI: 10.1017/S0007087400003449X.

[10] A. W. Edwards. *Foundations of Mathematical Genetics 2nd Edition*. Cambridge University Press, 2000. ISBN: 978-0-521-77544-1.

[11] Marc Harper et al. “python-ternary: Ternary Plots in Python”. In: *Zenodo 10.5281/zenodo.594435* (2015). DOI: 10.5281/zenodo.594435. URL: https://github.com/marcharper/python-ternary.

Artwork

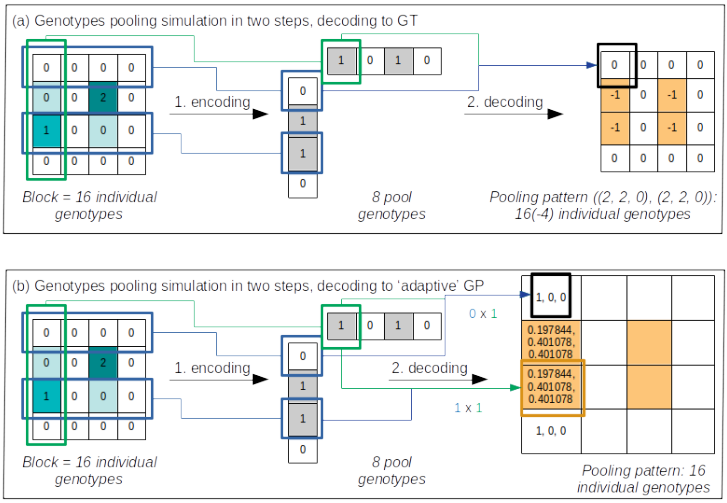
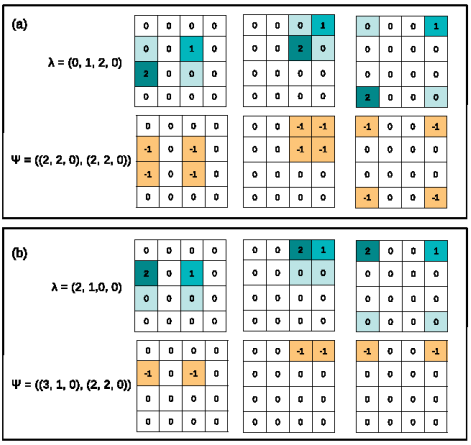


Figure 1:

Example of pooling simulation with a NORB algorithm.
A pooling block of $n_B = 16$ samples is modelled as a square matrix, the rows and the columns form 8 intersecting pools of 4 samples each. The encoding step assigns the 4 individual genotypes to a pool and pooling is done as follows: the genotype of a pool is 0 (resp. 2) iff all samples are 0 (resp. 2), as for example the top row-pool. In all others cases (allelic-heterogeneous pools), the genotype of the pool is 1, as for example the leftmost column-pool (green frame). The decoding step reconstructs the genotype of every sample based on the intersecting pools. Decoding is successful if at least 1 homogeneous pool (genotype 0 or 2) is involved. Otherwise, the genotype of the sample is indeterminate and considered as missing.
A block is described by its pooling pattern $\psi = (n_{G_{rows}}, n_{G_{columns}})$ where $n_{G_{rows}}$ (respectively $n_{G_{columns}}$) gives the number of row-pools (resp. column-pools) in the block having the genotype 0, 1, and 2.
Subfigure (a): the pooled genotypes are decoded into integer genotypes (GT format) in $\{0, 1, 2, -1\}$ representing, respectively, a homozygote for the reference allele, a heterozygote, a homozygote for the alternate allele, or a missing item. In this example, there are 4 indeterminate samples. The pooling pattern ψ is $((2, 2, 0), (2, 2, 0)$, and the sample highlighted by a black square is intersected by pools having genotype 0 and 1.
Subfigure (b): the pooled genotypes are decoded to adaptive genotype probabilities (GP format) that are computed with a Maximum Marginal Likelihood estimation method. We qualify the genotype probabilities as 'adaptive', as we estimate them relatively to the pattern of the pooling block that the samples are part of. Four samples have an ambiguous genotype, for which none the genotype probabilities is 1. ψ is the same as on the subfigure (a).



Permutations of block patterns.
Examples of two pooling patterns obtained from two distinct permutations from the same set of genotypes. λ denotes the subvector of blue-colored genotypes that are possible completions of $\mathbf{z}$.

Figure 2:

Subfigure (a): The carriers of the alternate allele e.g. having the genotype 1 or 2 are located on different rows and different columns, such that they never show up in the same pool. In the three pooling blocks shown, the pooling pattern $\psi = ((2, 2, 0), (2, 2, 0))$ is the same while they result from different permutations of the completed data $\mathbf{z}$.

Subfigure (b): The carriers of the alternate allele are located on different columns but the same rows, such they are genotyped together in the row pool. The three pooling blocks shown have the same pooling pattern $\psi = ((3, 1, 0), (2, 2, 0))$.

version	Pr(G=0)	Pr(G=1)	Pr(G=2)
0	0.250000	0.500000	0.250000
1	0.141208	0.572528	0.286264
2	0.143231	0.606085	0.250684
3	0.134242	0.627877	0.237881
4	0.313383	0.654295	0.032322

Table 1: Rescaled most likely genotype probabilities computed by different versions of the *simpool* algorithm for undecoded items in a pooling with pattern $\psi = ((2, 2, 0), (2, 2, 0))$

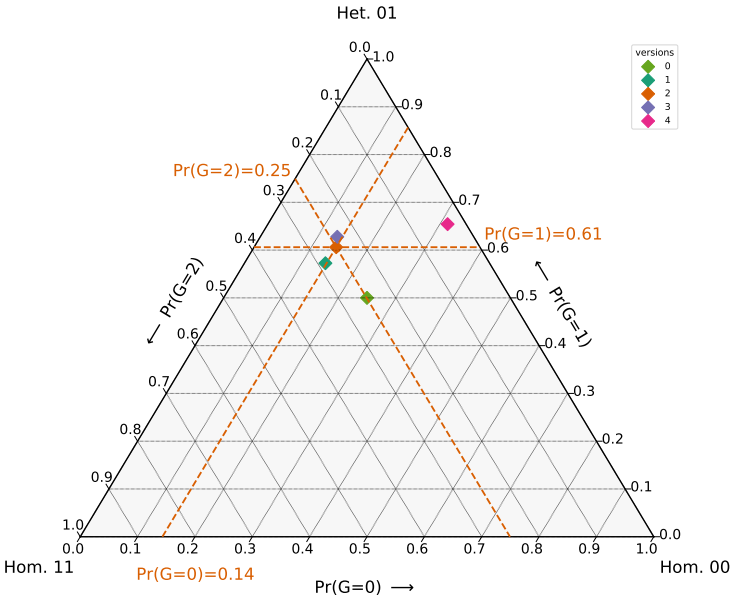


Figure 3: **Example of de Finetti diagram: Genotype probabilities estimates for the missing data in a pooling block with pattern $\psi = ((2, 2, 0), (2, 2, 0))$.** The annotations on the three axes indicate the coordinates of the orange point that is the GP estimate computed with the version 2 of *simpool*. The orange lines represent the projection of the data point on the axes. The values for all the data points displayed are given in Table 1. Each of the tops of the triangle is the position for a fully known genotype, either homozygous or heterozygous.

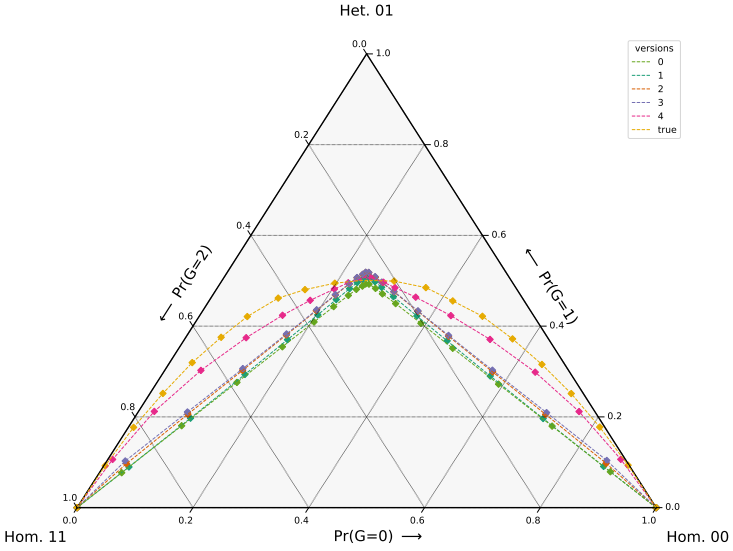


Figure 4: **De Finetti diagram of the averaged genotype probabilities in true and reconstructed pooled data for each allele frequency bin.**
The series of points represent the mean genotype probabilities computed from genetic markers with increasing allele frequency f: the smallest frequencies (from $f = 0.05$) are located at the bottom right corner Hom. 00 ($Pr(G = 0)$ is almost 1) and the largest frequencies (up to $f = 0.95$) are located at the bottom left corner Hom. 11 ($Pr(G = 2)$ is almost 1). The 'true' points and line in yellow show the bin-averaged genotype probabilities from the true data set used to simulate pooling. The other points show the bin-averaged genotype probabilities from rescaled pooled data that was completed with different versions of *simpool*.

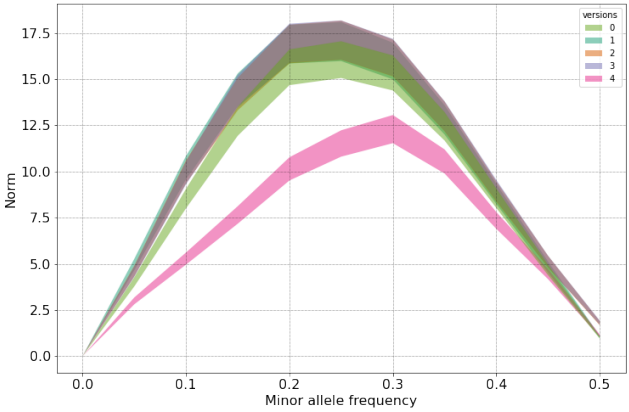


Figure 5:

95% bootstrap confidence intervals for the L2 distance.
The distributional L2 distance is computed between a 'true' empirical distribution and reconstructed empirical distributions. The true data consists of genotypes sampled under the HWE assumption, and used for simulating genotype pooling experiments. The reconstructed distributions consist of decoded pooled data and different estimates of the missing data that are computed with various versions of the *simpool* algorithm. Each data point in the reconstructed distribution is rescaled before averaging the genotype probabilities in each MAF-bin. The rescaling takes into account the heterozygotes degeneracy. The allele frequency is presented as MAF since the reference and the alternate alleles have symmetrical properties when the genotype data are pooled. A null value for the L2 distance indicates that the reconstructed distribution is perfectly consistent with the true one. The L2 distance computed from a reconstructed distribution based on the *simpool* version 4 has a different shape from all other versions and is the most consistent one. This is the only version of *simpool* that uses a geometrical resampling of the genotypes at each iteration of the algorithm.

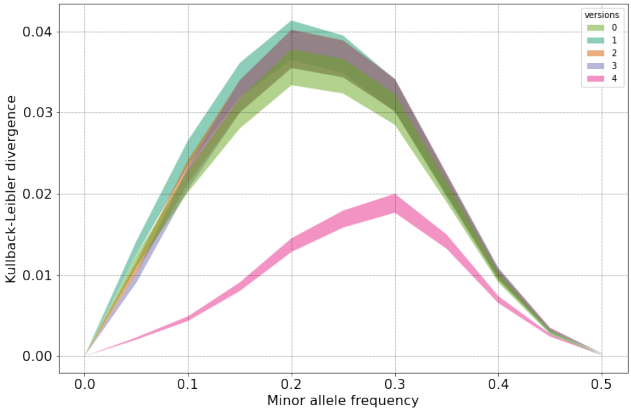


Figure 6:

95% bootstrap confidence intervals for the Kullback-Leibler divergence. The distributional divergence is computed between a 'true' empirical distribution and reconstructed empirical distributions. The true data consists of genotypes sampled under the HWE assumption, and used for simulating genotype pooling experiments. The reconstructed distributions consist of decoded pooled data and different estimates of the missing data that are computed with various versions of the *simpool* algorithm. Each data point in the reconstructed distribution is rescaled before averaging the genotype probabilities in each MAF-bin. The rescaling takes into account the heterozygotes degeneracy. The allele frequency is presented as MAF since the reference and the alternate alleles have symmetrical properties when the genotype data are pooled. A null value for the divergence indicates that the reconstructed distribution is perfectly consistent with the true one. Notably, the D_{KL} computed from a reconstructed distribution based on the *simpool* version 4 has a different shape from all other versions and is the most consistent one. This is the only version of *simpool* that uses a geometrical resampling of the genotypes at each iteration of the algorithm.

Recent licentiate theses from the Information Technology

2022-002 Gustaf Borgström: *Making Sampled Simulations Faster by Minimizing Warming Time*

2022-001 Sam Hylamia: *Secure In-body Communication and Sensing*

2021-002 Karl Bengtsson Bernander: *Improving Training of Deep Learning for Biomedical Image Analysis and Computational Physics*

2021-001 Niklas Gunnarsson: *On the Registration and Modeling of Sequential Medical Images*

2020-006 David Widmann: *Calibration of Probabilistic Predictive Models*

2020-005 Anna Wigren: *Exploiting Conjugacy in State-Space Models with Sequential Monte Carlo*

2020-004 Muhammad Osama: *Machine Learning for Spatially Varying Data*

2020-003 Christos Sakalis: *Securing the Memory Hierarchy from Speculative Side-Channel Attacks*

2020-002 Ulrika Sundin: *Global Radial Basis Function Collocation Methods for PDEs*

2019-007 Carl Andersson: *Deep Learning Applied to System Identification: A Probabilistic Approach*

2019-006 Kristiina Ausmees: *Efficient Computational Methods for Applications in Genomics*

2019-005 Carl Jidling: *Tailoring Gaussian Processes for Tomographic Reconstruction*

UPPSALA
UNIVERSITET

Department of Information Technology, Uppsala University, Sweden