GBScleanR: Robust genotyping error correction using hidden Markov model with error pattern recognition.

Tomoyuki Furuta[1,✉], **Toshio Yamamoto**[1], **and Motoyuki Ashikari**[2]

[1]Institute of Plant Science and Resources, Okayama University, Chu-oh 2-20-1, Kurashiki, Okayama, 710-0046, Japan
[2]Bioscience and biotechnology center, Nagoya University, Furo-cho, Chikusa, Nagoya, Aichi, 464-8601, Japan

The developments in sequencing technology have enabled researchers to acquire genotype data from large populations with dense markers. Recently developed methods that are based on reduced-representation sequencing (RRS), such as Genotyping By Sequencing (GBS), provide cost-effective and time-saving genotyping platforms; however, many drawbacks that are associated with these technologies, such as missing and false homozygous calls at heterozygous sites, significantly affect the accuracy. Several error correction methods that incorporate allele read counts in a hidden Markov model (HMM) have been developed to overcome these issues. Those methods assume that markers have a uniform error rate with no bias in the allele read ratio and infer a 50% chance of obtaining a read for either allele at a heterozygous site. However, bias does occur because of uneven amplification of genomic fragments and read mismapping. In this paper we introduce a novel error correction tool, GBScleanR, which enables robust and precise error correction for noisy RRS-based genotype data by incorporating marker-specific error rates into the HMM. The results indicate that GBScleanR improves the accuracy by 10-40 percentage points as compared to the existing tools in simulation datasets and achieves the most reliable genotype estimation in real data even with error prone markers.

Correspondence: ***f.tomoyuki@okayama-u.ac.jp***

DRAFT

Introduction

The rise of next generation sequencing (NGS) has opened the door to a new generation of genetic and genomic research (1). NGS provides genotyping platforms with dramatically improved flexibility and throughput that are now routinely applied to diverse fields of biology in which information about the nucleotide sequences of organisms is paramount (2–4). The cost of NGS has decreased significantly over the past decade. Nevertheless, obtaining a genotype via whole genome resequencing is still an expensive option for researchers, particularly those working on breeding projects that involve the genomes of thousands of individuals (5). Many different RRS-based genotyping methods have been introduced to meet the demands for cost-effective genotyping systems with dense markers (6). Since restriction-site associated DNA sequencing (RAD-seq) was first published in 2007, several derivative methods have been introduced, including methods such as ddRADseq, nextRAD, ezRAD, 2b-RAD, and RAD Capture (7–13). GBS and its derivative, two-enzyme GBS, are also outstanding RRS methods (14, 15). GBS has already been widely applied to many crops and animals, such as rice, sorghum, chickpea, cattle, and mice (16–20). GBS, RAD-seq and its derivatives basically obtain sequences from limited portions of genomes by fragmentating genomic DNA with restriction enzymes before sequencing from adapters that are ligated at the cut site.

This reduction in the number of sequence targets allows for multiplexed sequencing, freeing researchers from the laborious pre-surveys that are usually required to find useful genotyping markers for PCR-based and hybridization-based genotyping systems such as simple sequence repeat (SSR) genotyping and SNP genotyping array (21, 22). However, RRS has well-known drawbacks that are derived from its technological nature and include mismapped reads and the undercalling of heterozygous alleles. Although the most significant advantage of RRS lies in its multiplexibility, this characteristic reduces the number of reads that can be obtained from each sample and the sequences that can be derived are highly stochastic. These issues can lead to uneven read acquisition at some markers. The limited number of reads, together with uneven coverage, can result in collecting information about only one allele in a heterozygote, leading to its incorrect recognition as a homozygote. This typical error form is known as the undercalling of heterozygotes. The lack of reads that is associated with uneven read acquisition may also generate a substantial number of missing genotype calls. Several error correction tools have been developed to overcome these disadvantages, with two categories available, one of which imputes the missing genotype calls and the other both imputes the missing genotype calls and corrects the undercalling in heterozygotes. Beagle is the most popular example of the former, and is a tool for the haplotype-phasing that is frequently used in GWAS that can infer missing genotypes based on the phased haplotype information (23). The method that is used to correct the undercalling of heterozygotes was first published in 2009, and determines a genotype from the allele read ratio within a sliding window (24). Recent publications have introduced more statistically sophisticated methods that use HMMs, such as FSFHap, TIGER, LB-Impute, and magicImpute (25–28). An HMM is a statistical model in which it is assumed that a series of observed data is the output of a sequence of hidden states that follow a Markov process (29). In the case of genotyping data, the series of ob-

served data and the sequence of hidden states correspond to the observed genotype calls (or allele read counts) and the true genotype along a chromosome, respectively. The existing error correction methods provide a robust estimation of the true genotype against a background that includes the undercalling of heterozygotes and missed calls that result from stochastic error.

The existing error correction methods assume constant error rates for all markers. For example, a 50:50 probability that a read will be obtained for one of two possible alleles at a heterozygous site is assumed. Although this assumption is true in an ideal situation, in practice, genotypic data contains a significant number of error prone markers that show skewed probabilities in the allele read acquisition as a result of actual biological unevenness (30–32). Polymorphisms that exist between the genomes of samples change the fragmentation patterns of the different genomes. Even if two independent reads are mapped at the same locus, the sequences of the genomic fragments from which each of the two reads originated can vary in terms of GC content and length, and sometimes include large insertions or deletions in the unsequenced region of the fragments. Since the GC contents and the lengths of the restricted fragments are known to affect the amplification efficiency, the probability of observing a read for either allele may differ. These biases are likely to be more prominent in a polymorphism-rich population, for example, that derived from a cross between distant relatives. The mismatches that occur because the uniform error rate that is assumed in the models do not match those in the real data, which are not uniform, result in poor error correction accuracy and a misleading and biased estimation of the genotype.

Here we introduce a new R package "GBScleanR" that implements a novel HMM-based error correction algorithm. Our algorithm estimates the allele read bias and mismap rate per marker and incorporates these into the HMM as parameters to capture the skewed probabilities in read acquisitions. This paper demonstrates a comparison of GBScleanR and two well established error correction tools; LB-Impute and magicImpute (25, 28). While LB-Impute accepts only biparental populations that are derived from inbred founders, magicImpute can work on bi- and multi- parental populations that are derived from both inbred and outbred founders. The magicImpute algorithm is also able to estimate founder genotypes simultaneously with the offspring genotypes, meaning that it does not require complete and high-quality information concerning founder genotype. Like magicImpute, GBScleanR also supports many scenarios in which experimental crosses are usually conducted and is designed to estimate founder and offspring genotypes simultaneously with higher robustness for error prone markers. We first show simulation studies to present the accuracy and robustness of GBScleanR, which includes the applications of the algorithms for datasets that have severe allele read biases at error prone markers. The simulation assumes three scenarios; a biparental F_2 population, an outbred F_2 population, and an 8-way recombinant inbred line (RIL). We then demonstrate the reliability and robustness of GBScleanR using the real data derived from a cross between distant relatives of rice, which potentially contains a number of error prone markers.

ALGORITHM AND METHODS

Overview of the GBScleanR package. GBScleanR is an R package and consists of a main tool that conducts error correction and utility functions that handle genotype data in the variant call format (VCF). Supplemental Figure S1 presents the data analysis work-flow and the related functions that are implemented in GBScleanR. The algorithm is designed to estimate the true genotype calls along a chromosome from the given allele read counts in the VCF file that is generated by SNP callers such as GATK and TASSEL-GBS (33, 34). The current implementation supports genotypic data for mapping populations that are derived from two or more diploid founders followed by selfings, sibling crosses, or random crosses, for example F_2 and 8-way RILs. Our method assumes that markers are biallelic and ordered along the chromosomes, and the reads are mapped onto a reference genome sequence. To access the large amounts of genotype data required, each input VCF file is first converted into a genomic data structure (GDS) file (35). The current implementation cannot correctly convert a VCF file containing non-biallelic markers to a GDS file. Thus, any input VCF file is subjected to filtering so that only biallelic SNPs are retained using, for example, bcftools (36).
GBScleanR provides functions for data visualization, filtering, and loading/writing a VCF file. Furthermore, the data structure of the GDS file created via this package is compatible with those used in the SNPRelate, GWASTools and GENESIS packages, which are designed to handle large amounts of variant data and conduct regression analysis (35, 37, 38). Therefore, GBScleanR can be built in a pipeline for an association study that allows us smooth access to the given large genotype data for visualization, filtering, error correction, and regression analysis without the requirement for extra data format conversion steps.

Modeling for error correction. The error correction algorithm of GBScleanR is based on the HMM and treats the observed allele read counts for each SNP marker along a chromosome as outputs from a sequence of latent true genotypes. Our model supposes that a population of $N^o \geq 1$ sampled offspring is originally derived from crosses between $N^f \geq 2$ founder individuals. The founders can be inbred lines with homozygotes at all markers or outbred lines in which markers show a heterozygous genotype. Only one chromosome is considered for modeling due to the independence of chromosomes. Let $\boldsymbol{Y}^o = \{y^o_{mi}\}_{m=1\ldots M,\ i=1\ldots N^o}$ denote the observed allele read counts at marker m in offspring i. The element y^o_{mi} consists of two values, y^{ref} and y^{alt}, which represent the reference read count and the alternative read count, respectively. Similarly, the observed allele read counts at marker m in founder j are represented by $\boldsymbol{Y}^f = \{y^f_{mj}\}_{m=1\ldots M,\ j=1\ldots N^f}$. On the other hand, the matrices for hidden offspring genotypes and hidden founder genotypes are represented by $\boldsymbol{X}^o = \{x^o_{mi}\}_{m=1\ldots M,\ i=1\ldots N^o}$

DRAFT

and $\boldsymbol{X}^f = \{x^f_{mj}\}_{m=1\ldots M,\ j=1\ldots N^f}$, respectively. The element x^o_{mi} takes a value of 0, 1, or 2 to indicate the reference homozygote, heterozygote, and alternative homozygote genotypes without phasing information. Unlike the offspring genotype matrix, the founder genotype x^f_{mj} stores the phased genotype as $x^f_{mj} = (x^1, x^2)$ where x^1 and x^2 indicate alleles at marker m on one of the diploid chromosomes and another in founder j, respectively. The reference allele is represented by 0, while 1 denotes the alternative.

Considering the linkage between the markers and the independence of the founder genotypes, we can assume that the sequence of the genotypes in the ith sample, which is denoted as $\boldsymbol{x}^o_i$, does not follow a Markov process, while the sequence of the descendent haplotypes $\boldsymbol{h}^o_i$ that is derived from N^f founders along a pair of chromosomes does. To enable our estimation problem to be solved in the HMM framework, $\boldsymbol{H}^o = \{h^o_{mi}\}_{m=1\ldots M,\ i=1\ldots N^o}$ is introduced to represent the matrix of the phased descendent haplotypes. The element h^o_{mi} denotes a pair of descendent haplotypes (h^1, h^2) at marker m in offspring i. Therefore, h^1 and h^2 each takes one of the natural numbers that are less than or equal to the number of founders N^f if all founders are inbred or double the number of founders $2N^f$ if outbred, to indicate the origins of the descendent haplotypes.

The algorithm estimates $\boldsymbol{H}^o$ and $\boldsymbol{X}^f$ based on $\boldsymbol{Y}^o$ and $\boldsymbol{Y}^f$ by maximizing the joint probability $P(\boldsymbol{H}^o, \boldsymbol{X}^f, \boldsymbol{Y}^o, \boldsymbol{Y}^f)$. The parameters used in the model are not shown here. The joint probability can be transformed as seen below.

$$P(\boldsymbol{H}^o, \boldsymbol{X}^f, \boldsymbol{Y}^o, \boldsymbol{Y}^f) = P(\boldsymbol{H}^o, \boldsymbol{Y}^o | \boldsymbol{X}^f, \boldsymbol{Y}^f) P(\boldsymbol{X}^f, \boldsymbol{Y}^f).$$

Each probability on the right hand of the equation can be farther transformed.

$$P(\boldsymbol{H}^o, \boldsymbol{Y}^o | \boldsymbol{X}^f, \boldsymbol{Y}^f) = P(\boldsymbol{Y}^o | \boldsymbol{H}^o, \boldsymbol{X}^f) P(\boldsymbol{H}^o | \boldsymbol{X}^f),$$
$$P(\boldsymbol{X}^f, \boldsymbol{Y}^f) = P(\boldsymbol{Y}^f | \boldsymbol{X}^f) P(\boldsymbol{X}^f).$$

We assume the independence of the offspring and then rewrite the first equation as

$$P(\boldsymbol{H}^o, \boldsymbol{Y}^o | \boldsymbol{X}^f, \boldsymbol{Y}^f) = \prod_{i=1}^{N^o} P(\boldsymbol{y}^o_i | \boldsymbol{h}^o_i, X^f) P(\boldsymbol{h}^o_i | \boldsymbol{X}^f).$$

We can now treat $P(\boldsymbol{y}^o_i | \boldsymbol{h}^o_i, \boldsymbol{X}^f) P(\boldsymbol{h}^o_i | \boldsymbol{X}^f)$ as the HMM for i^{th} offspring's haplotypes and the read counts, given the founder genotypes $\boldsymbol{X}^f$. On the other hand, $P(\boldsymbol{Y}^f | \boldsymbol{X}^f) P(\boldsymbol{X}^f)$ is the product of the probability of observing $\boldsymbol{Y}^f$ under the condition $\boldsymbol{X}^f$ and the probability of obtaining $\boldsymbol{X}^f$. The genotypes at each marker from each founder can be considered independent of each other and do not follow a Markov process. Therefore, our algorithm tries to maximize the following joint probability:

$$\begin{aligned} P(\boldsymbol{H}^o, \boldsymbol{X}^f, \boldsymbol{Y}^o, \boldsymbol{Y}^f) = \prod_{i=1}^{N^o} \{ & \prod_{m=1}^{M} P(y^o_{mi} | h^o_{mi}, \boldsymbol{x}^f_m) \\ & \times \prod_{m=2}^{M} P(h^o_{mi} | h^o_{m-1,i}, \boldsymbol{x}^f_{m-1}) P(h^o_{1i} | \boldsymbol{x}^f_1) \} \\ & \times \prod_{m=1}^{M} P(\boldsymbol{y}^f_m | \boldsymbol{x}^f_m) P(\boldsymbol{x}^f_m), \end{aligned}$$

where $P(y^o_{mi} | h^o_{mi}, \boldsymbol{x}^f_m)$, $P(h^o_{mi} | h^o_{m-1,i}, \boldsymbol{x}^f_{m-1})$, and $P(h^o_{1i} | \boldsymbol{x}^f_1)$ correspond to the emission probability, the transition probability, and the initial probability of the HMM, respectively, and $P(\boldsymbol{y}^f_m | \boldsymbol{x}^f_m)$ is the probability of observing read counts for the founders at marker m when the combination of the true genotypes of the founders is $\boldsymbol{x}^f_m$. We assume that all SNP markers are biallelic and that $P(\boldsymbol{x}^f_m)$ follows a discrete uniform probability for all possible combinations of the founder genotypes, while omitting cases in which all founders have homozygotes on the same allele, e.g., when all founders have reference homozygote genotypes.

Emission probability. The descendent haplotype h^o_{mi} itself does not provide any information about the alleles that are obtained by the descendants of the founders. We can only calculate the probabilities of obtaining y^o_{mi} from x^o_{mi}, but not from h^o_{mi} directly. Therefore, to obtain the emission probability $P(y^o_{mi} | h^o_{mi}, \boldsymbol{x}^f_m)$, three additional probabilities are introduced; $P(y^o_{mi} | x'^o_{mi}, e^{seq}, w_m)$, $P(x'^o_{mi} | x^o_{mi}, e^{map}_m)$, and $P(x^o_{mi} | h^o_{mi}, \boldsymbol{x}^f_m)$, where e^{seq} and e^{map}_m represent the sequencing error rate and the mapping error rate of the observed reads, respectively. The allele read bias w_m is the key parameter of the HMM in GBScleanR that accounts for systematic bias toward either of the alleles in the reads. We assume that each marker has its own mapping error rate and allele read bias, while the sequencing error rate is set to be uniform for all markers. x'^o_{mi} indicates the observable genotype that results from accounting for the mismapping of reads with a mapping error rate of e^{map}_m. Combining these parameters allows the emission probability to be replaced with $l^o_{mi} = P(y^o_{mi} | h^o_{mi}, \boldsymbol{x}^f_m, e^{map}_m, e^{seq}, w_m)$ and it holds that

$$l^o_{mi} = \sum_{x^o_{mi}} P(y^o_{mi} | x^o_{mi}, e^{map}_m, e^{seq}, w_m) P(x^o_{mi} | h^o_{mi}, \boldsymbol{x}^f_m),$$
$$\begin{aligned} P(y^o_{mi} | x^o_{mi}, e^{map}_m, e^{seq}, w_m) = & \sum_{x'^o_{mi}} P(y^o_{mi} | x'^o_{mi}, e^{seq}, w_m) \\ & \times P(x'^o_{mi} | x^o_{mi}, e^{map}_m), \end{aligned}$$

where $P(x^o_{mi} | h^o_{mi}, \boldsymbol{x}^f_m)$ is given a value of 1 for the possible genotype under the constraint of the descendent haplotypes and founder genotypes in sample i and 0 for the other genotypes. For example, if sample 1 has $h^o_{1,1} = (h^1 = 1, h^2 = 4)$ at marker 1, indicating that the haplotypes descended from

DRAFT

the first chromosome of founder 1 and the second chromosome of founder 2 and if the founder genotypes are $x^f_{1,1} = (0,0)$ and $x^f_{1,2} = (0,1)$, the genotype can only be heterozygotic $x^o_{1,1} = 1$.

We assume that the observed read counts $y^o_{mi} = \{y^{ref}, y^{alt}\}$ follow binomial distribution

$$P(y^o_{mi}|x'^o_{mi} = 0, e^{seq}, w_m) \propto (1-e^{seq})^{y^{ref}}(e^{seq})^{y^{alt}},$$
$$P(y^o_{mi}|x'^o_{mi} = 1, e^{seq}, w_m) \propto (1-w_m)^{y^{ref}}(w_m)^{y^{alt}},$$
$$P(y^o_{mi}|x'^o_{mi} = 2, e^{seq}, w_m) \propto (e^{seq})^{y^{ref}}(1-e^{seq})^{y^{alt}},$$

given the observable genotype x'^o_{mi}. Where no bias is assumed in the allele reads, $w_m = 0.5$, is used as in the previously published models (25, 27, 28). The normalization constant for $P(y^o_{mi}|x'^o_{mi}, e^{seq}, w_m)$ is not shown in the equation above. In addition to sequencing errors and allele observation bias, read mismapping is also a well-known error form in NGS based genotyping. Mismapping leads to the false observation of reads and leads to false heterozygote detection when the true genotype is homozygotic. This type of error is accounted for by summing over x'^o_{mi} in the joint probability $P(y^o_{mi}|x'^o_{mi}, e^{seq}, w_m)P(x'^o_{mi}|x^o_{mi}, e^{map}_m)$. We assume that e^{map}_m can take two values (e^{ref}, e^{alt}), indicating the probability of observing mismapped alternative reads when the true genotype is the same as the reference homozygote and mismapped reference reads when the true genotype is an alternative homozygote. Thus, the probability $P(x'^o_{mi}|x^o_{mi}, e^{map}_m = (e^{ref}, e^{alt}))$ takes the values listed in Table 1.

Similar to the emission probability for offspring, the emission probability for founders can be denoted as $l^f_m = P(\boldsymbol{y}^f_m|\boldsymbol{x}^f_m, e^{seq}, w_m)$ that represents the probability of observing the given read counts in the founders $\boldsymbol{y}^f_m$ given a combination of the true founder genotypes $\boldsymbol{x}^f_m$. It is obtained by

$$\boldsymbol{l}^f_m \propto \prod_j^{N^f} P(y^f_{mj}|x^f_{mj}, e^{seq}, w_m),$$

where the mismapping of reads in the founders is not considered. The normalization constant for $\boldsymbol{l}^f_m$ is not shown. To omit the less probable founder genotypes, the probability $P(y^f_{mj}|x^f_{mj}, e^{seq}, w_m)$ is set at 0 if the result is less than 0.01.

Transition probability. The term transition probability $P(h^o_{mi}|h^o_{m-1,i})$ is used to describe the probability that a crossover occurs along a chromosome. The sequences of the descendent haplotypes are modelled using a continuous-time Markov chain, as described previously (39, 40). The probability that the haplotype state transitions from $h^o_{m-1,i}$ to h^o_{mi} between the markers can be obtained using

$$P(h^o_{mi}|h^o_{m-1,i}) = e^{\boldsymbol{Q}d_m}, m \neq 1,$$

where $\boldsymbol{Q}$ denotes the transition-rate matrix that defines the rates at which a state transition occurs per unit of time. The parameter $\boldsymbol{Q}$ is calculated based on the idea of junction densities following the modelling framework described in (39) and (40). The junction densities define the expected numbers of junction types per Morgan. If $J(abcd)$ represent a junction density of the type $(abcd)$ where ab indicates a pair of descendent haplotypes at a marker and cd denotes that of the next marker located at the other side of the junction. Each of a, b, c, and d can take one of the integers indicating the origin of the haplotype. $J(1112)$ indicates one junction from haplotype 1 to 2 on one chromosome, with the same haplotype maintained on the other chromosome. We can also say the marker before the junction is identical by descent (IBD), while the next is non-IBD. In the case of a biparental population, possible junction types are $J(1111)$, $J(1212)$, $J(2222)$, $J(1112)$, $J(1211)$, $J(1221)$, $J(1122)$, and the junction types in which the order of numbers are swapped, e.g $J(2121)$. As genetic markers are considered sufficiently dense in RRS-based genotype data, only one junction between adjacent markers is expected at most. Therefore, the rate in the transition-rate matrix $\boldsymbol{Q}$ was set to 0 for the junction types with more than one junction e.g., $J(1221)$ and $J(2112)$. Although $J(1122)$ also has two junctions, these are observed in a later generation because the chromosomes are both derived from one identical chromosome that had a junction between haplotypes 1 and 2 in a preceding generation. The possible junctions are selected based on pedigree information. There is no chance that a third haplotype, e.g., $J(1213)$, can be obtained in a biparental population, but it is possible in a population that is derived from four or more founders, such as 8-way RILs.

The matrix exponential is obtained by Higham's algorithm, which is implemented in the "expm" package of R (http://cran.r-project.org/web/packages/ expm/index.html). To obtain transition rates that are proportional to the distances between adjacent markers, $\boldsymbol{Q}$ is multiplied by the genetic distance between markers m and $m-1$ in Morgans d_m. While our algorithm requires genetic distances between markers, genotype data obtained via NGS generally provides only information about the physical distances. Therefore, the obtained d_m is calculated based on the physical distance and the expected genetic distance:

$$d_m = 10^{-6}(p_m - p_{m-1})E^d, m \neq 1,$$

where p_m represents the physical position of marker m and E^d is the expected genetic distance per mega base pairs.

For $m = 1$, the initial probability $\pi(h^o_{1i})$ is required and specified by the stationary distribution of the Markov process that can be obtained by normalizing the first eigenvector of the transposed transition probability for a marker interval.

Estimation of the best haplotype and genotype sequences. Together with the definitions described in the sections above, the joint probability that we need to maximize

DRAFT

can be rewritten as shown below.

$$P(\boldsymbol{H}^o, \boldsymbol{X}^f, \boldsymbol{Y}^o, \boldsymbol{Y}^f) = \prod_{m=1} l^f_m P(\boldsymbol{x}^f_m)$$
$$\times \prod_{i=1} \{ \prod_{m=1} l^o_{mi} \prod_{m=2} P(h^o_{mi}|h^o_{m-1,i}, \boldsymbol{x}^f_{m-1}) P(h^o_{1i}|\boldsymbol{x}^f_1)\}.$$

The Viterbi algorithm is used to maximize the joint probability and estimate the most probable haplotype sequences (41). Considering that the founder genotype itself does not follow the Markov process while the offspring haplotypes depend on the founder genotypes, the most probable founder genotypes at marker m can be estimated based on the scores (probabilities) and the states of the Viterbi trellises that were calculated for the offspring genotype from marker 1 to m. Therefore, our algorithm recursively calculates the Viterbi scores for offspring $v^o_{mi}(h^o_{mi}|\boldsymbol{x}^f_m)$ and founders $v^f_m(\boldsymbol{x}^f_m|\boldsymbol{x}^f_{m-1})$, and records the Viterbi paths for the sequences of offspring haplotypes $\psi^o(h^o_{mi}|\boldsymbol{x}^f_m)$ and founder genotypes $\psi^f(\boldsymbol{x}^f_m)$, respectively. The recursive calculation via an integrated Viterbi algorithm proceeds as follows: Initialization at $m = 1$:

$$v_{1i}(h^o_{1i}|\boldsymbol{x}^f_1) = l_{1i} P(\pi^o_{1i}).$$

Recursion at $2 \leq m \leq M$:

$$\tau_{mi}(h^o_{mi}|h^o_{m-1,i}, \boldsymbol{x}^f_{m-1}) = P(h^o_{mi}|h^o_{m-1,i})$$
$$\times v(h^o_{m-1,i}|\boldsymbol{x}^f_{m-1}),$$
$$\tau'_{mi}(h^o_{mi}|\boldsymbol{x}^f_m, \boldsymbol{x}^f_{m-1}) = l_{mi} \max_{\boldsymbol{h}^o_{m-1,i}} [\tau_{mi}(h^o_{mi}|h^o_{m-1,i}, \boldsymbol{x}^f_{m-1})],$$
$$v^f_m(\boldsymbol{x}^f_m|\boldsymbol{x}^f_{m-1}) = P(\boldsymbol{x}^f_{m-1}) l^f_{m-1}$$
$$\times \prod_i \sum_{h^o_{m,i}} \tau'_{mi}(h^o_{mi}|\boldsymbol{x}^f_m, \boldsymbol{x}^f_{m-1}),$$
$$\psi^f_m(\boldsymbol{x}^f_m) = \mathrm{argmax}_{\boldsymbol{x}^f_{m-1}} [v^f_m(\boldsymbol{x}^f_m|\boldsymbol{x}^f_{m-1})], \psi^o_{mi}(h^o_{mi}|\boldsymbol{x}^f_m) =$$
$$\mathrm{argmax}_{\boldsymbol{h}^o_{m-1,i}} [\tau_{mi}(h^o_{mi}|h^o_{m-1,i}, \boldsymbol{x}^f_{m-1} = \psi^f_m(\boldsymbol{x}^f_m))],$$
$$v_{mi}(h^o_{mi}|\boldsymbol{x}^f_m) = \tau'_{mi}(h^o_{mi}|\boldsymbol{x}^f_m, \boldsymbol{x}^f_{m-1} = \psi^f_m(\boldsymbol{x}^f_m)).$$

Termination at $m = M$:

$$v'^f_M(\boldsymbol{x}^f_M) = P(\boldsymbol{x}^f_M) l^f_M \prod_i \sum_{h^o_{Mi}} v_{Mi}(h^o_{Mi}|\boldsymbol{x}^f_M),$$
$$\hat{\boldsymbol{x}}^f_M = \mathrm{argmax}_{\boldsymbol{x}^f_M} [v'^f_M(\boldsymbol{x}^f_M)],$$

$$\hat{h}^o_{Mi} = \mathrm{argmax}_{h^o_{Mi}} [v_{Mi}(h^o_{Mi}|\hat{\boldsymbol{x}}^f_M)].$$

Backtracking from $M - 1$ to 1:

$$\hat{\boldsymbol{x}}^f_m = \psi^f_{m+1}(\hat{\boldsymbol{x}}^f_{m+1}),$$
$$\hat{h}^o_{mi} = \psi^o_{m+1,i}(\hat{h}^o_{m+1,i}|\hat{\boldsymbol{x}}^f_{m+1}).$$

If multiple arguments have the same maximum score, one is randomly selected. To increase the accuracy, our algorithm executes two rounds of estimation: one from marker 1 to M and one in the reverse direction. The reverse direction round starts from the last marker M and uses the estimated founder genotypes $\hat{\boldsymbol{x}}^f_M$ from the forward round as the only possible founder genotypes at marker M. The results from both rounds are then merged by combining the first half of the estimated genotypes ($m < M/2$) from the reversed direction round with the last half from the forward direction round.

The estimated genotype $\hat{x}^o_{mi}$ at each marker in each sample can now be obtained based on $\hat{h}^o_{mi}$ and $\hat{\boldsymbol{x}}^f_m$. To evaluate the reliability of the estimated genotypes, the marginal probability is also calculated:

$$P(x^o_{mi}|\boldsymbol{Y}^o, \hat{\boldsymbol{X}}^f) = \sum_{h^o_{mi}} P(x^o_{mi}|h^o_{mi}, \hat{\boldsymbol{X}}^f) P(h^o_{mi}|\boldsymbol{Y}^o, \hat{\boldsymbol{X}}^f),$$

where $P(h^o_{mi}|\boldsymbol{Y}^o, \hat{\boldsymbol{X}}^f)$ can be obtained using the forward-backward algorithm (41). The estimated genotype $\hat{x}^o_{mi}$ is set as missing if $P(\hat{x}^o_{mi}|\boldsymbol{Y}^o, \hat{\boldsymbol{X}}^f)$ is smaller than a user specified threshold P_{call}, with 0.9 as the default.

Estimation of allele read bias and mismap rate. GBScleanR requires four parameters E^d, e^{seq}, e^{map}_m, and w_m. The expected genetic distance per mega base pairs E^d and sequencing error rate e^{seq} take a single positive number that should arbitrarily be specified, while the mismap rates e^{map}_m and the allele read biases w_m can be estimated. To obtain precise estimations of the parameters, the algorithm iteratively performs genotype estimation via the Viterbi algorithm, as described in the previous section, and parameter estimation via the procedure described here. For the first cycle of genotype estimation, $e^{map}_m = (e^{ref}, e^{alt})$, and w_m are set to $(0, 0)$ and 0.5 for all markers, respectively. Thus, the first iteration does not consider allele read biases and mismapped reads, although e^{seq} can partially account for mismapped reads. With the estimated genotypes from the first cycle, we can obtain the numbers of reads for each estimated genotype. Let y^{ref}_{mi} and y^{alt}_{mi} be the number of reference reads and alternative reads at marker m in sample i including both offspring and founders. The mismap rate at marker m can be obtained using:

$$e^{ref} = \frac{\sum_i [\delta(y^{alt}_{mi} > 0)\delta(x^o_{mi} = 0)]}{\sum_i [\delta(x^o_{mi} = 0)]},$$

$$e^{alt} = \frac{\sum_i [\delta(y^{ref}_{mi} > 0)\delta(x^o_{mi} = 2)]}{\sum_i [\delta(x^o_{mi} = 2)]},$$

where δ is an indicator function that is equal to 1 if the argument is true and 0 if it is false.

Similarly, the allele read bias at marker m is estimated by the

DRAFT

following:

$$w_m = \frac{E_m^{ref}}{E_m^{ref} + E_m^{alt}},$$
$$E_m^{ref} = \frac{r_m^{ref}(0) + r_m^{ref}(1)}{n_m(0) + n_m(1)},$$
$$E_m^{alt} = \frac{r_m^{alt}(2) + r_m^{alt}(1)}{n_m(2) + n_m(1)},$$
$$r_m^x(g) = \sum_i [y_{mi}^x \delta(x_{mi}^o = g)], x \in \{ref, alt\},$$
$$n_m(g) = 2^{\delta(g \neq 1)} \sum_i [\delta(x_{mi}^o = g)],$$

where g is 0, 1, or 2 and indicates the genotype; E_m^{ref} represents the expected number of reference allele reads observed if one reference allele is on the chromosome, and E_m^{alt} represents the expected number of alternative allele reads if one alternative allele is on the chromosome. This calculation relies on the assumption that the reads at each marker in both heterozygous samples and homozygous samples can be biased if samples are prepared and sequenced together. To confirm if the given data meets this assumption, our algorithm first calculates $\boldsymbol{w} = \{w_m\}_{m=1...M}$ with $g = 1$ and $g = \{0, 2\}$ separately, and then evaluates the correlation between $\boldsymbol{w}$ obtained in both calculations. If the correlation coefficient is greater than or equal to 0.7, the equation above is employed; otherwise, $\boldsymbol{w}$ used with g = 1, although most of the datasets used in this paper meet the assumption of bias uniformity, even the real dataset.

Preparation of real data. The genotype data of an F_2 population that was derived from a cross between *Oryza sativa* and *O. longistaminata* were used in this study. The data potentially include a large number of error prone markers with mismap and allele read biases, as described in our previous paper (42). In brief, the F_2 population was produced via the self-pollination of F_1 plants that were derived from a cross between *O. sativa* ssp. *japonica* cv. Nipponbare and *O. longistaminata* acc. IRGC110404. GBS was performed with a *Kpn*I-*Msp*I restriction enzyme pair using MiSeq with 75 bp paired-end sequencing. Sequencing runs with nine 96-multiplex libraries generated an average $134,447 \pm 50,788$ reads per sample. We obtained a total of 2,539,459 and 3,481,218 reads for *O. sativa* and *O. longistaminata* from the independent runs, respectively, which should be sufficient for fairly accurate determination of the founder genotypes. The obtained reads were then processed via the TASSEL-GBS pipeline v2 by following the instructions in the manual with the default parameters (https://bitbucket.org/tasseladmin/tassel-5-source/wiki/Tassel5GBSv2Pipeline)(34). The obtained SNP markers were then filtered to retain only those markers that were homozygous in each founder but biallelic between founders. Only the first SNP was selected if multiple SNPs were present within a 75 bp stretch. Considering the distant cross that was used to obtain the F_2 population, our genotype data might include mismap-prone markers due to the repetitive sequences in the two founder genomes. The mismapping patterns can differ between samples which have different descendent genome compositions. Therefore, to filter out the erroneous genotype calls that were caused by mismapping reads from repetitive sequences, both the reference and the alternative allele read counts were set to 0 for genotype calls where either the reference or the alternative allele reads were higher than the 90th percentile for each genotype call in each sample, respectively. The resulting genotype data had 5035 SNP markers for 814 F_2 individuals with an average read depth of 0.85×.

Generation of simulation data. Genotype data were simulated for three scenarios: a 2-way F_2 population from inbred founders (homoP2_F2), a 2-way F_2 population from outbred founders (hetP2_F2), and 8-way RILs from inbred founders (homoP8_RIL). For each scenario, several datasets were created using different numbers of offspring and markers together with different expected read depths for both offspring and founders, as listed in Table 2. To generate the descendent haplotype and genotype data for each dataset in each scenario, the breeding pedigree was first simulated via the following. The homoP2_F2 scenario assumes that the two inbred founders have homozygotes at all markers and are biallelic between founders, the founders are crossed and then self-pollinate, as usually conducted in the genetic analysis of self-fertilizing plants. The hetP2_F2 scenario assumes two outbred founders with independently random genotypes at any marker with the minor allele observed in at least one of the four possible haplotypes. These outbred founders are subjected to mating to produce 1000 progenies and then 100 sibling combinations are used to produce 10 progenies in each. The last generation of hetP2_F2 has 1000 offspring in total. This scenario assumes the case of genetic analyses for non-self-fertilizing organisms. The pedigree of homoP8_RIL consists of three pairing generations and five self-fertilized generations. In the pairing stage, four pairs are selected from the eight founders with no duplication, and one descendant is created from each pair. The four descendants are then further paired to produce 100 progenies from each of the two pairs. At this step, we have two sibling families with 100 individuals in each. At the last pairing, 50 pairs are produced by selecting one individual per sibling family to produce pairs from each of which 50 progenies are generated. The resultant 2500 individuals are then self-fertilized five times so that one offspring is produced per individual in order to maintain the population size. This kind of pedigree is usually found in populations that have been subjected to genetic survey and breeding and known as Multiparent Advanced Generation Inter-Cross (MAGIC) population. A given number of offspring is sampled from the last generation to prepare each dataset.

Descendent haplotype data from each population was generated by introducing crossovers into the gametes that have descended from the parent(s) of each progeny at each mating step. Only one pair of 50 Mb chromosomes as diploid genome was simulated for all datasets and the expected ge-

DRAFT

netic distance per mega base pairs was set to 0.04. Markers were located randomly throughout the chromosome. The occurrence of crossovers between adjacent markers in a gamete follows Poisson distribution with a mean value that is equal to the genetic distance of the given pair of adjacent markers. The genetic distances between the markers were calculated by multiplying the physical distances by 0.04×10^{-6}. If an odd number of crossovers was assigned to a marker pair, gamete switching descendent haplotypes were generated, while the assigned crossover was ignored if the donor parent of the gamete showed IBD around the markers. The offspring genotypes were obtained by replacing the descendent haplotype at each marker with the genotype that was assigned to the founder haplotype.

Based on the simulated genotype data, read counts were assigned to each marker of each individual. For each of the three scenarios, two situations were considered to simulate the read count data. One was simulated with the assumption of no allele read bias and no mismap (noError datasets), while the other was simulated by mimicking the allele read biases observed in the real data (realError datasets). Generally, the total number of reads in read count data that is obtained via NGS is competitively allocated to markers of sequenced samples with biases that depend on the proportion of sequencing targets in a sequencing library. To mimic the competitiveness of reads between markers of samples, reads were allocated following the joint probability for all markers and samples as explained below.

The total number of reads per marker y_m^T is assumed to follow exponential distribution with the rate parameter λ, $Exp(\lambda)$. The probability of a read at marker m can be obtained by the following:

$$P(m) = \frac{y_m^T}{\sum y_m^T},\ y_m^T \sim Exp(\lambda),$$

where λ is set at 0.25. The read obtained at marker m can be either reference or alternative with probabilities of p^{ref} and $(1-p^{ref})$, respectively, where p^{ref} represents the probability that the read is a reference allele. The probability p^{ref} takes different values depending on the genotype at marker m in sample i, x_{mi}. When the genotype is the homozygous reference, $p^{ref} = (1-e^{seq})$, where e^{seq} represents a sequencing error. Similarly, $p^{ref} = 0.5$ and $p^{ref} = e^{seq}$ for heterozygous and homozygous alternatives. To incorporate allele read bias, the genotype-dependent read observation probability $P(z|w_m, x_{mi}), z \in \{ref, alt\}$ was introduced, which takes the values shown in Table 3. The joint probability of having a read for allele z at marker m in sample i can then be calculated:

$$P(z,m,i|w_m,x_{mi}) = P(z|w_m,x_{mi})P(m)P(i),$$
$$P(i) = \frac{1}{N},\ z \in \{ref, alt\},$$

where N is the number of samples. Mismapping error is introduced by replacing the genotypes with the probability $P(x'_{mi}|x_{mi}, e_m^{map} = (e^{ref}, e^{alt}))$, taking the values listed in Table 1. The number of total reads is calculated by multiplying the number of samples by the expected read depth and the number of markers. We then allocate those reads based on the joint probability $P(z,m,i|w_m,x'_{mi})$, which is not conditional on the true genotype x_{mi}. The read count data for the founders and the offspring are simulated separately with given expected read depths for each datasets.

For the realError datasets, the number of markers was set to the same number as that in the real data. $P(m)$ was also obtained from the read data by calculating the total number of reads per marker y_m^T by summing up the read counts for each marker. The values w_m were obtained by calculating $y_m^{ref}/(y_m^{ref} + y_m^{alt})$, where y_m^{ref} and y_m^{alt} denote the total numbers of reference and alternative reads per marker observed in the real data, respectively. The mismapping rate was randomly assigned to markers by sampling the probabilities from an exponential distribution with the mean at 0.05, since the mismapping pattern in the real data is unknown. On the other hand, w_m and e_m^{map} were set to 0.5 and $(0,0)$, respectively, for all markers in the noError datasets. The sequencing error rate e^{seq} was set to 0.0025 for both the noError and the realError datasets. The realError datasets were simulated using only real data from chromosome 7, which showed relatively severe bias (Supplemental Figure S2). Each realError dataset is therefore simulated to consist of 620 markers on one chromosome with physical lengths of 50 Mb and genetic lengths of 2 Morgans.

Evaluation of algorithms. The algorithms implemented in three tools were tested in this study; the previously published error correction tools LB-impute and magicImpute (25, 28) and the novel tool GBScleanR. The genotype estimation accuracy of each algorithm for the simulated datasets is evaluated by calculating the proportions of correctly estimated genotypes (correct call rate), wrongly estimated genotypes (miscall rate), and missing genotype calls (missing call rate) in offspring.

The default settings were used for the majority of arguments in LB-impute and magicImpute with some dataset-specific settings as explained below. In the case of LB-impute, the "impute" option was set to the "-method" argument and the "-offspringimpute" flag was used to run it in the offspring impute mode. A vector of strings stating the names of the founder samples was specified as "-parents". The arguments in magicImpute include the mandatory arguments "model" and "popdesign" in addition to the genotype data in the "magicsnp" format. The simulated data and real data in the VCF format were converted to the magicsnp format. Marker positions in centimorgan are required for magicImpute. Thus, we calculated those by multiplying the physical positions of the markers by the expected genetic distance per mega base of 0.04, which was used to generate the simulation data. The "jointModel" option was set to the "model" argument for all datasets. The "popdesign" argument that affects on the state transition-rates was specified based on the pedigree in each scenario as follows. The popdesign was set to ("Pairing", "Selfing") for the scenario homoP2_F2, while ("Pairing",

DRAFT

"Pairing", "Pairing", "Selfing", "Selfing", "Selfing", "Selfing", "Selfing") was specified for homoP8_RIL. Although the popdesign for hetP2_F2 was supposed to be ("Pairing", "Sibling") based on the pedigree and the instructions in the manual, magicImpute did not accept it and returned an error message saying "invalid popdesign". Therefore, ("Sibling", "Sibling") was used for the hetP2_F2 scenario, which is expected to be the proper choice according to the calculation procedure of the recombination numbers described in the paper (39) that is supposed to be implemented in magicImpute. The estimated genotype data generated by magicImpute includes an incomplete estimation of the genotype represented by 1N, N1, 2N, and N2, where N points to an unknown allele and the numbers 1 and 2 indicate reference and alternative alleles, respectively. Although these incomplete estimations provide partial information about the genotype at the markers, we treat them as completely missing because downsream genetic analyses generally cannot handle such incomplete information.

GBScleanR includes some arguments that can be used to modify its behaviour. Similar to magicImpute, pedigree information is also required in GBScleanR and was set as simulated for each dataset. Other arguments that can be used to tweak this tool are the expected genetic distance per mega base pairs E^d, sequencing error rate e^{seq}, threshold for the probability of calling estimated genotype P_{call}, and the number of iterations required for genotype estimation. To decide the default values for these arguments, we tested the algorithm using the realError dataset with 100 offspring, 1000 markers, and $3\times$ read depth in the homoP2_F2 scenario under various settings. As shown in Supplemental Figure S3, the tests indicated that P_{call} have relatively large impacts on the genotype estimation accuracy, while the differences associated with the number of iterations (if more than one), E^d, and e^{seq} were negligible. The values of $E^d = 0.04$ and $e^{seq} = 0.0025$ that were used in the data simulation seem to be good choices when set for genotype estimation. Increasing P_{call} from 0.8 to 0.95 improved the correct call rate in an almost proportional manner, while the miscall rate worsened at the same time. Therefore, we set P_{call} at 0.9 for the algorithm evaluation as used in the default setting for magicImpute. In the case of iteration number, we found that a larger number leads to better correct call and miscall rates, which reached a plateau at four or more iterations. Thus, we decided to set the number of iterations to four. To evaluate GBScleanR, we tested the algorithm with and without the iterative optimization of the parameters w_m and e_m^{map}.

For the evaluation of algorithms on simulation datasets, correct call rates, miscall rates, and missing rates were calculated for each offspring in the datasets and then the means and standard deviations of these values were obtained for each dataset processed by each algorithm. The means and standard deviations are shown in S1-S6 of the Supplemental Data. In the case of the real data, we first selected relatively reliable genotype calls supported by more than 10 reads at markers that have less than 5% of missing rate and more than 40% of minor allele frequency. Those selected genotype calls were then set to missing by removing all read counts observed for them. This masked real data was subjected to genotype estimation using GBScleanR, LB-Impute, and magicImpute. The resultant estimated genotype data was evaluated based on the concordance rate that is the proportion of estimated genotype calls matching with the raw genotype calls at the masked genotype calls. Furthermore, we also calculated the averages of missing rate, the number of recombinations, and segregation distortion level in each chromosome. The segregation distortion level at each marker was obtained by dividing the chi-square value, which was calculated based on the assumption that the genotype ratio to be 0.25, 0.5, and 0.25 for three possible genotypes, by the number of non missing genotype calls.

RESULTS

Evaluation using simulation datasets with no allele read bias. We first evaluated the three algorithms in the simulation datasets assuming no allele read bias (noError datasets) in the scenarios homoP2_F2, hetP2_F2, and homoP8_RIL. If 100 offspring were simulated in the homoP2_F2 scenario, GBScleanR and magicImpute showed similar correct call rates, with GBScleanR performing slightly better in most cases (Figure 1A, D and G and Supplemental Data S1). In addition, both GBScleanR and magicImpute were found able to improve the correct call rates using 1000 markers as compared to the performance with 100 and 10 markers, while LB-Impute showed a limited improvement in terms of correct call rates with increased marker numbers. LB-Impute is limited with regards to the marker number used in the Viterbi trellis for genotype estimation, for which seven markers are used as default. This limitation probably causes the difference observed.

Since LB-Impute only accepts biparental populations with inbred founders, the datasets hetP2_F2 and homoP8_RIL were subjected to error correction in GBScleanR and magicImpute. Unlike the cases in the homoP2_F2 scenario, GBScleanR scored approximately 10 percentage points higher in terms of correct call rates in the hetP2_F2 scenario when the offspring read depth was $\geq 0.5\times$ and 50 or 100 markers were assumed for 100 offspring (Figure 1B and E and Supplemental Data S2). GBScleanR and magicImpute scored similar correct rates for the datasets with 1000 markers, although magicImpute showed unexpectedly lower accuracy for the dataset with $1\times$ offspring read depth and $3\times$ founder read depth compared with the other conditions (Figure 1H and Supplemental Data S2). The genotype estimation algorithms implemented in GBScleanR and magicImpute randomly select one of the possible genotype patterns if multiple patterns are equally likely to occur. Especially, the genotype estimation of outbred founders can have equally possible genotype patterns when founders do not have enough read counts to determine their genotypes unambiguously. Even considering the stochastic occurrences of less accurate estimations, GBScleanR demonstrated more robust estimation than magiImpute. On the other hand, unlike in the F2 populations, we observed only negligible differences between GBScleanR and

DRAFT

magicImpute in the homoP8_RIL scenario (Figure 1C, F, and I and Supplemental Data S3).

Even if the offspring number were changed to 10 or 1000, the trends observed in the cases that simulated 100 offspring showed no change (Supplemental Figure S4-S5 and Supplemental Data S1-S3). We found that the correct call rates that were obtained using magicImpute for datasets with fewer markers and offspring reads in the hetP2_F2 scenario were slightly higher than those obtained by GBScleanR (Figure 1B and E and Supplemental Figure S5B, E, and H). Nevertheless, the miscall rates obtained with magicImpute were also higher than those observed using GBScleanR for the same datasets (Supplemental Data S2). GBScleanR assumes all markers are biallelic and does not accept founders that have the same homozygous genotype at any marker. On the other hand, the results from magicImpute include markers at which all founders are estimated to have the same homozygous genotype. Compared with the implementation in GBScleanR, the use of magicImpute seems to result in fewer missing calls but leads to more miscalls when only limited amount of information is available for genotype estimation. No particular difference was observed whether the iterative parameter optimization was used or not for all cases in which no allele read bias was assumed (Figure 1 and Supplemental Figure S4-S5). The absolute scores for correct call rates and miscall rates were largely improved by increasing the number of offspring in GBSleanR and magicImpute, whereas decreasing the number of offspring appeared to worsen the scores (Supplemental Data S1-S3).

Effects of allele read bias on genotype estimation accuracy. To compare the robustness of the algorithms against allele read bias, genotype estimation was also conducted for the realError datasets. If no allele read bias was observed, the proportion of the reference allele reads per marker would be expected to follow normal distribution with a mean of 0.5. However, the values that were derived from the real data showed a highly biased distribution, with a large number of markers showing bias towards more reference allele reads than expected (Supplemental Figure S2). Some markers also showed more alternative than reference allele reads. Under the condition of such severe biases, the genotype estimations made by the algorithms was highly disturbed, except for GBScleanR under the iterative parameter optimization (Figure 2, Supplemental Figure S6-S7, and Supplemental Data S4-S6). In the homoP2_F2 scenario with 100 offspring, GBScleanR with the iterative parameter optimization scored the best for all conditions and achieved a correct call rate of more than 95% at maximum when the data included rich reads for offspring (Figure 2A and Supplemental Data S4). On the other hand, magicImpute scored only 15-20 percentage points lower than GBScleanR in terms of correct call rates under all conditions in the homoP2_F2 scenario and reached an accuracy of only 82.5% at most. LB-Impute completely failed to estimate the offspring genotypes using biased data. Unlike the cases of the noError datasets, GBScleanR with the iterative parameter optimization improved the offspring genotype estimation accuracy and the improvement was larger in datasets with more offspring reads. Interestingly, the GBScleanR algorithm showed better accuracy than magicImpute for datasets with relatively few offspring reads, even without the iterative optimization. Increasing the offspring number led to a decrease of the difference between the correct call rates obtained by GBScleanR and maigcImpute, while the differences increased in datasets with less offspring, especially those with fewer offspring reads (Figure 2A and Supplemental Figure S6A and S7A). Nonetheless, GBScleanR outperformed the other algorithms in most cases.

In hetP2_F2 with 100 offspring, GBScleanR generally scored much better than magicImpute with regards to correct call rates, although magicImpute showed the highest correct call rates for the datasets with $0.1\times$ offspring read depth (Figure 2B and Supplemental Data S5). Similar to the noError datasets, the better correct call rates obtained by magicImpute were accompanied by worse miscall rates than those achieved by GBScleanR. This trend did not change for the datasets with 1000 offspring (Supplemental Figure S6B). On the other hand, GBScleanR and magicImpute showed similar accuracy in the case with 10 offspring, although GBScleanR obtained superior correct call rates when offspring had a read depth of more than $0.5\times$ (Supplemental Figure S7B).

As only negligible differences were observed for the noError datasets in the homoP8_RIL scenario, little difference was observed in the genotype estimation accuracy for the RIL datasets with 100 offspring, even with allele read biases (Figure 2C and Supplemental Data S6). The difference between GBScleanR and magicImpute in terms of correct call rates was a maximum of approximately 3 percentage points in the dataset with $0.1\times$ offspring reads and $3\times$ founder reads (Figure 2C and Supplemental Data S6). Similar results were observed for the datasets with 10 and 1000 offspring (Supplemental Figure S6C and S7C). The iterative parameter optimization of GBScleanR did not improve the estimation accuracy in the RIL datasets. Since the offspring in the homoP8_RIL scenario had homozygous genotypes at almost all markers due to repeated self-fertilization and only the reads denoting either allele could be observed at each genotype call, allele read bias is unlikely to have affected the genotype estimation.

Less biased estimation by GBScleanR. We further checked the quality of the estimated genotype data by visualizing the estimated genotype ratio at each marker. This visual inspection employed a dataset of 100 offspring with $3\times$ reads depth for offspring and founders in the homoP2_F2 scenario as a 1:2:1 genotype ratio always can be expected in offspring. As shown in Figure 3, the estimations by GBScleanR showed genotype ratios per marker that were similar to those obtained for the true ratios. In addition, miscalls in the estimated genotype data were distributed almost uniformly throughout the entire chromosome with very few miscalls per marker (Figure 3B). In contrast, LB-Impute showed severely biased genotype ratios with high miscall rates clustered at several loci on the chromosome (Figure 3C). Although the degree of bias was smaller than that observed for

DRAFT

LB-Impute, magicImpute also generated biased genotype estimations at several loci on the chromosome, where more reference homozygotes and less heterozygotes were called than the true data (Figure 3D). We found that both LB-Impute and magicImpute estimated the same genotype for all founders at some of the markers, although all markers were expected to be biallelic (gray lines in Figure 3C and D). Those loci resulted in high miscall rates, especially for magicImpute (Figure 3D). The loci showing higher miscall rates overlapped with the plots for LB-Impute and magicImpute. Thus, these miscalls and biased genotype estimations were probably a result of error prone markers at which severe allele read biases were clustered. Similar results were observed in the dataset with 1000 offspring (Supplemental Figure S8). The increased number of offspring could not improve the biased estimation by magicImpute and LB-Impute (Supplemental Figure S8C and D). However, for the dataset with 10 offspring, GBScleanR produced miscalls clustered at several loci, as did magicImpute (Supplemental Figure S9). GBScleanR optimizes the allele read bias parameter w_m based on observation and estimates the genotypes from the given data. Thus, using a smaller number of offspring results in a less precise estimation of the biases. Even though GBScleanR showed a reduction in the estimation accuracy with fewer offspring, the highest correct call rates and the lowest miscall rates were scored by GBScleanR for the datasets that were investigated in the visual inspection (Supplemental Data S4).

Estimation of founder genotype and phased genotype. GBScleanR and magicImpute have been designed to simultaneously estimate the genotype of both founder and offspring. Thus, we next compared the founder genotype estimation accuracy of the algorithms. As expected from the estimation accuracy for offspring, GBScleanR outperformed magicImpute in most cases (Figure 4 and Supplemental Figure S10). For the noError datasets, GBScleanR and magicImpute showed similar accuracy and both were unstable with low founder read depth ($1\times$) in the hetP2_F2 scenario (Supplemental Figure S10). The lower number of offspring in the given datasets also resulted in a reduction in the accuracy when estimating founder genotype. Similar effects were observed in the number of markers. Therefore, larger numbers of offspring and markers give more information from which founder genotypes can be precisely estimated, resulting in a more precise estimation of the offspring genotype.

The results achieved using the GBScleanR algorithm with the realError datasets also showed good performance (Figure 4). On the other hand, the iterative parameter optimization did not affect the founder genotype estimation in the homoP2_F2 or the homoP8_RIL scenarios (Figure 4A, D, G, C, F, and I). Improvements in the estimation accuracy by the iterative parameter optimization were observed only in the hetP2_F2 scenario with a relatively large number of offspring and offspring reads (Figure 4B, E, and H). This might be because heterozygotes are only assumed in the founder genotypes under the hetP2_F2 scenario but not in the other scenario; the homozygous genotypes of the inbred founders were not affected by allele read biases. Our simulation also indicated that estimating the genotype of the two outbred founders in an F_2 population might be more difficult than estimating the genotypes of the eight inbred founders in a RIL population.

In addition to the genotype information of offspring and founders, both GBScleanR and magicImpute provide phasing information for the offspring genotypes. To evaluate the phasing accuracy, phased haplotype information that was obtained via GBScleanR was converted to phased genotype information according to the estimated founder genotypes; this was performed because GBScleanR outputs only phased haplotype information and does not output phased genotypes as default even though it internally calculates phased genotypes. On the other hand, magicImpute outputs phased genotype data. The phased genotype data were then compared with the true phased genotypes to calculate the correct call rates. As expected from the superiority for both founder and offspring genotype estimation, GBScleanR outperformed magicImpute in the realError datasets, while they showed similar accuracy in the noError datasets (Supplemental Figure S11 and S12). Although the founder read depth appears to have not particularly affected the accuracy in most cases, datasets with a $3\times$ founder read depth resulted in slightly better estimations than $1\times$ when only 10 offspring were included, regardless of the algorithm used.

Evaluation using the real data. We finally evaluated the algorithms using the real data. As usually conducted to evaluate the performances of genotpye error correction tools for real data, the concordance rate of the estimated genotype calls was measured using the masked genotype data derived from the real data. First of all, GBScleanR scored the best concordance rate in all of the 12 chromosomes in the comparison with LB-Impute and magicImpute (Figure 5A and Supplemental Data S7). Although the same ranking, in which GBScleanR is the top followed by magicImpute, was observed in all chromosomes, the relative difference between the concordance rate of GBScleanR and magicImpute differed in each chromosome. The largest difference was observed in chromosome 9 and reached 13.3 percentage points higher in GBScleanR. On the other hand, chromosome 5 showed only a tiny difference in the concordance rate that was 0.9 percentage points between GBScleanR and magicImpute. Chromosome 1, 3 and, 6 recorded relatively higher concordance rate that were more than 90% by GBScleanR, while chromosome 11 reached 71.2% at most. The chromosomes with a severer allele read bias, e.g. chromosome 7-12, seems to tend to show a less concordance rate (Supplemental Figure S2).

Even though GBScleanR showed the highest concordance rate, the genotype concordance does not directly mean the higher accuracy because we cannot know the true genotype of the real data and there is no evidence whether the genotype calls selected for masking were truly reliable or contained a large number of erroneous genotype calls that unexpectedly passed through the filtering. To further evaluate the performance of the algorithms, we then calculated the averages of missing rates and the number of recombinations in each chromosome. In the case of missing rate, the lowest values were

DRAFT

scored by GBScleanR for all chromosomes, while the tool at the second lowest differed in the chromosomes (Figure 5B and Supplemental Data S8). Similar with the case of the concordance rate, the chromosomes having severer allele read biases showed relatively higher missing rates. A prominent difference between the tools was observed in the number of recombinations (Figure 5C and Supplemental Data S9). The magicImpute algorithm estimated much higher numbers of recombinations especially in chromosome 2, where the average and the maximum numbers of recombinations per sample were 21.7 and 46 (Supplemental Data S8). On the other hand, the genotype data estimated by GBScleanR and LB-Impute have 1-2 and 1.8-3.1 recombinations per chromosome on average, respectively (Supplemental Data S8). The previous study reported that the shortest and longest chromosomes in the estimated genetic map of rice are chromosome 10 and chromosome 1, which are 83.8 and 181.8 cM, respectively (43). Therefore, the expected numbers of recombinations per chromosome would range 1.6-3.6. The results of LB-Impute seemed to fit to the expected values more than those of GBScleanR. To confirm whether LB-Impute truly estimated more accurate numbers of recombination events, we measured the genomic segment lengths between the recombination breakpoints in chromosomes (Supplemental Figure S13 and Supplemental Data S10). We found that LB-Impute estimated much more numbers of short segments than expected. The genotype data estimated by GBScleanR had 550 and 2099 segments in the ranges of 0-1 and 1-2 Mb, which account for 2% and 8% of the total observed genomic segments, respectively. On the other hand, LB-Impute estimated the genotype data with 1352 (4%) and 4300 (12%) of 0-1 and 1-2 Mb segments, respectively. In addition, GBScleanR and LB-Impute had 27 and 172 of the 0-1 Mb segments that were heterozygous and flanked by the same homozygous genotypes at the both sides, which indicate double crossovers within 1 Mb stretches. The probability of a double crossover within 1 Mb is approximately 0.08% when 1 Mb equals to 4 cM. Thus, the expected number of double crossovers is less than 2 in the population with 814 individuals derived from 1628 gametes. These results suggested that GBScleanR estimated more probable and acceptable numbers of recombinations as F_2 population, while an unexpectedly large number of short segments were estimated by LB-Impute and magicImpute and pushed up the number of recombinations artificially.

At last, we measured the segregation distortion level in the estimated genotype data. Since our real data was obtained from the rice F_2 population, it could be expected that any markers show a genotype segregation ratio of 1:2:1 for reference homozygotes, heterozygotes, and alternative homozygotes although segregation distortion due to the reproductive isolation could occur. Relatively low distortion level was observed in the genotype data obtained via GBScleanR compared with those estimated by LB-Impute and magicImpute (Figure 5D). The largest difference was observed in chromosome 8 where magicImpute and LB-Impute resulted in 4.9- and 13.2-fold higher distortion levels than GBScleanR did (Supplemental Data S10). Interestingly, chromosome 1 was the most distorted one in the genotype data estimated by GBScleanR, while the other algorithms also showed similar distortion levels in this chromosome (Figure 5D and Supplemental Data S8). To visually inspect the segregation distortion levels, we created the line plots of the estimated genotype ratios (Figure 6 and Supplemental Figure S14). Figure 6 shows the plots for chromosomes 1 and 8, which were selected to contrast the negative impact of allele read bias and also to confirm the reason of the distortions observed at similar levels regardless the algorithm. The line plots indicated that all of the algorithms estimated quite similar genotypes for chromosome 1, while clearer differences were observed in the plots for chromosome 8 (Figure 6). As fewer markers in chromosome 1 showed sever allele read biases, while chromosome 8 showed relatively higher number of markers preferentially accumulated reference allele reads, chromosome 1 might have less error prone markers (Supplemental Figure S2). This relatively less error prone chromosome does not seem to have given sever negative impact on the estimation by LB-Impute and magicImpute, and resulted in the similar estimation with GBScleanR (Figure 6A-C). In addition, the distortion observed in chromosome 1 of Figure 5D was not due to the misestimation but the true genetic characteristic of this population. In the case of chromosome 8, however, LB-Impute and magiImpute obviously failed in estimating the genotypes and produced artificially distorted genotype ratio patterns (Figure 6D-E). In contrast, GBScleanR showed an acceptable pattern as the genotype ratio of a F_2 population throughout the chromosome (Figure 6F). The other chromosomes also exhibited similar results to those observed for chromosomes 1 and 8 and demonstrated the superiority and robustness of GBScleanR against error prone markers (Supplemental Figure S14).

Comparing the running times. In addition to the comparisons of the algorithms in the aspects of accuracy and robustness, we also compared the running times for the noError simulation datasets. As shown in Table 4, GBScleanR completed all processes of genotype estimation faster than LB-Impute and magicImpute for the simulation datasets of biparental F_2 populations. In the case of the largest dataset that consists 1000 individuals and 1000 markers, GBScleanR took only 27 seconds even with the iterative parameter optimization, while magicImpute and LB-Impute consumed 5.3 and 105.7 times as long. Even for the simulation datasets of HetP2_F2 , which require more numbers of calculations than HomoP2_F2, GBScleanR with the iterative estimation outperormed magicImpute. On the other hand, a 1.5 times longer calculation at maximum was observed in GBScleanR with four iterative cycles of the parameter optimization compared with magicImpute for the homoP8_RIL datasets. Nevertheless, the iterative parameter optimization was not required basically for RIL populations as shown in Figure 2, and GBScleanR without the iterative parameter optimization could finish the calculations within less than half the running times of magicImpute.

DRAFT

DISCUSSION

Advantage of GBScleanR. The present study introduced a novel algorithm to provide robust error correction on genotype data with error prone markers that show biased probabilities for allele read acquisition. Any genotype data derived from GBS, or other RRS-based genotyping systems, has a chance to include such error prone markers even after erroneous markers were filtered out. Particularly, a population derived from a cross between distant relatives, as in our case in which a cultivated rice *O. sativa* and a wild rice *O. longistaminata* were crossed, may potentially have a relatively high proportion of error prone markers. If a model uniformly assumes a 50% chance of obtaining a read for either allele for all markers at a heterozygous site, any HMM-based methods and sliding window methods can theoretically be deceived by the error prone markers.

Our simulation studies showed the advantage of using GBScleanR to precisely estimate the true genotype by incorporating the unevenness of allele reads into the model when using noisy GBS data (Figure 2 and 3). Although the simulation datasets used in this study were created with relatively sever settings in the number of markers and read counts available for the genotype estimation, GBScleanR basically outperformed the other algorithms in any situation and resulted in the most stable and accurate genotype estimation even for the datasets with extremely poor information like a dataset with $0.1\times$ reads at only 50 markers of 10 individuals (Supplemental Figure S5 and S7). When the reads more than $3\times$ at 100-1000 markers for more than 100 individuals could be provided for a 2-way F_2 population derived from inbred founders, GBScleanR are able to make nearly 100% accurate estimation even in the datasets with error prone markers (Figure 1 and 2). On the other hand, 8-way RIL populations and 2-way populations derived from outbred founders seem to require more numbers of markers for precise estimation (Figure 1, 2, and Supplemental Figure S4-S7). To demonstrate the reliability and robustness of GBScleanR, we next evaluated the algorithms using the real data that potentially contains error prone markers. The result clearly illustrated the negative impact of error prone markers having sever allele read biases and how the iterative estimation with error pattern recognition implemented in GBScleanR outperforms the existing algorithms (Figure 5, 6 and Supplemental Figure S13-S14). The estimated genotype data for the real data by LB-Impute resulted in the lowest concordance rate and highest distortion level compared with the other algorithms (Figure 5). Together with the relatively poor correct call rate in the simulation data (Figure 2), the LB-Impute algorithm seems to be highly affected by biased read counts at the error prone markers and fill up wrong genotype calls at surrounding markers. Therefore, LB-Impute may deteriorate genotype data quality by enhancing wrong genotype calls using erroneous information of biased read counts if given genotype data contains error prone markers. Unlike LB-Impute, the magicImpute algorithm resulted in the extreme numbers of recombinations, while it scored better concordance rates and distortion levels (Figure 5). The genotype ratios visualized in Figure 6 indicates that magicImpute is able to keep the negative effects of biased read counts within limited stretches nearby error prone markers. On the other hand, GBScleanR could score the best concordance rates and the lowest missing rates in all chromosomes (Figure 5). In addition, although the numbers of recombinations found in each chromosome did not fit the expected numbers according to the genetic map of rice (43), the lowest number of the improbable double recombinations within short stretches was observed in the genotype data estimated by GBScleanR (Supplemental Figure S13). These results, together with the higher accuracy in the simulation datasets, indicates that GBScleanR is able to precisely incorporate the information of biased read counts at error prone markers and give robust and accurate estimation. The plots of the estimated genotype ratios also illustrate that GBScleanR can largely reduce the negative impacts of biased read counts and minimize artificial distortions even in the chromosomes containing a relatively large number of error prone markers (Figure 6 and Supplemental Figure S14).

The distortion level and the recombination frequency have large effects on downstream genetic analyses using genotype data and also those values themselves are research targets. Any genetic and genomic study relies on the given genotype data and the accuracy of genotyping has a crucial effect on the quality of such research. The accuracy of genotyping can be evaluated by three criteria: correct calls, miscalls, and missing calls. Usually, miscalls have a more destructive effect on post-genotyping analyses than missing calls, because post-genotyping analyses regularly suppose all genotype calls to be true even if those include miscalls, and potentially generate misleading results. Furthermore, if the wrongly genotyped markers are clustered in a particular region, all statistical calculations for that region will be unreliable. Our study demonstrated the reliability of GBScleanR as the support tool for genetic studies that handle genotype data even with or without error prone markers.

GBScleanR requires no strict filtering. In this study, we conducted a filtering based on a quantile of read counts for the real data to filter out genotype calls supported by overrepresented sequences that are possibly derived from paralogous and repetitive sequences. No one is able to exactly distinguish which reads were wrongly mapped. The error correction algorithms basically "believe" that the genotype calls supported by the higher number of reads were more probable, while fewer reads give the algorithms more space to guess the true genotype. The large number of reads at the genotype calls with overrepresented sequences does not increase the reliability of the genotype calls but rather causes misestimation of homozygous calls as heterozygous. Although GSBcleanR tries to detect mismapped reads, we recommend to filter out genotype calls supported by overrepresented sequences to reduce their negative impact on estimation. Nevertheless, GBScleanR accepts the error prone markers in a given dataset and recognises the error pattern via the iterative parameter optimization. Therefore, no strict filtering of markers based on allele frequency and missing rate is required, which is normally performed for RRS-based genotyping data (6, 44).

DRAFT

These filtering steps remove a large portion of the detected SNP markers (45–48). In the case of our real data, filtering to remove possible error prone markers based on, for example, minor allele frequency > 0.2 and missing rate < 0.5 left only 2145 of 5035 SNPs for the 12 chromosomes in the real data. Markers that would be filtered out may possibly contain information about the true genotype at the markers even if the data are partial and skewed. GBScleanR is able to integrate those less informative markers into the error correction process by tweaking the HMM. Since there is no clear guideline for filtering strength, which should differ depending on the dataset used, filtering criteria are left in the hands of researchers. GBScleanR allows users to skip such an arbitrary step and improves the cost efficiency of the RRS-based genotyping system by reducing the number of unused reads obtained via costly NGS.

Limitations and perspectives of GBScleanR. The current implementation of GBScleanR treats all chromosomes as diploid autosomes. Although sex chromosomes can be processed by our algorithm, the calculation of transition probabilities between autosomes and sex chromosomes in which any recombination does not occur in either male or female individuals would be expected to differ. The algorithm therefore overestimates the number of recombinations in the sex chromosomes, while Supplemental Figure S3 shows that an overrated recombination frequency does not produce any severe effects on genotype estimation. A previously published paper (49) introduced a model that can be used to infer the number of recombinations in the sex chromosomes of experimental populations. We plan to further implement this model into GBScleanR in future updates.

In addition, there are obvious demands for applying GBScleanR to polyploid organisms. Several dosage estimation algorithms have already been implemented some tools (50–52). Nevertheless, all of these infer the dosage at each marker independently and do not incorporate information at adjacent markers, as performed in a HMM. The dosage estimation using a HMM may increase the robustness against error prone markers, while increasing the complexity of the algorithm at the same time. The codes of GBScleanR have been basically organized to allow us to expand its functions for polyploid organisms. Thus, GBScleanR will support polyploids in the future.

The recent remarkable development in third-generation sequencing, named Sequel by Pacific Bioscience and MinION by Oxford Nanopore Technologies, enables us to obtain high quality genome sequences at reasonable cost (53, 54). Reference genome information for any organism is expected to become available soon. Thus, RRS-based genotyping systems that are underpinned by a flexible robust error correction tool such as GBScleanR will become practical platforms for use in breeding programs that require a prompt and efficient scheme under limited periods and budgets. Distant relatives, especially the ancestral wild species of domesticated species, are important genetic resources, particularly in plant breeding, that can improve abiotic and biotic stress resistance (55–57). For such genetic studies and breeding projects using wild species, GBScleanR could be the first choice as a fundamental tool for genotype identification coupled with a RRS-based genotyping system.

DATA AVAILABILITY

The stable version of GBScleanR is available on Bioconductor (https://bioconductor.org/packages/GBScleanR/). The stable version and the latest developmental version are available on GitHub (https://github.com/tomoyukif/GBScleanR). This package and its source codes are distributed under the license GNU General Public License version 3. The vcf file for the F_2 population and R scripts for the data analyses performed in this study are also available on GitHub (https://github.com/tomoyukif/GBSR_SupFiles). Further description of the sequencing reads for the real data can be found in (42).

FUNDING

This work was supported by the CREST program of the Japan Science and Technology Agency [JPMJCR13B1 to .M.A]; the SATREPS program in collaboration of the Japan Science and Technology Agency and the Japan International Cooperation Agency [JPMJSA1706 to T.F. and M.A.]; KAKENHI of the Ministry of Education, Culture, Sports, Science and Technology of Japan [JP20H05912 to M.A.]; and KAKENHI of Japan Society for the Promotion of Science [JP20K15503 to T.F. and JP20H02958 to T.Y.].

Conflict of Interest

None declared.

Referenc

1. Goodwin,S., McPherson,J.D. and McCombie,W.R. (2016) Coming of age: Ten years of next-generation sequencing technologies. *Nat. Rev. Genet.*, **17**, 333–351.
2. Davey,J.W., Hohenlohe,P.A., Etter,P.D., Boone,J.Q., Catchen,J.M. and Blaxter,M.L. (2011) Genome-wide genetic marker discovery and genotyping using next-generation sequencing. *Nat. Rev. Genet.*, **12**, 499–510.
3. Torkamaneh,D., Boyle,B. and Belzile,F. (2018) Efficient genome-wide genotyping strategies and data integration in crop plants. *Theor. Appl. Genet.*, **131**, 499–511.
4. Unamba,C.I., Nag,A. and Sharma,R.K. (2015) Next generation sequencing technologies: The doorway to the unexplored genomics of non-model plants. *Front. Plant Sci.*, **6**, 1074.
5. Bhat,J.A., Ali,S., Salgotra,R.K., Mir,Z.A., Dutta,S., Jadon,V., Tyagi,A. Mushtaq,M., Jain,N., Singh,P.K., et al.. (2016) Genomic selection in the era of next generation sequencing for complex traits in plant breeding. *Front. genet.*, **7**, 221.
6. Scheben,A., Batley,J. and Edwards,D. (2017) Genotyping-by-sequencing approaches to characterize crop genomes: choosing the right tool for the right application. *Plant Biotechnol. J.*, **15**, 149–161.
7. Ali,O.A., O'Rourke,S.M., Amish,S.J., Meek,M.H., Luikart,G., Jeffres,C. and Miller,M.R. (2016) Rad capture (Rapture): Flexible and efficient sequence-based genotyping. *Genetics*, **202**, 389–400.
8. Baird,N.A., Etter,P.D., Atwood,T.S., Currey,M.C., Shiver,A.L., Lewis,Z.A., Selker,E.U., Cresko,W.A. and Johnson,E.A. (2008) Rapid SNP discovery and genetic mapping using sequenced RAD markers. *PLoS ONE*, **3**, e3376.
9. Fu,Z., Epstein,B., Kelley,J.L., Zheng,Q., Bergland,A.O., Castillo Carrillo,C.I., Jensen,A.S., Dahan,J., Karasev,A.V. and Snyder,W.E.(2017) Using NextRAD sequencing to infer movement of herbivores among host plants. *PLOS ONE*, **12**, e0177742.
10. Miller,M.R., Dunham,J.P., Amores,A., Cresko,W.A. and Johnson,E.A. (2007) Rapid and cost-effective polymorphism identification and genotyping using restriction site associated DNA (RAD) markers. *Genome Res.*, **17**, 240–248.
11. Peterson,B.K., Weber,J.N., Kay,E.H., Fisher,H.S. and Hoekstra,H.E. (2012) Double Digest RADseq: An inexpensive method for de novo SNP discovery and genotyping in model and non-model species. *PLoS ONE*, **7**, e37135.
12. Toonen,R.J., Puritz,J.B., Forsman,Z.H., Whitney,J.L., Fernandez-Silva,I., Andrews,K.R. and Bird,C.E. (2013) ezRAD: A simplified method for genomic genotyping in non-model organisms. *PeerJ*, **2013**, e203.

DRAFT

13. Wang,S., Meyer,E., Mckay,J.K. and Matz,M.V. (2012) 2b-RAD: A simple and flexible method for genome-wide genotyping. *Nat. Methods*, **9**, 808–810.
14. Elshire,R.J., Glaubitz,J.C., Sun,Q., Poland,J.A., Kawamoto,K., Buckler,E.S. and Mitchell,S.E. (2011) A robust, simple genotyping-by-sequencing (GBS) approach for high diversity species. *PLoS ONE*, **6**, e19379.
15. Poland,J.A., Brown,P.J., Sorrells,M.E. and Jannink,J. (2012) Development of high-density genetic maps for barley and wheat using a novel two-enzyme genotyping-by-sequencing approach. *PLoS ONE*, **7**, e32253.
16. De Donato,M., Peters,S.O., Mitchell,S.E., Hussain,T. and Imumorin,I.G. (2013) Genotyping-by-sequencing (GBS): A novel, efficient and cost-effective genotyping method for cattle using next-generation sequencing. *PLoS ONE*, **8**, e62137.
17. Kujur,A., Upadhyaya,H.D., Shree,T., Bajaj,D., Das,S., Saxena,M.S., Badoni,S., Kumar,V., Tripathi,S., Gowda,C.L., et al.. (2015) Ultra-high density intra-specific genetic linkage maps accelerate identification of functionally relevant molecular tags governing important agronomic traits in chickpea. *Sci. Rep.*, **5**.
18. Morris,G.P., Ramu,P., Deshpande,S.P., Hash,C.T., Shah,T., Upadhyaya,H.D., Riera-Lizarazu,O., Brown,P.J., Acharya,C.B., Mitchell,S.E., et al.. (2013) Population genomic and genome-wide association studies of agroclimatic traits in sorghum. *Proc. Natl. Acad. Sci. USA*, **110**, 453–458.
19. Parker,C.C., Gopalakrishnan,S., Carbonetto,P., Gonzales,N.M., Leung,E., Park,Y.J., Aryee,E., Davis,J., Blizard,D.A., Ackert-Bicknell,C.L., et al.. (2016) Genome-wide association study of behavioral, physiological and gene expression traits in outbred CFW mice. *Nat. Genet.*, **48**, 919–926.
20. Spindel,J., Begum,H., Akdemir,D., Virk,P., Collard,B., Redoña,E., Atlin,G., Jannink,J. and McCouch,S.R. (2015) Genomic selection and association mapping in rice (Oryza sativa): effect of trait genetic architecture, training population composition, marker number and statistical model on accuracy of rice genomic selection in elite, tropical rice breeding lines. *PLoS Genet.*, **11**, e1004982.
21. Litt,M. and Luty,J.A. (1989) A hypervariable microsatellite revealed by in vitro amplification of a dinucleotide repeat within the cardiac muscle actin gene. *Am. J. Hum. Genet.*, **44**, 397–401.
22. Syvanen,A.C. (2005) Toward genome-wide snp genotyping. *Nat. Genet.*, **37**, S5-S10.
23. Browning,S.R. and Browning,B.L. (2007) Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *Am. J. Hum. Genet.*, **81**, 1084–1097.
24. Huang,X., Feng,Q., Qian,Q., Zhao,Q., Wang,L., Wang,A., Guan,J., Fan,D., Weng,Q., Huang,T., et al.. (2009) High-throughput genotyping by whole-genome resequencing. *Genome Res.*, *19*, 1068–1076.
25. Fragoso,C.A., Heffelfinger,C., Zhao,H., Dellaporta,S.L. (2016) Imputing genotypes in biallelic populations from low-coverage sequence data. *Genetics*, **202**, 487–495.
26. Rowan,B.A., Patel,V., Weigel,D. and Schneeberger,K. (2015) Rapid and inexpensive whole-genome genotyping-by-sequencing for crossover localization and fine-scale genetic mapping. *G3*, **5**, 385–398.
27. Swarts,K., Li,H., Romero Navarro,J.A., An,D., Romay,M.C., Hearne,S., Acharya,C., Glaubitz,J.C., Mitchell,S., Elshire,R.J., et al.. (2014) Novel methods to optimize genotypic imputation for low-coverage, next-generation sequence data in crop plants. *Plant Genome*, **7**, plantgenome2014.05.0023.
28. Zheng,C., Boer,M.P. and van Eeuwijk,F.A. (2018) Accurate genotype imputation in multiparental populations from low-coverage sequence. *Genetics*, **210**, 71–82.
29. Eddy,S.R. (2004) What is a hidden Markov model? *Nat. Biotechnol.*, **22**, 1315-–1316
30. DaCosta,J.M. and Sorenson,M.D. (2014) Amplification biases and consistent recovery of loci in a double-digest RAD-seq protocol. *PLoS ONE*, **9**, e106713.
31. Davey,J.W., Cezard,T., Fuentes-Utrilla,P., Eland,C., Gharbi,K. and Blaxter,M.L. (2013) Special features of RAD Sequencing data: Implications for genotyping. *Mol. Ecol.*, **22**, 3151–3164.
32. Wijnker,E., James,G.V., Ding,J., Becker,F., Klasen,J.R., Rawat,V., Rowan,B.A., de Jong,D.F., de Snoo,C.B., Zapata,L., et al.. (2013) The genomic landscape of meiotic crossovers and gene conversions in Arabidopsis thaliana. *eLife*, **2013**.
33. McKenna,A., Hanna,M., Banks,E., Sivachenko,A., Cibulskis,K., Kernytsky,A., Garimella,K., Altshuler,D., Gabriel,S., Daly,M. and DePristo,M.A. (2010) The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.*, **20**, 1297–1303.
34. Glaubitz,J.C., Casstevens,T.M., Lu,F., Harriman,J., Elshire,R.J., Sun,Q. and Buckler,E.S. (2014) TASSEL-GBS: A high capacity genotyping by sequencing analysis pipeline. *PLoS ONE*, **9**, e90346.
35. Zheng,X., Levine,D., Shen,J., Gogarten,S.M., Laurie,C. and Weir,B.S. (2012) A high-performance computing toolset for relatedness and principal component analysis of SNP data. *Bioinformatics*, **28**, 3326–3328.
36. Li,H., and Barrett,J. (2011) A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, **27**, 2987–2993.
37. Gogarten,S.M., Bhangale,T., Conomos,M.P., Laurie,C.A., McHugh,C.P., Painter,I., Zheng,X., Crosslin,D.R., Levine,D., Lumley,T., et al.. (2012) GWASTools: an R/Bioconductor package for quality control and analysis of genome-wide association studies. *Bioinformatics*, **28**, 3329–3331.
38. Gogarten,S.M., Sofer,T., Chen,H., Yu,C., Brody,J.A., Thornton,T.A., Rice,K.M. and Conomos,M.P. (2019) Genetic association testing using the GENESIS R/Bioconductor package. *Bioinformatics*, **35**, 5346–5348.
39. Zheng,C., Boer,M.P. and Van Eeuwijk,F.A. (2014) A general modeling framework for genome ancestral origins in multiparental populations. *Genetics*, **198**, 87–101.
40. Zheng,C., Boer,M.P. and van Eeuwijk,F.A. (2015) Reconstruction of genome ancestry blocks in multiparental populations. *Genetics* **200**, 1073–1087.
41. Rabiner,L.R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**, 257–286.
42. Furuta,T., Ashikari,M., Jena,K.K., Doi,K. and Reuscher,S. (2017) Adapting genotyping-by-sequencing for rice F2 populations. *G3*, **7**, 881–893.
43. Chen,M., Presting,G., Barbazuk,W.B., Goicoechea,J.L., Blackmon,B., Fang,G., Kim,H., Frisch,D., Yu,Y., Sun,S., et al.. (2002) An Integrated Physical and Genetic Map of the Rice Genome. *Plant Cell*, **14**, 537–545.
44. Kim,C., Guo,H., Kong,W., Chandnani,R., Shuang,L.S. and Paterson,A.H. (2016) Application of genotyping by sequencing technology to a variety of crop breeding programs. *Plant Sci.*, **242**, 14–22.
45. Boyles,R.E., Pfeiffer,B.K., Cooper,E.A., Zielinski,K.J., Myers,M.T., Rooney,W.L. and Kresovich,S. (2017) Quantitative trait loci mapping of agronomic and yield traits in two grain sorghum biparental families. *Crop Sci.*, **57**, 2443–2456.
46. Celik,I., Bodur,S., Frary,A. and Doganlar,S. (2016) Genome-wide SNP discovery and genetic linkage map construction in sunflower (Helianthus annuus L.) using a genotyping by sequencing (GBS) approach. *Mol. Breed.*, **36**: 1–9.
47. Gardner,K.M. Brown,P., Cooke,T.F., Cann,S., Costa,F., Bustamante,C., Velasco,R., Troggio,M. and Myles,S. (2014) Fast and cost-effective genetic mapping in apple using next-generation sequencing. *G3*, **4**, 1681–1687.
48. Pootakham,W., Jomchai,N., Ruang-areerate,P., Shearman,J.R., Sonthirod,C., Sangsrakru,D., Tragoonrung,S. and Tangphatsornruang.S. (2015) Genome-wide SNP discovery and identification of QTL associated with agronomic traits in oil palm using genotyping-by-sequencing (GBS). *Genomics*, **105**, 288–295.
49. Zheng,C. (2015) Modeling X-linked ancestral origins in multiparental populations. *G3*, **5**, 777–801.
50. Clark,L.V., Lipka,A.E. and Sacks,E.J. (2019) polyRAD: Genotype calling with uncertainty from sequencing data in polyploids and diploids. *G3*, **9**, 663–673.
51. Gerard,D., Ferrão,L.F.V., Garcia,A.A.F. and Stephens,M. (2018) Genotyping polyploids from messy sequencing data. *Genetics*, **210**, 789–807.
52. Zych,K., Gort,G., Maliepaard,C.A., Jansen,R.C. and Voorrips,R.E. (2019) FitTetra 2.0 - Improved genotype calling for tetraploids with multiple population and parental data support. *BMC Bioinformatics*, **20**, 148.
53. Lu,H., Giordano,F. and Ning,Z. (2016) Oxford Nanopore MinION sequencing and genome assembly. *Genomics, Proteomics and Bioinformatics*, **14**, 265–279.
54. Rhoads,A. and Au,K.F. (2015) PacBio sequencing and its applications. *Genomics, Proteomics and Bioinformatics*, **13**, 278–289.
55. Atwell,B.J., Wang,H. and Scafaro,A.P. (2014) Could abiotic stress tolerance in wild relatives of rice be used to improve Oryza sativa? *Plant Sci.*, **215-216**, 48–58.
56. Brozynska,M., Furtado,A. and Henry,R.J. (2016) Genomics of crop wild relatives: Expanding the gene pool for crop improvement. *Plant Biotechnol. J.*, **14**, 1070–1085.
57. Kole,C., Muthamilarasan,M., Henry,R., Edwards,D., Sharma,R., Abberton,M., Batley,J., Bentley,A., Blakeney,M., Bryant,J., et al.. (2015) Application of genomics-assisted breeding for generation of climate resilient crops: Progress and prospects. *Front. Plant Sci.*, **6**, 563.

DRAFT

Table 1 Probability $P(x'^{o}_{mi}|x^{o}_{mi}, e^{map}_{m} = (e^{ref}, e^{alt}))$

		Observable genotype x'^{o}_{mi}		
		0	1	2
True genotype x^{o}_{mi}	0	$(1-e^{ref})$	e^{ref}	0
	1	0	1	0
	2	0	e^{alt}	$(1-e^{alt})$

Table 2 Parameters used in creating simulation datasets

Parameter	Choices
Number of offspring	10, 100, and 1000
Number of markers	50, 100, and 1000
Founder read depth*	$1\times$ and $3\times$
Offspring read depth*	$0.1\times$, $0.25\times$, $0.5\times$, $0.75\times$, $1\times$, $2\times$, and $3\times$

*Founder read depth and offspring read depth indicate the expected read depth per marker in the founder and offspring samples, respectively.

Table 3 Genotype-dependent read observation probability

		Observable allele read z	
		Ref	Alt
Simulated genotype x_{mi}	0	$(1-e^{seq})w_m$	$e^{seq}w_m$
	1	$0.5w_m$	$0.5(1-w_m)$
	2	$e^{seq}(1-w_m)$	$(1-e^{seq})(1-w_m)$

DRAFT

Table 4 Running time

Population	Size	GBSR_noOptim	GBSR	Magic	LB
homoP2_F2	10	1, 1, 1	1, 1, 3	1, 1, 3	2, 4, 33
homoP2_F2	100	1, 1, 2	1, 1, 5	1, 2, 20	17, 32, 305
homoP2_F2	1000	1, 2, 10	2, 4, 27	9, 16, 143	142, 300, 2854
hetP2_F2	10	1, 1, 2	1, 1, 5	1, 1, 6	NA
hetP2_F2	100	1, 1, 5	1, 2, 18	2, 4, 33	NA
hetP2_F2	1000	2, 5, 47	7, 17, 139	14, 25, 243	NA
homoP8_RIL	10	1, 2, 6	1, 4, 19	1, 2, 26	NA
homoP8_RIL	100	2, 3, 31	5, 8, 105	3, 6, 89	NA
homoP8_RIL	1000	12, 23, 260	41, 79, 927	29, 59, 616	NA

Running times in seconds for datasets with 50, 100, and 1000 markers are shown together as comma-separated values. Values less than 1 are shown as 1.

Figures

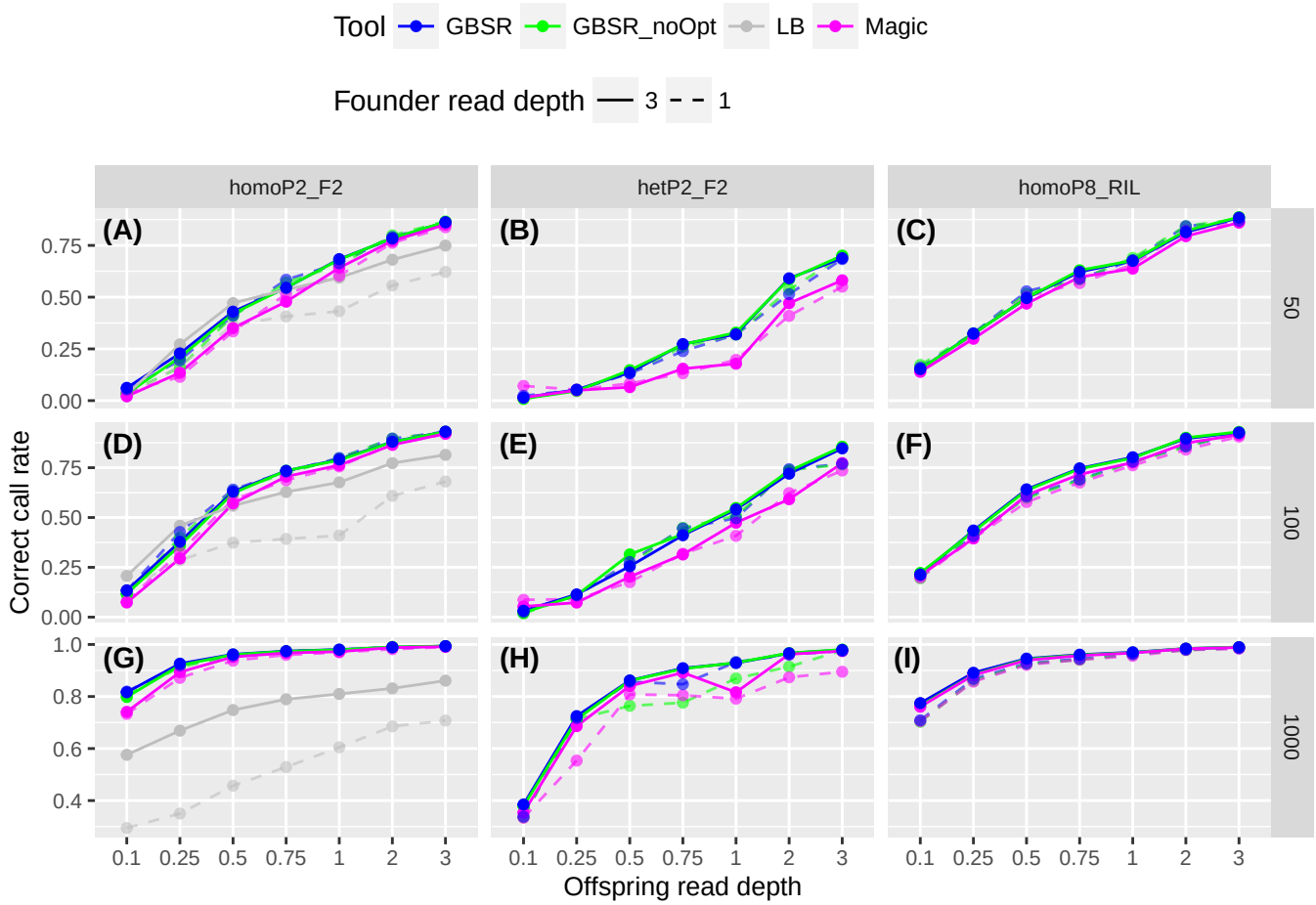


Fig. 1. Evaluation of offspring genotype estimation accuracy for the noError simulation data with 100 offspring. No allele read bias was assumed in the simulated data. The plots show correct call rates for the datasets with given read depths for offspring (x axis) and 50 (A-C), 100 (D-F), and 1000 (G-I) markers in the homoP2_F2 (A, D, and G), hetP2_F2 (B, E, and H), and homoP8_RIL (C, F, and I) scenarios. Solid lines and dashed lines indicate the results for the datasets with read depths at $3\times$ and $1\times$ for founders, respectively. "GBSR" and "GBSR_noOpt" represent GBScleanR with and without iterative optimization for w_m and e_m^{map}. "LB" and "Magic" indicate LB-Impute and magicImpute, respectively.

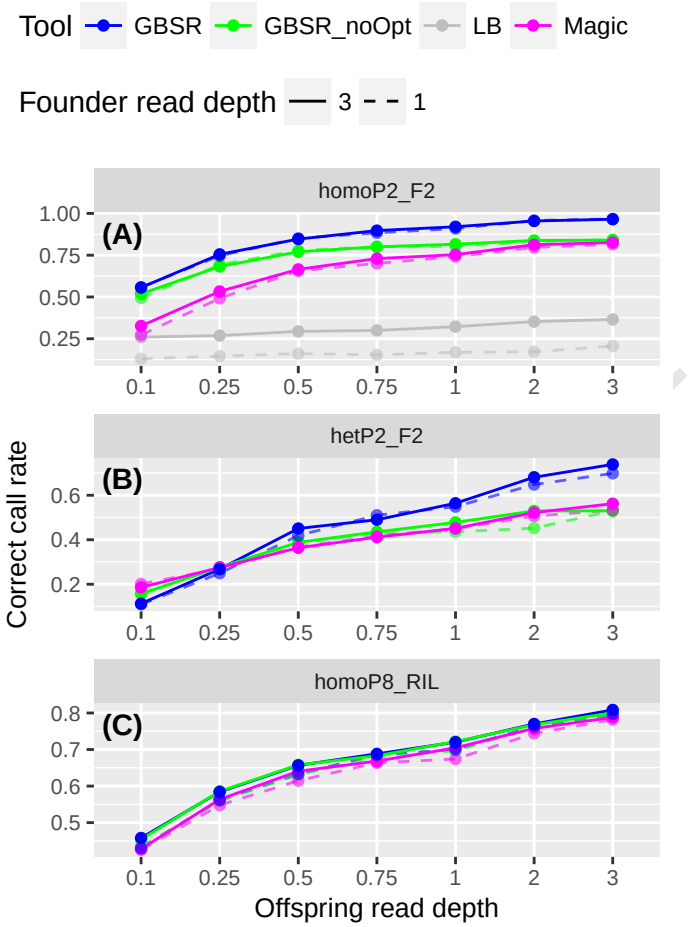


Fig. 2. Evaluation of offspring genotype estimation accuracy for the realError simulation data with 100 offspring and 620 markers. The plots show correct call rates for the datasets with given read depths for offspring (x axis) in the homoP2_F2 (A), hetP2_F2 (B), and homoP8_RIL (C) scenarios. Solid lines and dashed lines indicate the results for the datasets with read depths at $3\times$ and $1\times$ for founders, respectively. "GBSR" and "GBSR_noOpt" represent GBScleanR with and without iterative optimization for w_m and e_m^{map}. "LB" and "Magic" indicate LB-Impute and magicImpute, respectively.

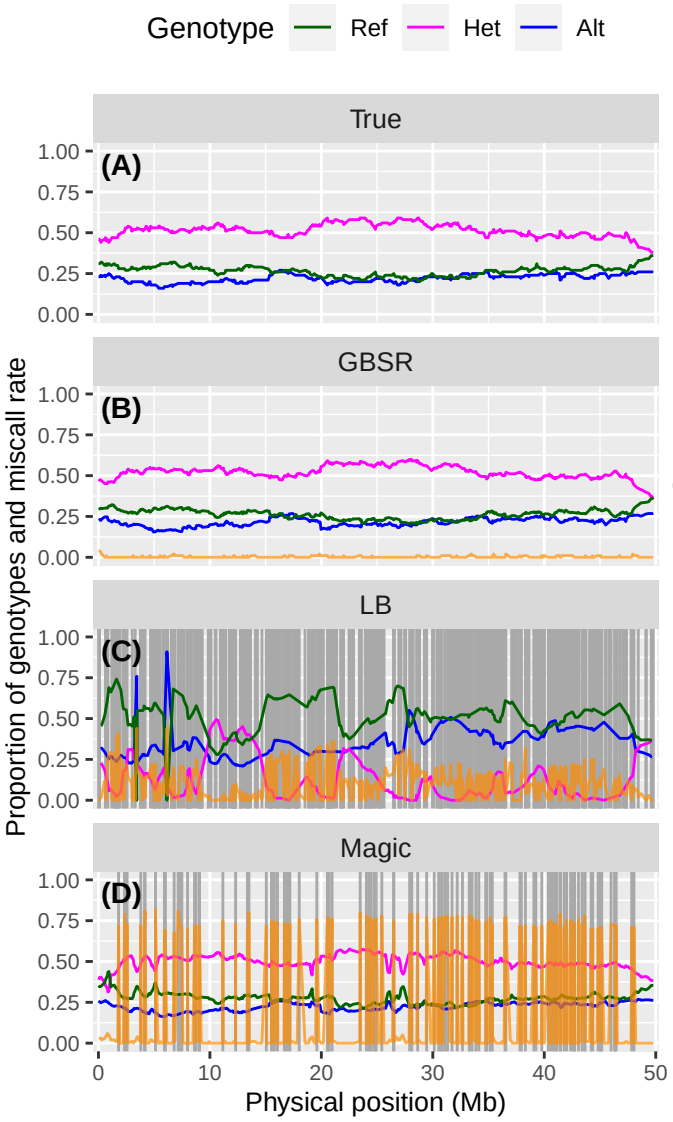


Fig. 3. Genotype ratios at markers ordered along a simulated chromosome. The plots show the genotype ratios calculated in the genotype estimation results for the realError dataset with 100 offspring and 620 markers with read depths of $3\times$ for offspring and founders in the homoP2_F2 scenario. "True" (A) indicates true genotype, while "GBSR" (B), "LB" (C), and "Magic" (D) indicates estimated genotype by GBScleanR, LB-Impute, and magicImpute, respectively. Proportions of reference homozygous, heterozygous, and alternative homozygous genotypes are represented by green, magenta, and blue lines, respectively. Orange lines indicate miscall rates at markers, while gray vertical lines indicate markers at which the algorithms wrongly assigned the same genotype for all founders.

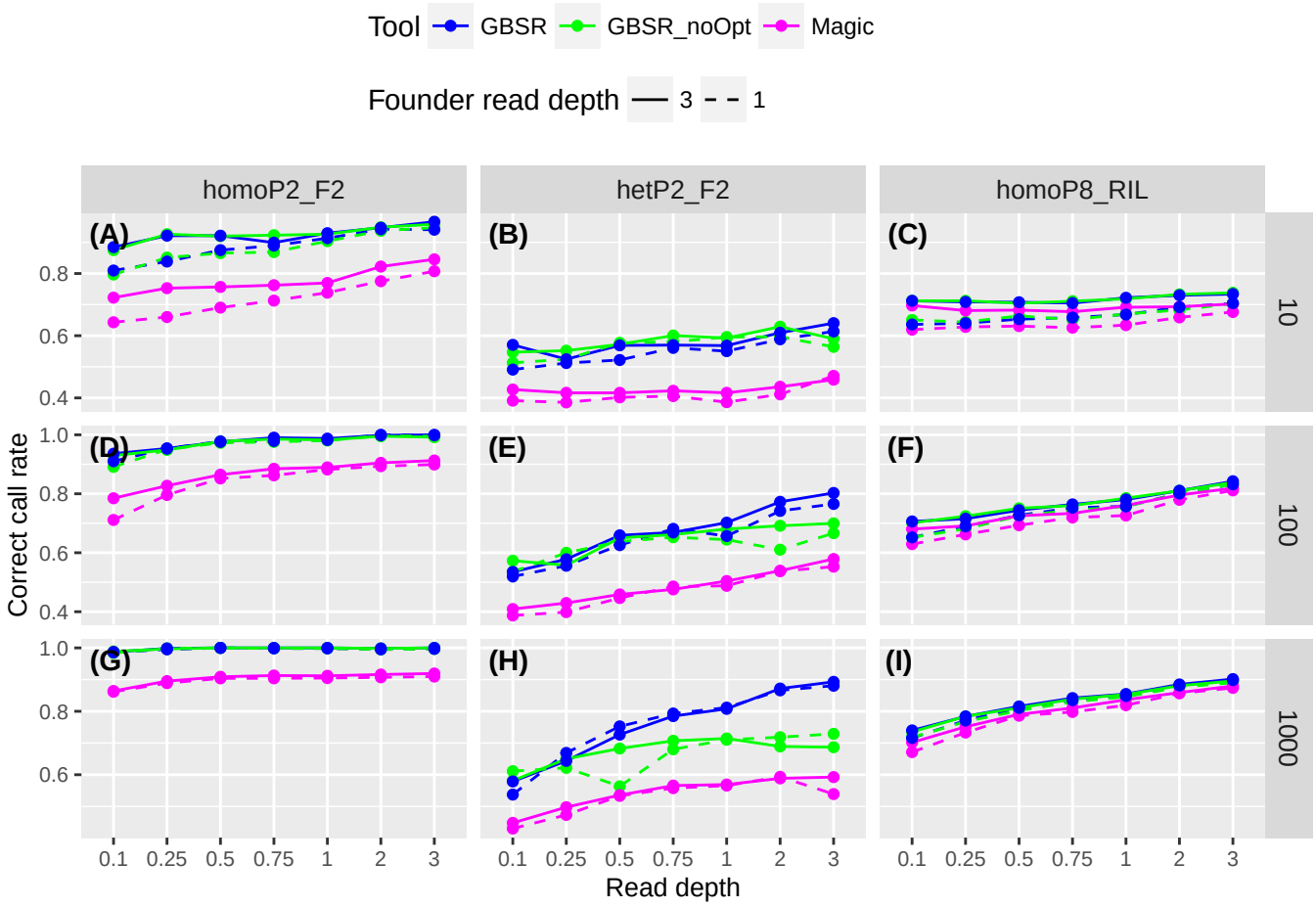


Fig. 4. Evaluation of founder genotype estimation accuracy for the realError simulation data. Plots show correct call rates in the datasets with 620 markers with given read depths for offspring and 10 (A-C), 100 (D-F), and 1000 (G-I) offspring in the homoP2_F2 (A, D, and G), hetP2_F2 (B, E, and H), and homoP8_RIL (C, F, and I) scenarios. Solid lines and dashed lines indicate the results for datasets with read depths at $3\times$ and $1\times$ for founders, respectively. "GBSR" represents GBScleanR with iterative optimization for w_m and e_m^{map}, while "Magic" denotes magicImpute.

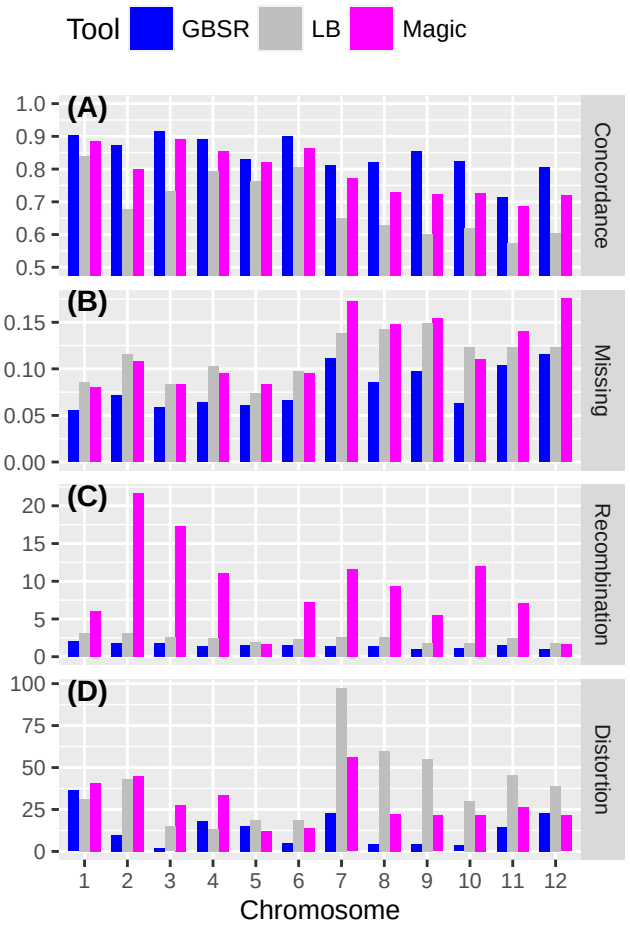


Fig. 5. (A) Concordance rate of true genotype calls and estimated genotype calls as the masked genotype calls are shown for each chromosome. The averages of missing rate (B), the number of recombinations (C), and segregation distortion level (D) are also visualized as bar plots for each chromosome. The segregation distortion level at each marker was obtained by dividing the chi-square value, which was calculated based on the assumption that the genotype ratio to be 0.25, 0.5, and 0.25 for three possible genotype calls, by the number of non missing genotype calls. Blue, gray, and magenta bars indicate the values of GBScleanR (GBSR), LB-Impute (LB), and magicImpute (Magic), respectively.

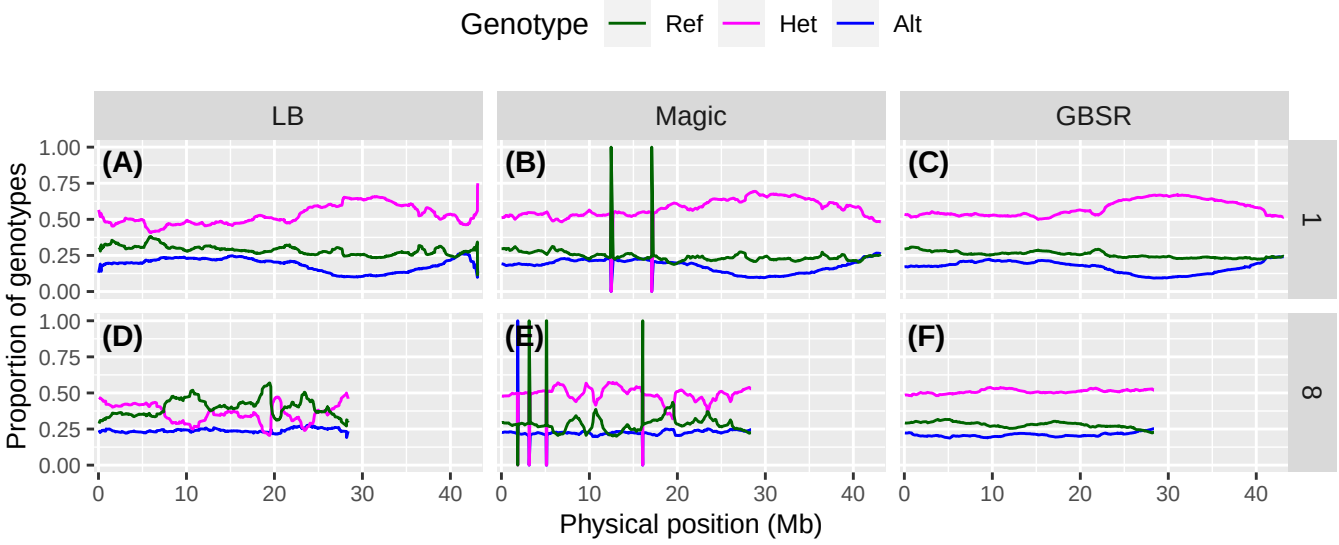


Fig. 6. Genotype ratios at markers along chromosome 1 (A-C) and 8 (D-F) in the real data are shown as line plots. "LB", "Magic", and "GBSR" indicate LB-Impute, magicImpute, and GBScleanR, respectively. Proportions of the reference homozygous, heterozygous, and alternative homozygous genotypes are represented by green, magenta, and blue lines, respectively.

Supplemental Figures for

GBScleanR: Robust genotyping error correction using hidden Markov model with error pattern recognition.

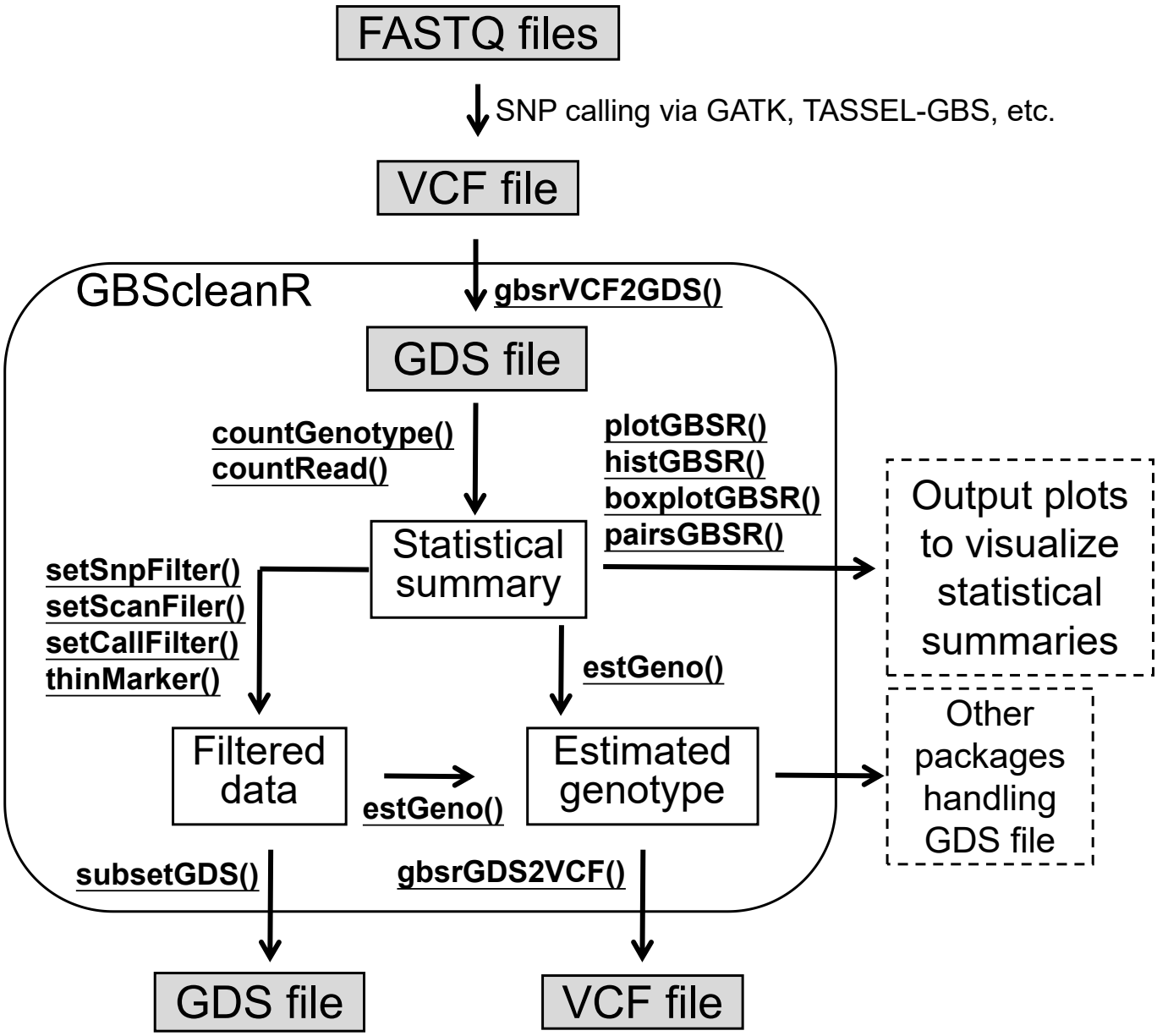
Tomoyuki Furuta[1*], Toshio Yamamoto[1], and Motoyuki Ashikari[2]

[1]Institute of Plant Sci. and Resources, Okayama University, Chu-oh 2-20-1, Kurashiki, Okayama, 710-0046, Japan
[2]Bioscience and biotechnology center, Nagoya University, Furo-cho, Chikusa, Nagoya, Aichi, 464-8601, Japan

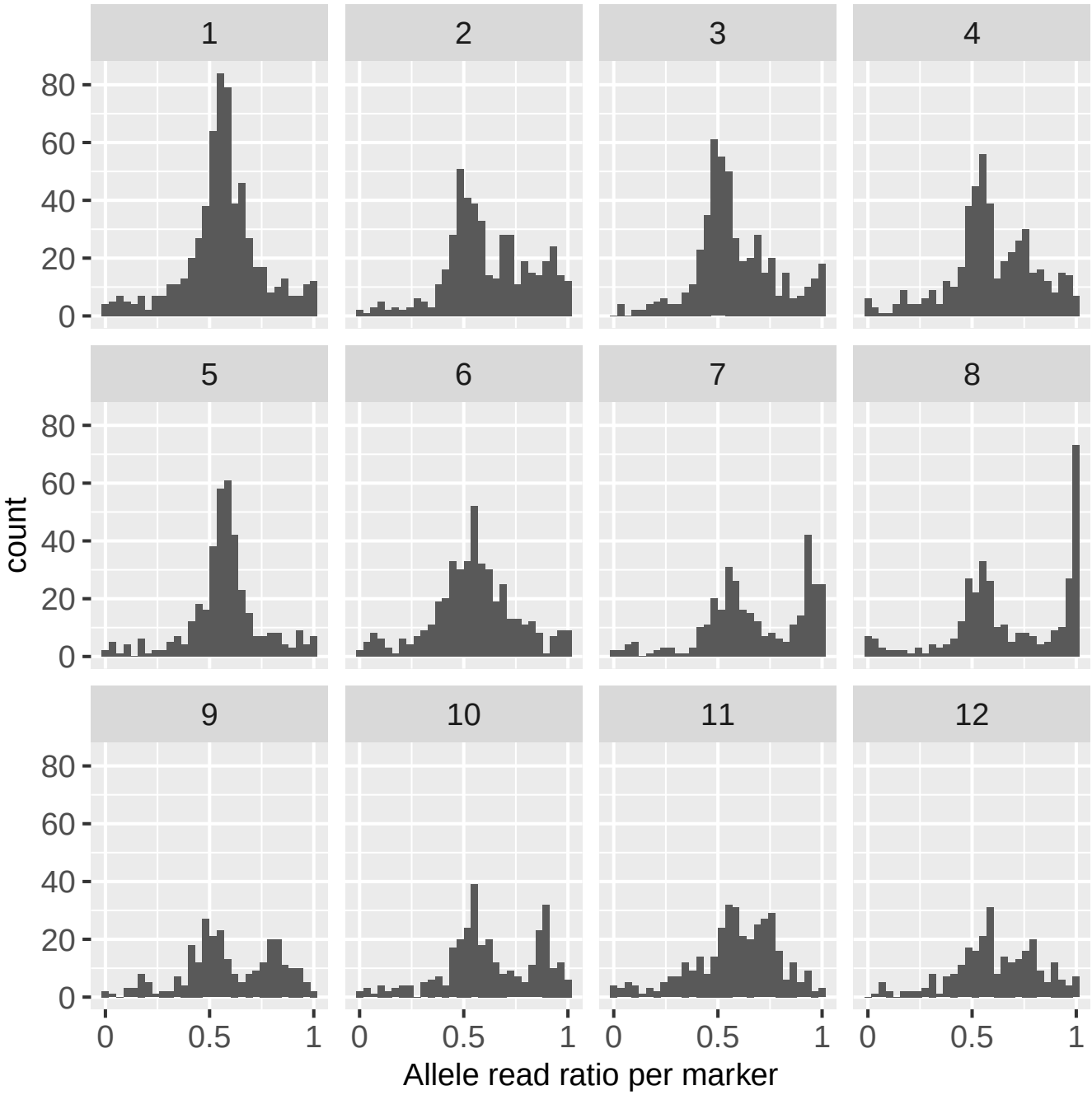
*To whom correspondence should be addressed.
Tel: +81 086 4341208; Fax: +81 086 434; Email: f.tomoyuki@okayama-u.ac.jp

DRAFT

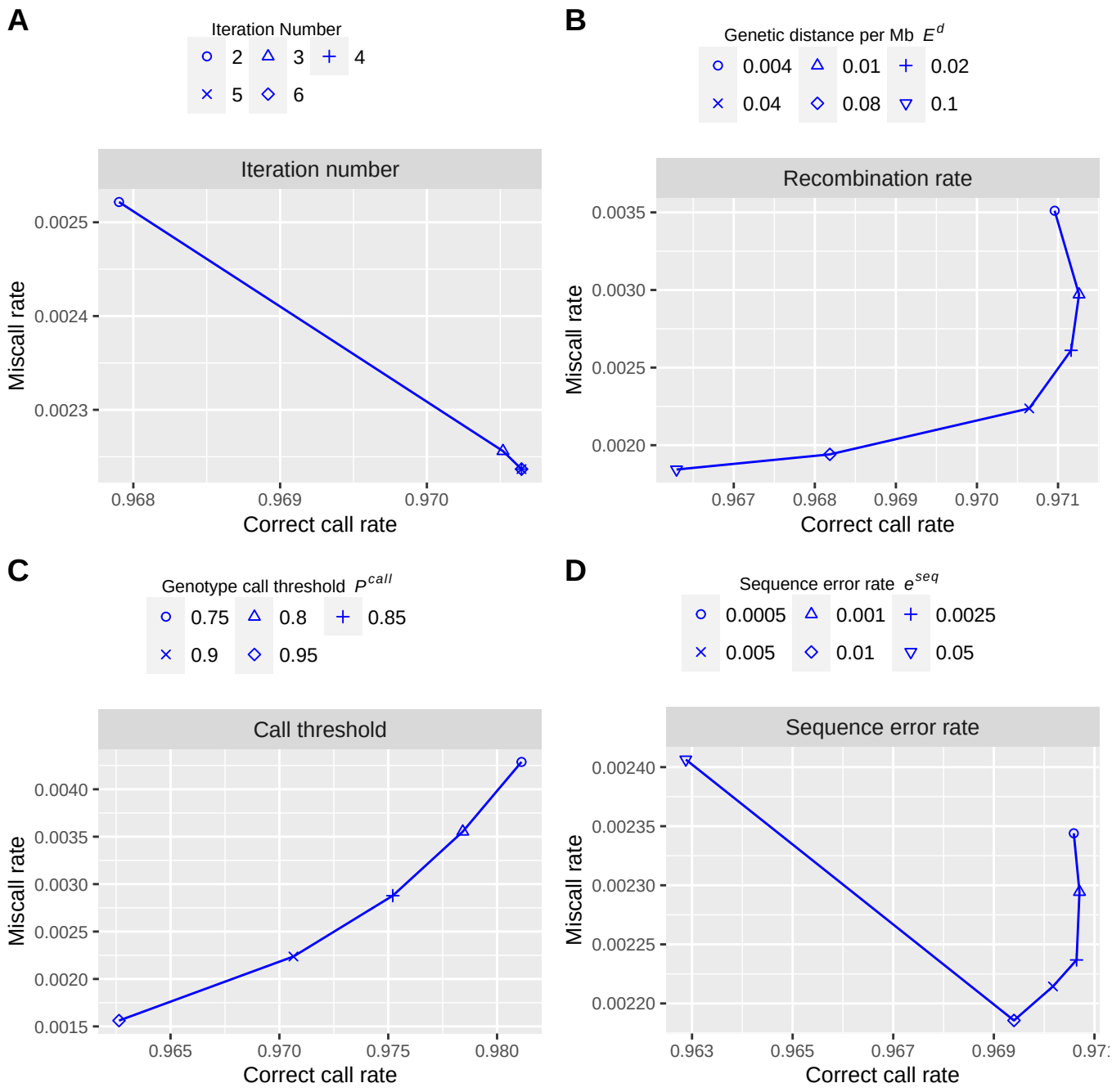
Supplemental Figures S1-S14



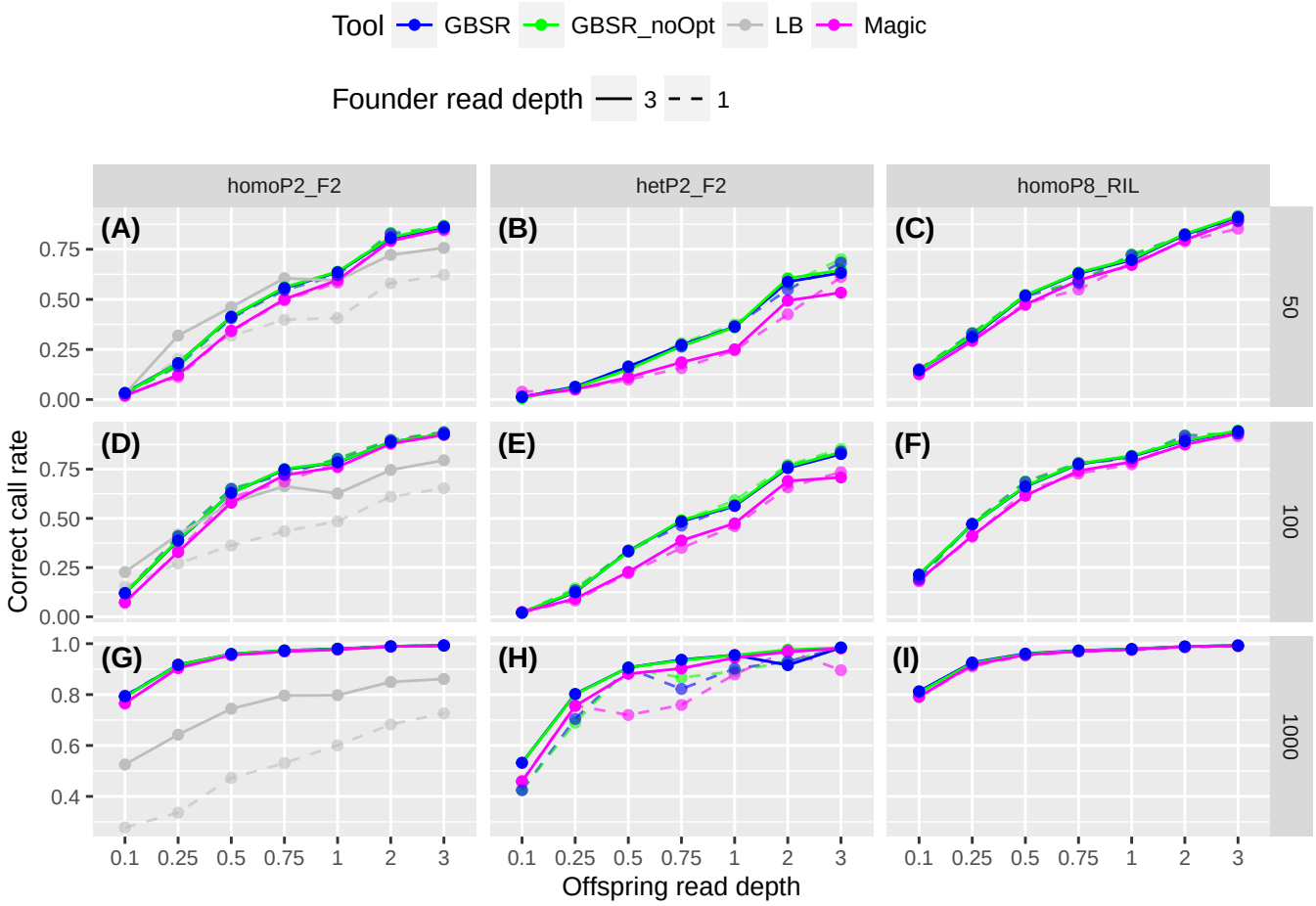
Supplemental Figure S1. Schematic image of workflow with the GBScleanR package. Gray boxes represent data files, while white boxes are data created by GBScleanR in the R environment. Under-lined text indicates the function names of GBScleanR. GBScleanR takes a VCF file generated via SNP callers such as GATK and TASSEL-GBS as inputs. The input VCF file first has to be converted to a GDS file to handle large amounts of genotype data on the R environment with less RAM usage. Statistical summaries of genotype and read counts can be obtained via the countGenotype() and countRead() functions. The summary data can be then visualised using several plotting functions. Users can also filter out samples and markers based on the statistical summaries using the setSnpFilter() and setScanFilter() functions, e.g., missing rate, heterozygosity, and minor allele frequencies for markers and samples. The function setCallFilter() allow users to filter out single genotype calls at markers based on read counts, while thinMarker() maintains one of the markers within a specified stretch in order to reduce the number of markers with redundant information. The estGeno() function is the core function of GBScleanR that estimates genotype based on given read count data. The filtered genotype data and the estimated genotype data can be output as a new GDS file and a new VCF file by the subsetGDS() and gbsrGDS2VCF() functions, respectively.



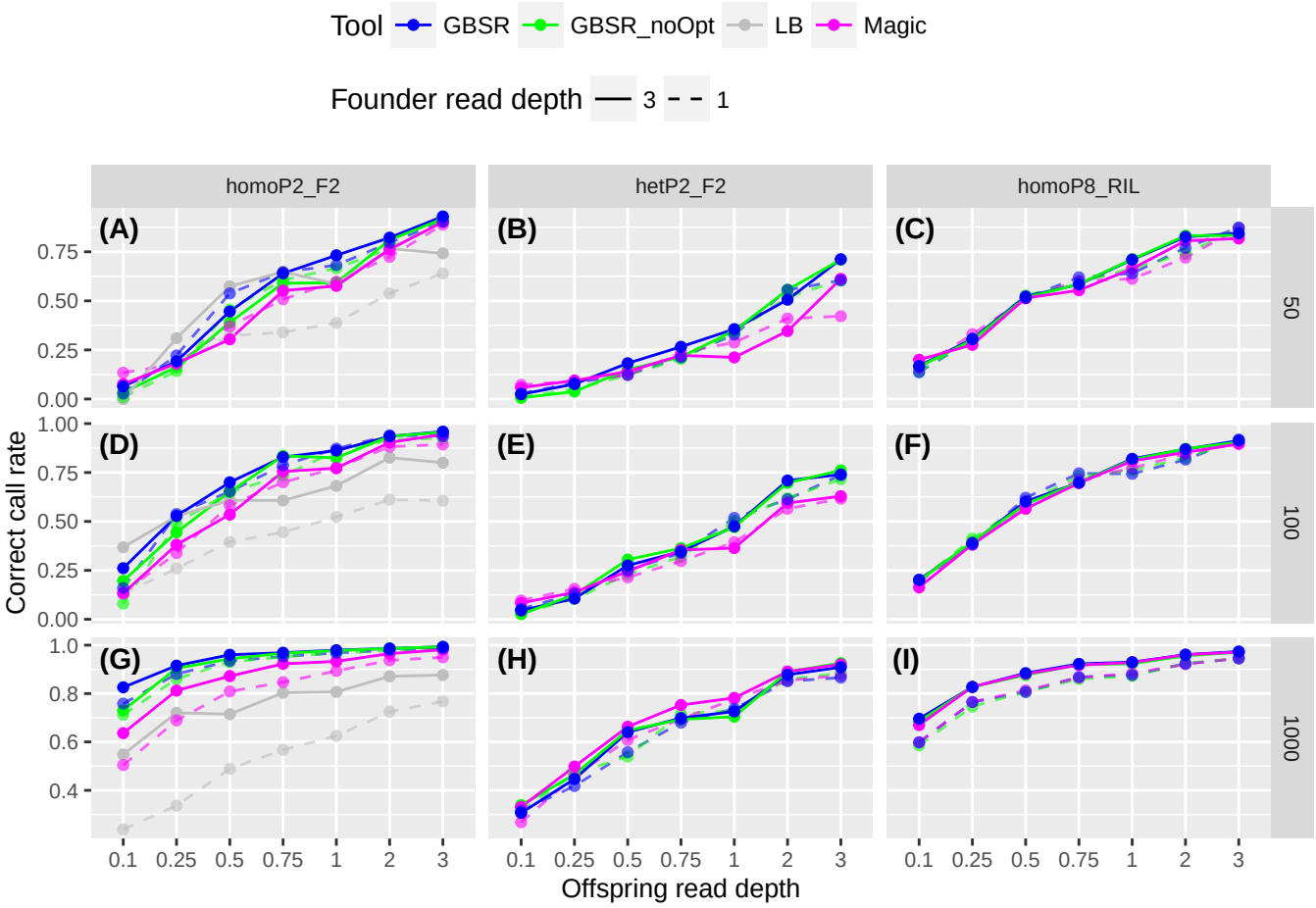
Supplemental Figure S2. Allele read ratio per marker observed in the real data. The histograms of the allele read ratios calculated for each marker are plotted separately for chromosomes 1-12. If the allele read ratio is greater than 0.5, reference allele reads are preferably observed at the marker. These histograms indicates that many markers tend to have reference reads. The majority of the markers counted in the right-most bin of each histogram were associated with high missing rates, which may lead to apparent allele read biases. The markers in the real data seem to prefer reference allele reads in all chromosomes.



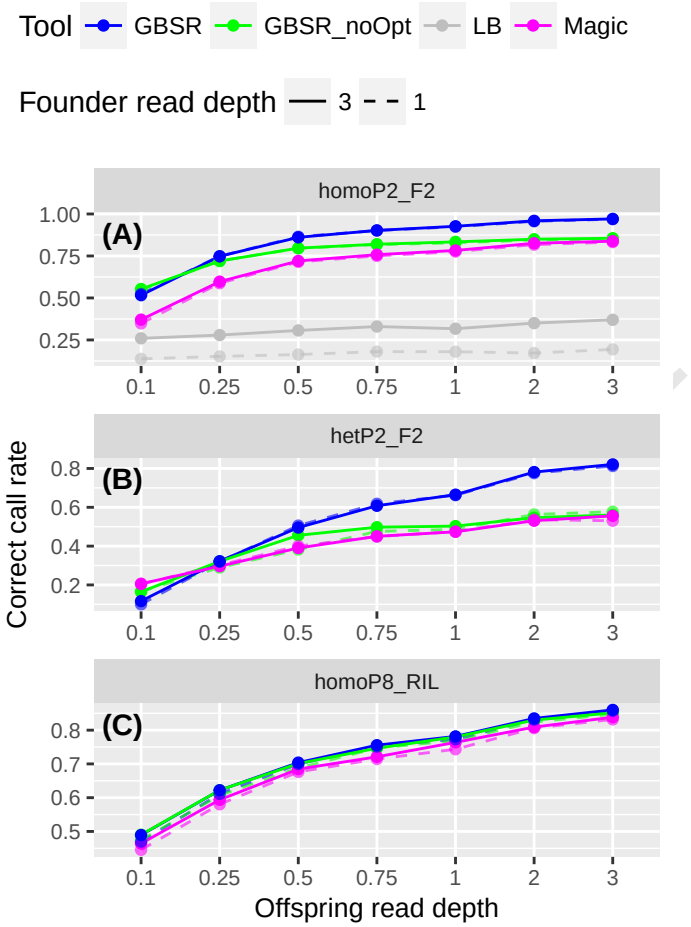
Supplemental Figure S3. Effects of parameter setting on genotype estimation accuracy. Correct call rates and miscall rates at various parameter settings are plotted for iteration number, P_{call} threshold, expected genetic distance per mega base pairs E^d, and sequence error rate e^{seq} from the top to bottom. Different symbols indicate different settings for each parameter.



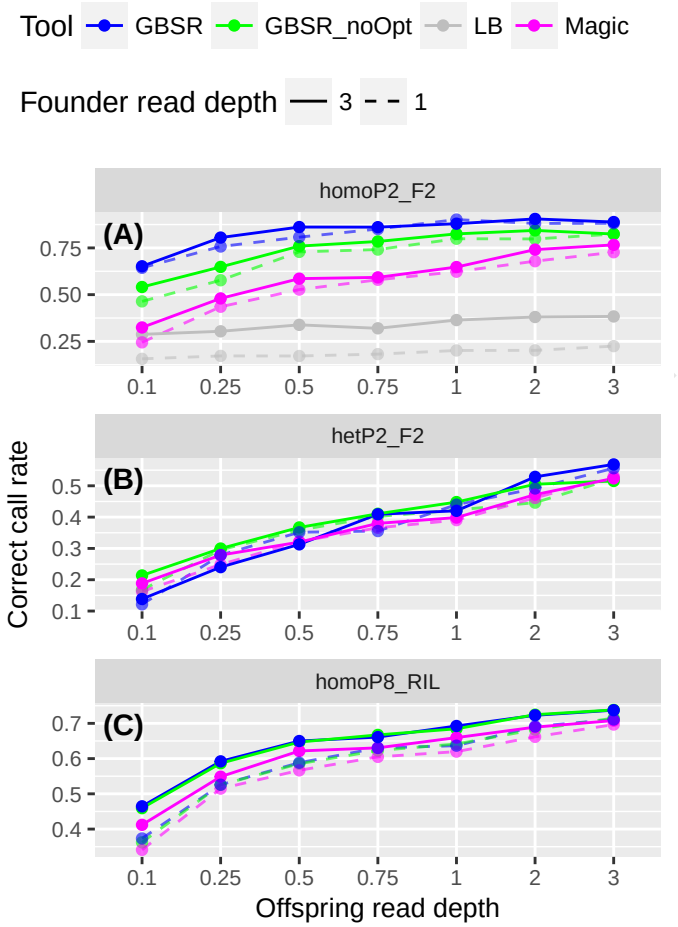
Supplemental Figure S4. Evaluation of offspring genotype estimation accuracy on the noError simulation data with 1000 offspring. No allele read bias was assumed in the simulated data. The plots show correct call rates for datasets with given read depths for offspring (x axis) and 50 (A-C), 100 (D-F), and 1000 (G-I) markers in the homoP2_F2 (A, D, and G), hetP2_F2 (B, E, and H), and homoP8_RIL (C, F, and I) scenarios. Solid lines and dashed lines indicate the results for the datasets with read depths at $3\times$ and $1\times$ for founders, respectively. "GBSR" and "GBSR_noOpt" represent GBScleanR with and without iterative optimization for w_m and e_m^{map}. "LB" and "Magic" indicate LB-Impute and magicImpute, respectively.



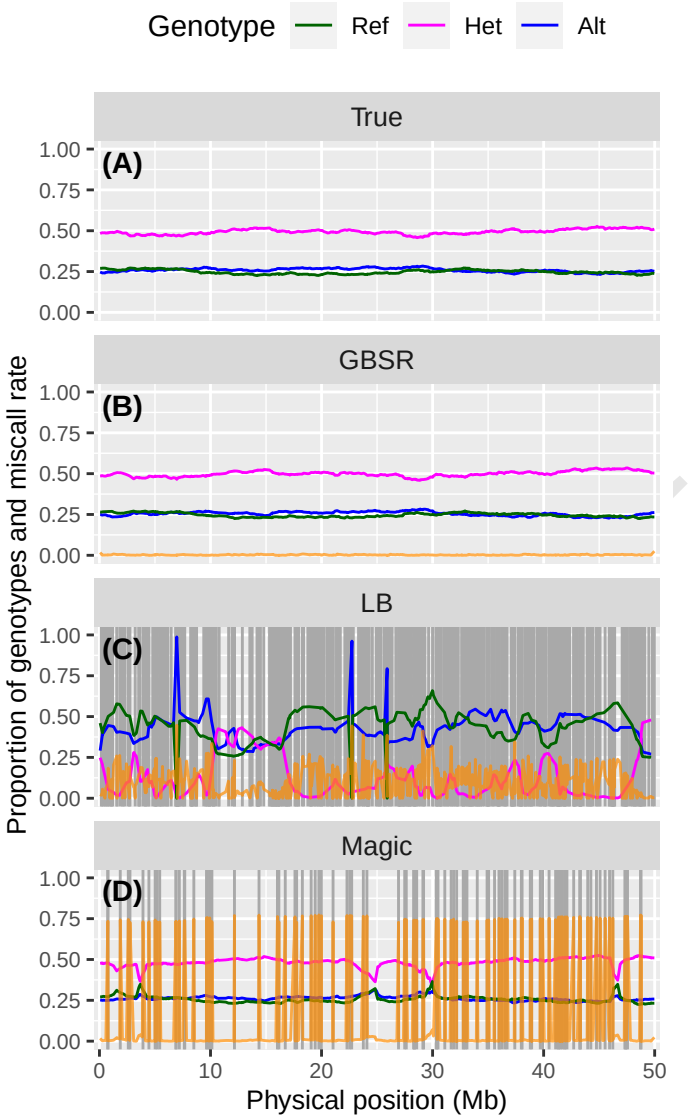
Supplemental Figure S5. Evaluation of offspring genotype estimation accuracy on the noError simulation data with 10 offspring. No allele read bias were assumed in the simulated data. The plots show correct call rates for datasets with given read depths for offspring (x axis) and 50 (A-C), 100 (D-F), and 1000 (G-I) markers in the homoP2_F2 (A, D, and G), hetP2_F2 (B, E, and H), and homoP8_RIL (C, F, and I) scenarios. Solid lines and dashed lines indicate results for datasets with read depths of $3\times$ and $1\times$ for founders, respectively. "GBSR" and "GBSR_noOpt" represent GBScleanR with and without iterative optimization for w_m and e_m^{map}. "LB" and "Magic" indicate LB-Impute and magicImpute, respectively.



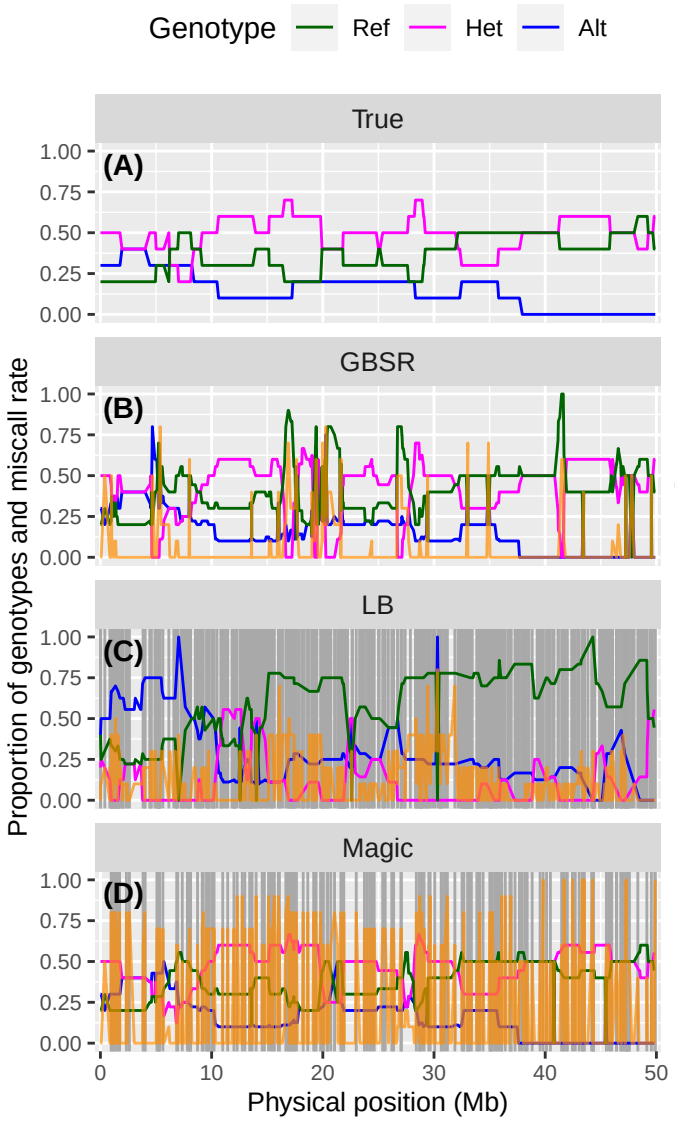
Supplemental Figure S6. Evaluation of offspring genotype estimation accuracy on the realError simulation data with 1000 offspring and 620 markers. Plots show correct call rates for the datasets with given read depths for offspring (x axis) in the homoP2_F2 (A), hetP2_F2 (B), and homoP8_RIL (C) scenarios. Solid lines and dashed lines indicate the results for the datasets with read depths of $3\times$ and $1\times$ for founders, respectively. "GBSR" and "GBSR_noOpt" represent GBScleanR with and without iterative optimization for w_m and e_m^{map}. "LB" and "Magic" indicate LB-Impute and magicImpute, respectively.



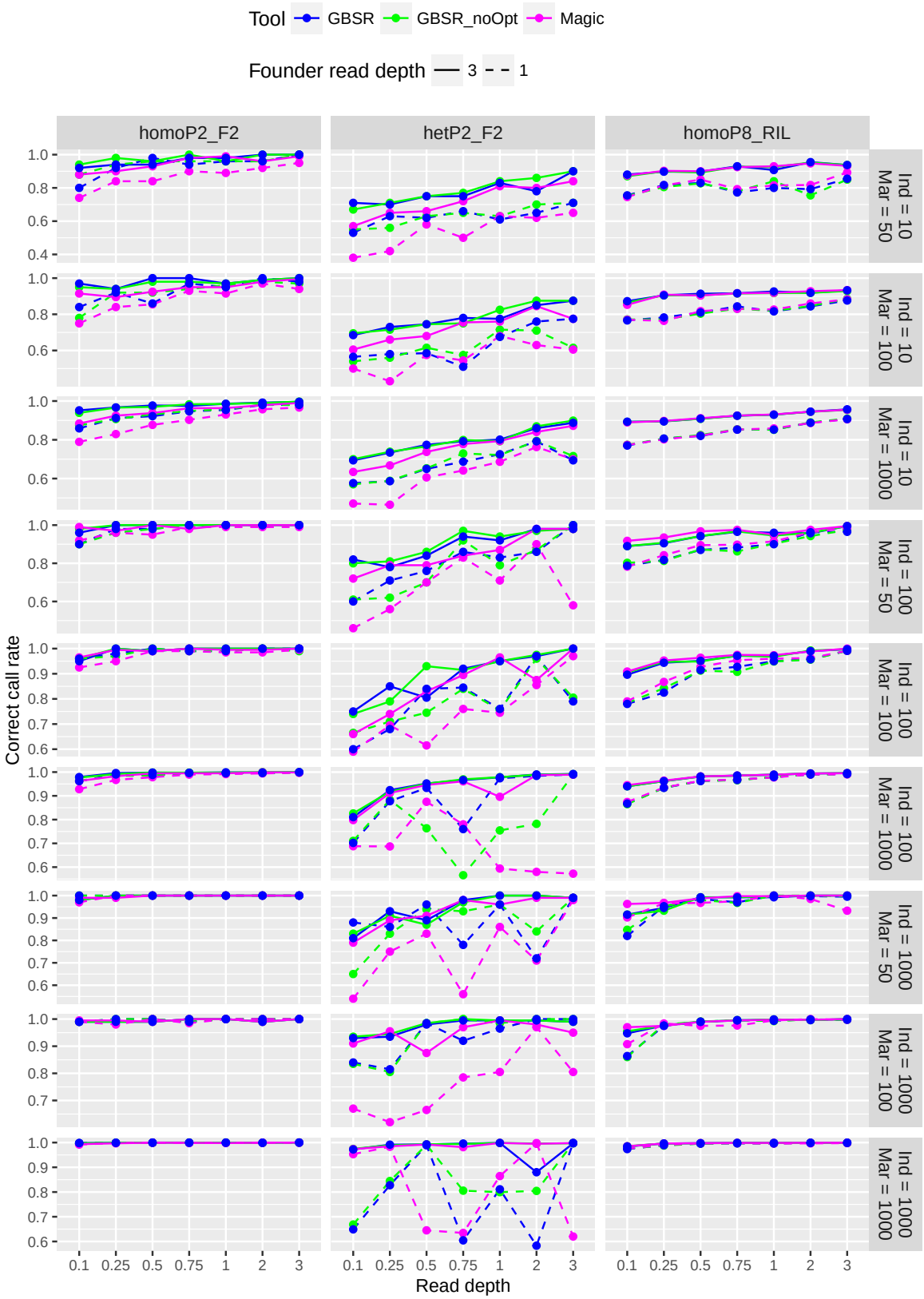
Supplemental Figure S7. Evaluation of offspring genotype estimation accuracy for realError simulation data with 10 offspring and 620 markers. Plots show correct call rates for the datasets with given read depths for offspring (x axis) in the homoP2_F2 (A), hetP2_F2 (B), and homoP8_RIL (C) scenarios. Solid lines and dashed lines indicate the results for the datasets with read depths of $3\times$ and $1\times$ for founders, respectively. "GBSR" and "GBSR_noOpt" represent GBScleanR with and without iterative optimization for w_m and e_m^{map}. "LB" and "Magic" indicate LB-Impute and magicImpute, respectively.



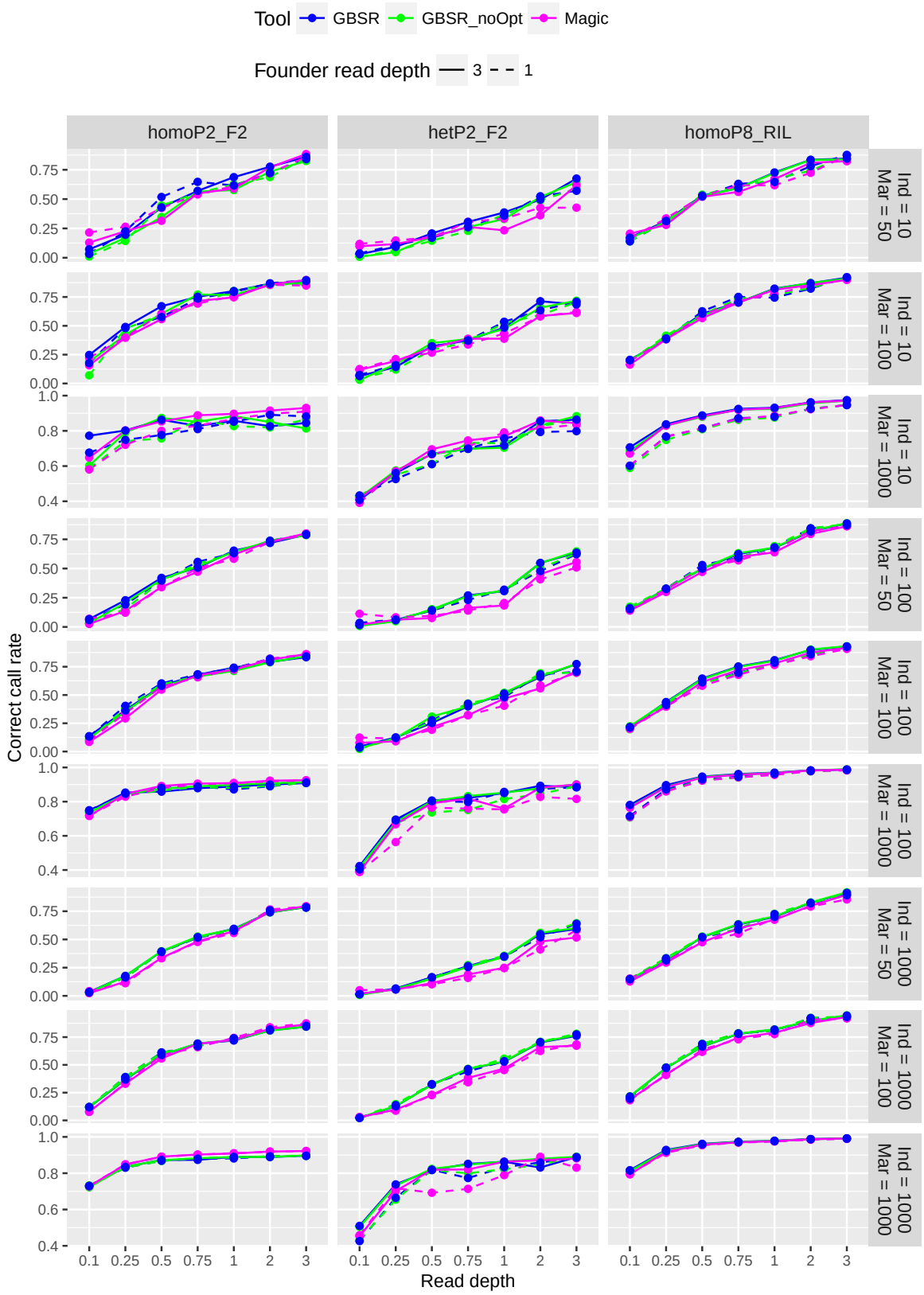
Supplemental Figure S8. Genotype ratios at markers ordered along a simulated chromosome. The plots show genotype ratios calculated in the genotype estimation results for the realError dataset with 1000 offspring and 620 markers with read depths of $3\times$ for offspring and founders in the homoP2_F2 scenario. "True" (A) indicates true genotype, while "GBSR" (B), "LB" (C), and "Magic" (D) indicates estimated genotype by GBScleanR, LB-Impute, and magicImpute, respectively. Proportions of reference homozygous, heterozygous, and alternative homozygous genotypes are represented by green, magenta, and blue lines, respectively. Orange lines indicate miscall rates at markers, while gray vertical lines indicate the markers at which the algorithms wrongly assigned the same genotype for all founders.



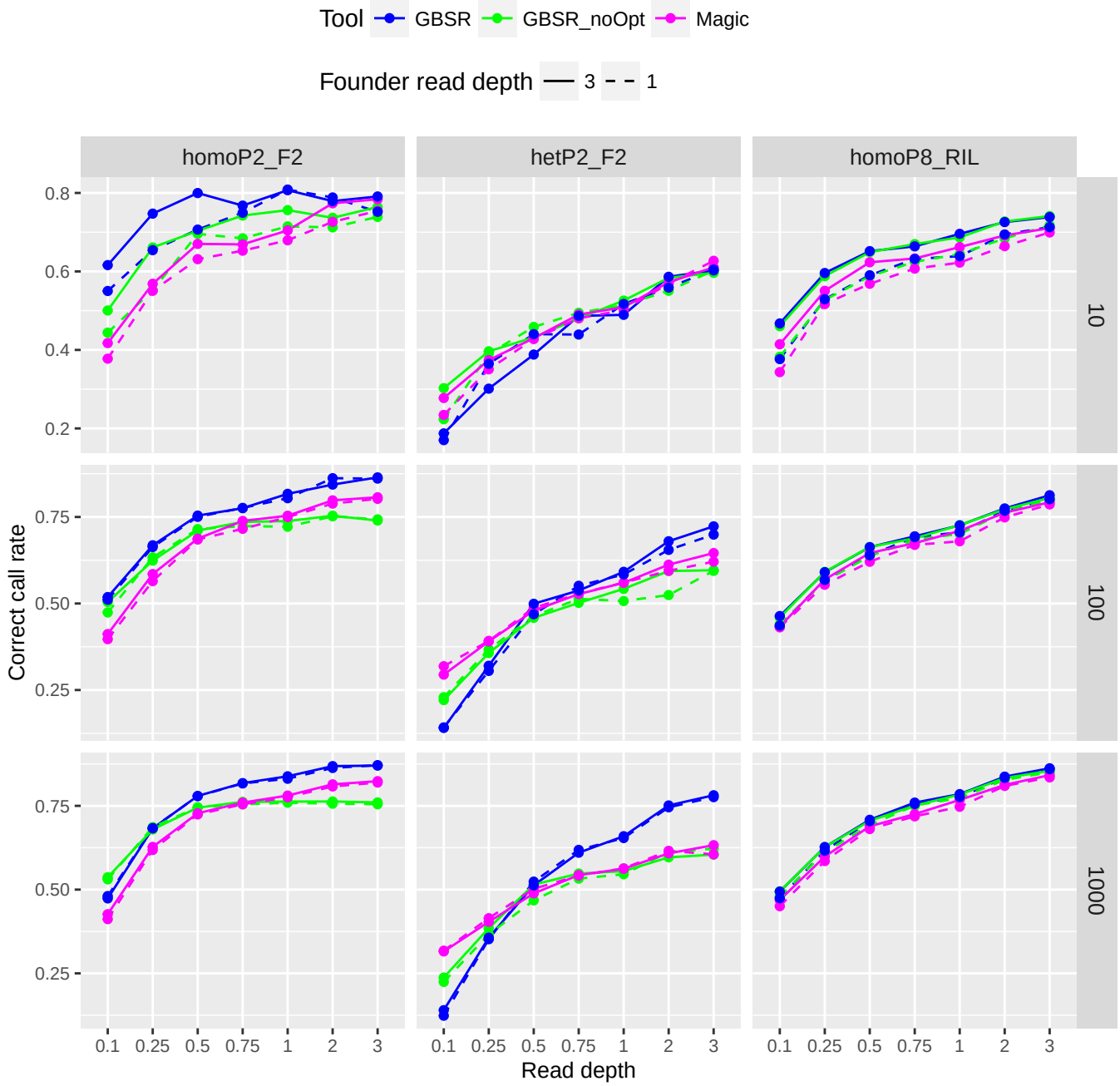
Supplemental Figure S9. Genotype ratios at markers ordered along a simulated chromosome. Plots show genotype ratios calculated in the genotype estimation results for the realError dataset with 10 offspring and 620 markers with read depths of $3\times$ for offspring and founders in the homoP2_F2 scenario. "True" (A) indicates true genotype, while "GBSR" (B), "LB" (C), and "Magic" (D) indicates estimated genotype by GBScleanR, LB-Impute, and magicImpute, respectively. Proportions of reference homozygous, heterozygous, and alternative homozygous genotypes are represented by green, magenta, and blue lines, respectively. Orange lines indicate miscall rates at markers, while gray vertical lines indicate the markers at which the algorithms wrongly assigned the same genotype for all founders.



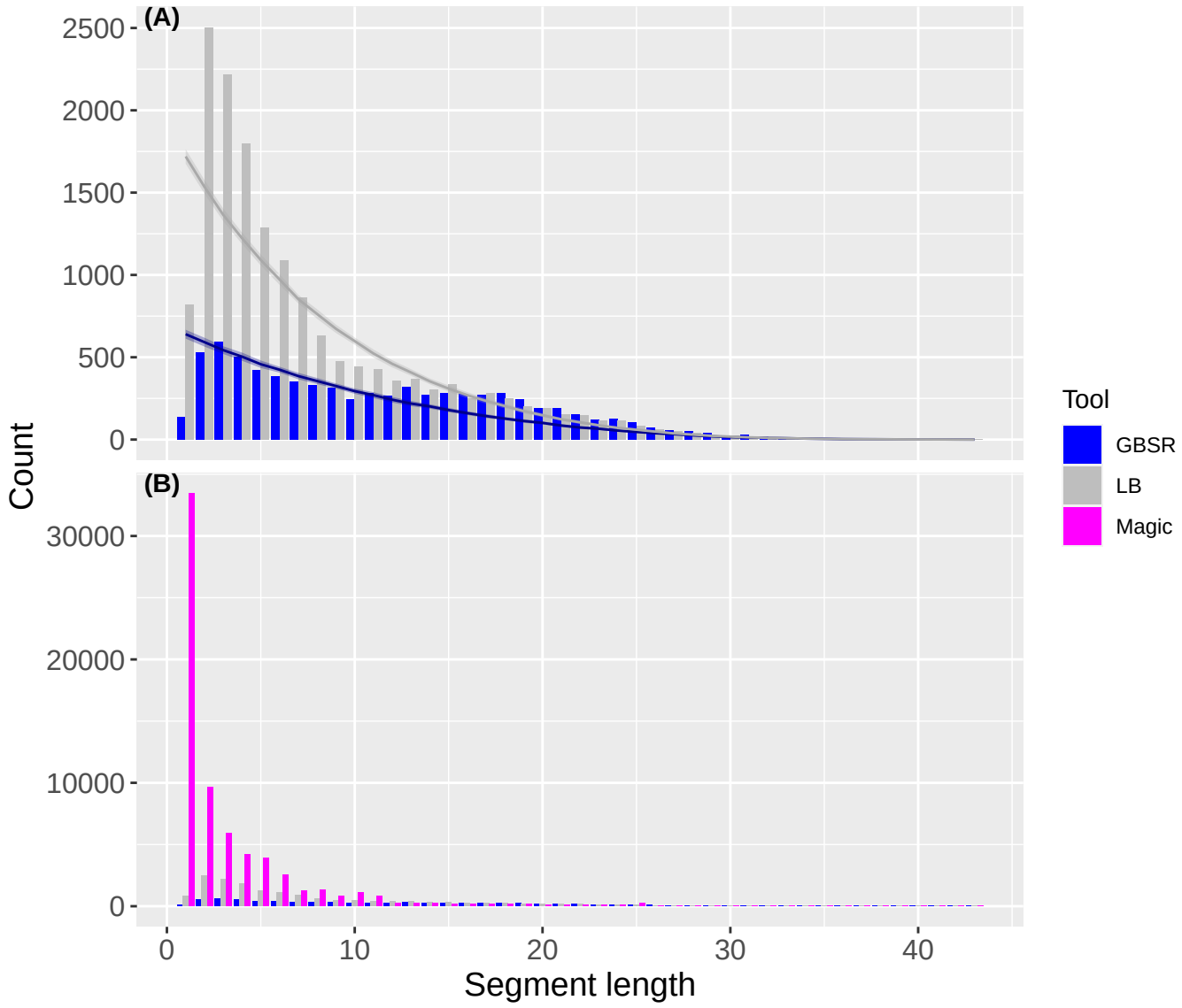
Supplemental Figure S10. Evaluation of founder genotype estimation accuracy on the noError simulation data. The plots show correct call rates in the datasets with given read depth (x axis) in the homoP2_F2, hetP2_F2, and homoP8_RIL scenarios. The number of individuals and markers in each dataset are indicated on the right side of the figure. Solid lines and dashed lines indicate the results for datasets with read depths of $3\times$ and $1\times$ for founders, respectively. "GBSR" and "GBSR_noOpt" represent GBScleanR with and without iterative optimization for w_m and e_m^{map}, while "Magic" denotes magicImpute.



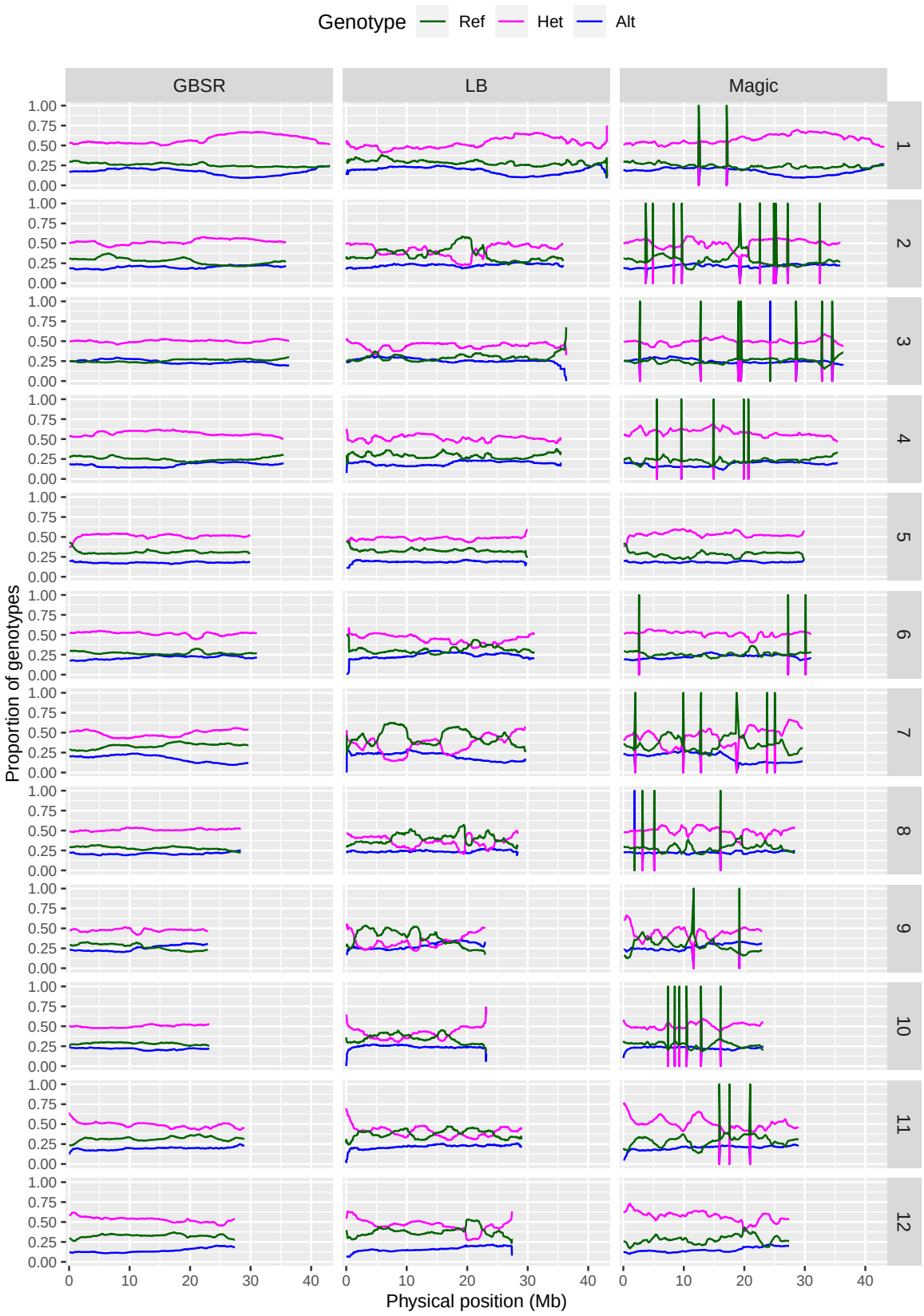
Supplemental Figure S11. Evaluation of phased genotype estimation accuracy on the noError simulation data. The plots show correct call rates in datasets with given read depths (x axis) in the homoP2_F2, hetP2_F2, and homoP8_RIL scenarios. The number of individuals and markers in each dataset are indicated on the right side of the figure. Solid lines and dashed lines indicate the results for datasets with read depths of $3\times$ and $1\times$ for founders, respectively. "GBSR" and "GBSR_noOpt" represent GBScleanR with and without iterative optimization for w_m and e_m^{map}, while "Magic" denotes magicImpute.



Supplemental Figure S12. Evaluation of phased genotype estimation accuracy on the realError simulation data. Plots show correct call rates in the datasets with 620 markers for given read depths (x axis) in the homoP2_F2, hetP2_F2, and homoP8_RIL scenarios. The number of individuals in each dataset are indicated on the right side of the figure. Solid lines and dashed lines indicate the results for the datasets with read depths of $3\times$ and $1\times$ for founders, respectively. "GBSR" and "GBSR_noOpt" represent GBScleanR with and without iterative optimization for w_m and e_m^{map}, while "Magic" denotes magicImpute.



Supplemental Figure S13. Distribution of the chromosome segment lengths between recombination breakpoints. The x-axis indicates the bins of a 1 Mb interval from 0 to 43 Mb and the y-axis indicates the counts of the chromosome segments having the lengths grouped in each bin. For better visibility, the counts for GBScleanR and LB-Impute (A) and the counts for GBScleanR, LB-Impute, and magicImpute (B) are separately shown. The darkblue line and the darkgray line indicate the expected numbers of segments in each bin without assuming the crossover interference based on the observed recombination frequencies in the results of GBScleanR and LB-Impute, respectively. The shaded areas around the lines represent standard deviations obtained from 100 times of repeated calculation of the expected numbers of segments.



Supplemental Figure S14. Genotype ratios at markers ordered along chromosomes in the real data. "GBSR", "LB", and "Magic" indicate GBScleanR, LB-Impute, and magicImpute, respectively. Proportions of reference homozygous, heterozygous, and alternative homozygous genotypes are represented by green, magenta, and blue lines, respectively. Genotype ratios for all chromosomes 12 are shown in this figure.