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

* To whom correspondence should be addressed.

Tel: +81-86-434-1208

Email: f.tomoyuki@okayama-u.ac.jp

Abstract

Reduced-representation sequencing (RRS) provides cost-effective and time-saving genotyping platforms. Although the outstanding advantage of RRS in throughput, the obtained genotype data usually contains a large number of errors. Several error correction methods employing hidden Morkov 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. However, bias does occur because of uneven amplification of genomic fragments and read mismapping. In this paper we introduce an 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 more than 25 percentage points at maximum as compared to the existing tools in simulation datasets and achieves the most reliable genotype estimation in real data even with error prone markers.

28 **Introduction**

29 The rise of next generation sequencing (NGS) has opened the door to a new generation
30 of genetic and genomic research[1]. NGS provides genotyping platforms with dramatically
31 improved flexibility and throughput that are now routinely applied to diverse fields of
32 biology in which information about the nucleotide sequences of organisms is paramount[2–
33 4]. The cost of NGS has decreased significantly over the past decade. Nevertheless,
34 obtaining a genotype via whole genome resequencing is still an expensive option for
35 researchers, particularly those working on breeding projects that involve the genomes of
36 thousands of individuals[5]. Many different RRS-based genotyping methods have been
37 introduced to meet the demands for cost-effective genotyping systems with dense markers
38 [6]. RRS is a sequencing technique using NGS with reducing sequence targets by taking
39 reads only from the limited portion of a genome[6]. Since restriction-site associated DNA
40 sequencing (RAD-seq) was first published in 2008, several derivative methods have been
41 introduced, including methods such as ddRADseq, nextRAD, ezRAD, 2b-RAD, and
42 RAD Capture[7–12]. Genotyping-BY-Sequencing (GBS) and its derivative, two-enzyme
43 GBS, are also outstanding RRS methods[13,14]. GBS has already been widely applied to
44 many crops and animals, such as rice, sorghum, chickpea, cattle, and mice[15–19]. GBS,
45 RAD-seq and its derivatives basically obtain sequences from a limited portion of a

46 genome by fragmentating genomic DNA with restriction enzymes before sequencing
47 from adapters that are ligated at the cut site. This reduction in the number of sequence
48 targets allows for multiplexed sequencing, freeing researchers from the laborious pre-
49 surveys that are usually required to find useful genotyping markers for PCR-based and
50 hybridization-based genotyping systems such as simple sequence repeat (SSR)
51 genotyping and SNP genotyping array[20,21]. However, RRS has well-known drawbacks
52 that are derived from its technological nature and include mismapped reads and the
53 undercalling of heterozygous genotypes. Although the most significant advantage of RRS
54 lies in its multiplexibility, this characteristic reduces the number of reads per NGS run.
55 Since the sequences that can be retrieved via NGS are highly stochastic, the limited
56 number of reads can result in the undercalling of heterozygotes, in which recovering only
57 one allele at a heterozygous site leads to its incorrect identification as a homozygote. The
58 lack of reads, in addition, also generate a substantial number of missing genotype calls.
59 Several error correction tools have been developed to overcome these disadvantages,
60 with two categories available, one of which imputes the missing genotype calls and the
61 other both imputes the missing genotype calls and corrects the undercallings. Beagle is
62 the most popular example of the former, and is a tool for the haplotype-phasing that is
63 frequently used in genome-wide association study (GWAS) that can infer missing

64 genotypes based on the phased haplotype information[22]. The method that is used to
65 correct the undercalling of heterozygotes was first published in 2009, and determines a
66 genotype from the allele read ratio within a sliding window[23]. Recent publications have
67 introduced more statistically sophisticated methods that use HMMs, such as FSFHap,
68 TIGER, LB-Impute, and magicImpute[24–27]. An HMM is a statistical model in which it is
69 assumed that a series of observed data is the output of a sequence of hidden states that
70 follow a Markov process[28]. In the case of genotyping data, the series of observed data and
71 the sequence of hidden states correspond to the observed genotype calls (or allele read
72 counts) and the true genotype along a chromosome, respectively. The existing error
73 correction methods provide a robust estimation of the true genotype against a background
74 that includes the undercalling of heterozygotes and missing calls that result from
75 stochastic error.

76 The existing error correction methods, however, assume constant error rates for all
77 markers. For example, those assume a 50:50 probability that a read could be obtained for
78 one of two possible alleles at a heterozygous site. Although this assumption is true in an
79 ideal situation, in practice, genotype data contains a significant number of error prone
80 markers that show skewed probabilities in the allele read acquisition as a result of actual
81 biological unevenness[29–31]. Polymorphisms that exist between the genomes of samples

82 change the fragmentation patterns of the different genomes. Even if two independent
83 reads are mapped at the same locus, the sequences of the genomic fragments from which
84 each of the two reads originated can vary in terms of GC content and length, and
85 sometimes include large insertions or deletions in the unsequenced region of the
86 fragments. Since the GC contents and the lengths of the restricted fragments are known
87 to affect the amplification efficiency, the probability of observing a read for either allele
88 may differ. These biases are likely to be more prominent in a polymorphism-rich
89 population, for example, that derived from a cross between distant relatives. Hence,
90 mismatches occur between the real data, which shows marker specific error rates, and the
91 models that assume the uniform error rate. Those mismatches then result in biased
92 genotype estimation and poor error correction accuracy.

93 Here we introduce an R package "GBScleanR" that implements an HMM-based error
94 correction algorithm. Our algorithm estimates the allele read bias and mismap rate per
95 marker and incorporates those into the HMM as parameters to capture the skewed
96 probabilities in read acquisitions. This paper demonstrates a comparison of GBScleanR
97 and two well established error correction tools; LB-Impute and magicImpute[26,27]. While
98 LB-Impute accepts only biparental populations that are derived from inbred founders,
99 magicImpute can work on bi- and multi- parental populations that are derived from both

100 inbred and outbred founders. The magicImpute algorithm is also able to estimate founder
101 genotypes simultaneously with the offspring genotypes, meaning that it does not require
102 complete and high-quality information concerning founder genotype. Similar with
103 magicImpute, GBScleanR also supports many scenarios in which experimental crosses
104 are usually conducted and is designed to estimate founder and offspring genotypes
105 simultaneously with higher robustness for error prone markers. We first show simulation
106 studies to present the accuracy and robustness of GBScleanR using simulation datasets
107 that have severe allele read biases at error prone markers. The simulation assumes three
108 scenarios; a biparental F_2 population (homoP2_F2), an outbred F_2 population (hetP2_F2),
109 and an 8-way recombinant inbred line (homoP8_RIL). We then demonstrate the reliability
110 and robustness of GBScleanR using the real data derived from a cross between distant
111 relatives of rice, which potentially contains a number of error prone markers.
112

Results

Evaluation using simulation datasets

We first evaluated the three algorithms using the simulation datasets in the scenarios homoP2_F2, hetP2_F2, and homoP8_RIL. If no allele read bias was observed, the proportion of reference allele reads per marker would be 0.5 on average. 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 S1). Some markers also showed more alternative than reference allele reads. The simulation datasets mimicking allele read biases observed in the real data were subjected to error correction via GBScleanR, magicImpute, and LB-Impute. The superiority of GBScleanR lies in the iterative parameter optimization (IPO) in which allele read biases and mismap rates of markers are iteratively estimated. Thus, we also tested GBScleanR without IPO to demonstrate the effect of IPO on genotype estimation.

Under the condition of such biases, the genotype estimations made by the algorithms was highly disturbed, except for GBScleanR with IPO (Figure 1 and Supplemental Data S1). In the homoP2_F2 scenario, GBScleanR with IPO scored the best correct genotype call rates for almost all conditions (Figure 1A-C). When the simulation datasets were generated without allowance to have no read at some markers in founders (indicated as

131 “nonzero” in Figure 1), LB-Impute showed the highest correct call rates for the datasets
132 with a 0.1x offspring read depth. However, LB-Impute also scored the highest miscall
133 rate for those datasets (Supplemental Data 1). The reduction of accuracy was observed in
134 all of the tested algorithms, when missing genotypes due to no read observation at some
135 markers were allowed for founders (shown as “allowzero” in Figure 1). Nevertheless,
136 GBScleanR was less affected by missing genotypes in founders and outperformed the
137 other algorithms. In addition, GBScleanR with IPO only kept increasing the estimation
138 accuracy as offspring read depth increased, while the other algorithms showed a plateau
139 at 0.5-20x in the results for the “nonzero” datasets (Figure 1A-C). The difference in
140 correct call rates between GBScleanR with IPO and the other algorithms reached more
141 than 25 percentage points at maximum in the datasets with a 20x offspring read depth.
142 On the other hand, the algorithms except for GBScleanR with IPO showed an increase of
143 the miscall rate as the read depth increased (Supplemental Figure 2).
144 Similar with the results in homoP2_F2, GBScleanR showed the best correct call rates in
145 almost all datasets of hetP2_F2 and the difference in the scores between the algorithms
146 reached more than 25 percentage points at maximum (Figure 1D-F). Although the
147 accuracy increased as the number of samples and reads increased, the lower absolute
148 scores were observed than those observed in the datasets of homoP2_F2, indicating the

149 difficulty of the genotype estimation on a population derived from outbred founders
150 having heterozygous genomes. Unlike the homoP2_F2 datasets, the genotype estimation
151 accuracy for the hetP2_F2 datasets was highly affected by the number of samples (Figure
152 1A-F). The increase in correct call rate between the datasets with 10 and 1000 samples
153 reached around 26 percentage points at maximum in the hetP2_F2 datasets with a 0.5x
154 offspring read depth, whereas the maximum increase of the score in the homoP2_F2
155 datasets was about only 5 percentage points (Supplemental Data 1).
156 Unlike the F2 datasets, we could not find any remarkable difference in the accuracy of
157 genotype estimation for the homoP8_RIL datasets (Figure 1G-I). This result seems to
158 have reflected the higher homozygosity in the simulated genotypes. Since the offspring
159 in the homoP8_RIL datasets had homozygous genotypes at almost all markers due to
160 repeated self-fertilization, only the reads of either allele representing the true genotype
161 could be observed even at error prone markers with allele read biases.
162
163 *Estimation of founder genotype and phased genotype*
164 GBScleanR and magicImpute have been designed to simultaneously estimate the
165 genotypes of both founders and offspring. Thus, we next compared the founder genotype
166 estimation accuracy of the algorithms. As expected from the estimation accuracy for

167 offspring, GBScleanR outperformed magicImpute (Figure 2 and Supplemental Data 2).
168 While both algorithms showed similar and high correct call rates in the homoP2_F2 and
169 homoP8_RIL datasets when “nonezero”, magicImpute resulted in less accurate founder
170 genotype estimations compared with GBScleanR for the “allowzero” datasets (Figure 2A-
171 C and G-I). Unlike the offspring genotype estimation, IPO did not affect the founder
172 genotype estimation in the homoP2_F2 and the homoP8_RIL datasets (Figure 2A-C and
173 G-I). Improvement in the estimation accuracy by IPO was observed only in the hetP2_F2
174 datasets with a relatively large number of offspring and offspring reads (Figure 2D-F).
175 This might be because heterozygotes in the founder genotypes were only assumed in the
176 hetP2_F2 datasets but not in the others; the homozygous genotypes of the inbred founders
177 were not affected by allele read biases.
178 In addition to the genotype information of offspring and founders, both GBScleanR and
179 magicImpute provide phasing information for the offspring genotypes. Phased genotype,
180 which is also referred to as haplotype, is the sequence of alleles on each of a chromosome
181 pair. Correct call rates were evaluated by scoring the proportion of estimated haplotype
182 that matched with the simulated haplotype. As expected from the superiority for both
183 founder and offspring genotype estimation, GBScleanR also outperformed magicImpute
184 in the haplotype estimation (Supplemental Figure 3 and Supplemental Data 3). Although

185 allowance of no read in founders appears to have not remarkably affected the accuracy in
186 most cases, the "nonzero" datasets resulted in better estimations than those of "allowzero"
187 in the homoP2_F2 datasets, especially in the results of magicImpute.
188
189 *Less biased estimation by GBScleanR*
190 We further evaluated the quality of the estimated genotype data by visualizing the
191 estimated genotype ratio at each marker (Figure 3 and Supplemental Figure 4 and 5). This
192 visual inspection employed the homoP2_F2 datasets that have 1000 offspring with
193 offspring reads depths of 0.5x, 3x, and 20x without allowance of no read in founders. In
194 an F_2 population derived from inbred parents, we can expect a 1:2:1 genotype ratio in
195 offspring if no segregation distortion. As shown in Figure 3, the estimations by
196 GBScleanR showed 1:2:1 segregation at almost all markers, which nearly completely
197 overlapped with the true genotype ratio throughout the chromosome, except for the
198 dataset with only 0.5x offspring read (Figure 3A-C). In addition, miscalls in the estimated
199 genotype data were distributed almost uniformly throughout the entire chromosome with
200 very small amount per marker. In contrast, LB-Impute showed severely biased genotype
201 ratios with high miscall rates clustered at several loci on the chromosome (Figure 3D-F).
202 The increase of offspring read depth made biased genotype estimation severer and

203 increased miscall rates as also shown in Supplemental Figure 2. Similar results were
204 observed in the genotypes estimated by magicImpute (Figure 3G-I). Even though less
205 miscalls were generated in magicImpute compared to LB-Impute, many regions showed
206 the biased estimation of reference homozygotes, in which reference homozygotes were
207 preferentially estimated while heterozygotes were left as missing, particularly visible in
208 the dataset with 0.5x offspring read depth (Figure 3G). The loci showing higher miscall
209 rates overlapped in the plots for LB-Impute and magicImpute. Thus, these miscalls and
210 biased genotype estimations seemed to be caused by error prone markers at which severe
211 allele read biases prevent the algorithms from correct genotype estimation. Similar results
212 were observed in the datasets with 10 and 100 offspring (Supplemental Figure 4 and 5).
213 Although the true genotype did not show 1:2:1 ratio due to the small population size in
214 the datasets with 10 offspring, GBScleanR showed the estimated genotype ratio that
215 highly overlapped with the true ratio (Supplemental Figure 4A-C). These results
216 demonstrated the robustness of genotype estimation by GBScleanR even at error prone
217 markers.
218
219 *Evaluation using real data*
220 We finally evaluated the algorithms using the real data that was the genotype data

221 obtained by GBS on an F_2 population consist of 814 individuals derived from a distant
222 cross between *Oryza sativa* and *O. longistaminata*. As usually conducted to evaluate the
223 performances of genotpye error correction tools for real data, the concordance rate of the
224 estimated genotype calls was measured using the masked genotype data derived from the
225 real data. First of all, GBScleanR scored the best concordance rate in all chromosomes
226 except for chromosome 11 in the comparison with LB-Impute and magicImpute (Figure
227 4A and Supplemental Data S4). Relatively large differences were observed in
228 chromosome 7, 8, and 10, in which GBSclenaR scored 10.5, 12.5, and 13.0 percentage
229 points higher concordance rates than magicImpute. On the other hand, chromosome 5
230 showed only a tiny difference in the concordance rate that was 1.5 percentage points
231 between GBScleanR and magicImpute. The chromosomes with severer allele read biases,
232 e.g. chromosome 7-12, seemed to tend to show smaller concordance rates than those
233 observed in the other chromosomes (Figure 4A and Supplemental Figure 1).
234 Even though GBScleanR showed better concordance rates than the others, the genotype
235 concordance does not directly mean the higher accuracy because we cannot know the true
236 genotype of the real data and there is no evidence whether the genotype calls selected for
237 masking were truly reliable or contained a large number of erroneous genotype calls. To
238 further evaluate the performance of the algorithms, we then calculated the averages of

239 missing rates and the number of recombinations in each chromosome. In the case of
240 missing rate, the highest values were scored by magicImpute for all chromosomes, while
241 GBScleanR and LB-impute showed the similar scores in a majority of chromosomes
242 (Figure 4B and Supplemental Data 4). Similar with the case of the concordance rate, the
243 chromosomes having severer allele read biases showed relatively higher missing rates. A
244 clear difference between the algorithms was observed in the number of recombinations
245 (Figure 4C and Supplemental Data 4). LB-Impute always estimated the most frequent
246 recombinations followed by magicImpute, whereas GBScleanR showed the fewest
247 numbers of recombinations in all chromosomes. The previous study reported that the
248 shortest and longest chromosomes in the estimated genetic map of rice are chromosome
249 10 and chromosome 1, which are 83.8 and 181.8 cM, respectively[32]. Therefore, the
250 expected numbers of recombinations per chromosome would range 1.6-3.6. The results
251 of LB-Impute seemed to fit to the expected values more than those of GBScleanR. To
252 confirm whether LB-Impute truly estimated more accurate numbers of recombination
253 events, we measured the genomic segment lengths between the recombination
254 breakpoints in chromosomes (Table 1 and Supplemental Data 5). We found that LB-
255 Impute estimated much more numbers of short segments. The genotype data estimated
256 by GBScleanR had 261 and 668 segments in the ranges of 0-1 and 1-2 Mb, respectively

257 (Table 1). On the other hand, LB-Impute estimated the genotype data with 2164 and 4166
258 of 0-1 and 1-2 Mb segments, respectively. The magicImpute algorithm generated a fewer
259 number of short segments than LB-Impute but a larger number than GBScleanR, which
260 were 627 and 1716. In addition, GBScleanR, magicImpute, and LB-Impute estimated 58,
261 235, and 948 of 0-1 Mb segments that were heterozygous and flanked by the same
262 homozygous genotypes at the both sides, which indicate double crossovers within 1 Mb
263 stretches (Table 1). The probability of a double crossover within 1 Mb is approximately
264 0.08% when 1 Mb equals to 4 cM. Thus, the expected number of double crossovers is
265 less than 2 in the population with 814 individuals derived from 1628 gametes. Even
266 though the 58 double crossovers within 1Mb stretches generated by GBScleanR was too
267 much frequent for an ideal F_2 population, considering the distant cross to generate the
268 given population, these results suggested that GBScleanR estimated more probable and
269 acceptable numbers of recombinations. On the other hand, unexpectedly large numbers
270 of short segments were estimated by LB-Impute and magicImpute, which indicated
271 double crossovers at approximately 4- and 16-fold higher frequencies than those
272 estimated by GBScleanR. Those excess crossovers in short distances seemed to push up
273 the numbers of recombinations artificially.

274 At last, we measured the segregation distortion level in the estimated genotype data

275 (Figure 4D and Supplemental Data 4). Since our real data was obtained from the rice F_2
276 population, it could be expected that any markers show a genotype segregation ratio of
277 1:2:1 for reference homozygotes, heterozygotes, and alternative homozygotes although
278 segregation distortion due to the reproductive isolation could occur. Relatively low
279 distortion level was observed in the genotype data estimated via GBScleanR compared
280 with those by LB-Impute and magicImpute (Figure 4D). Chromosome 1 was the most
281 distorted one in the genotype data estimated by GBScleanR, while the other algorithms
282 also showed similar distortion levels in this chromosome. On the other hand, the highest
283 distortion level was observed in chromosome 7 in the cases of LB-Impute and
284 magicImpute (Figure 4D). Unlike chromosome 1, GBScleanR showed a much smaller
285 distortion level that was about a half of the levels scored by the other algorithms. In
286 contrast, the three algorithms all resulted in the mildest distortion in chromosome 3
287 (Figure 4D). To visually inspect the segregation distortion levels, we created the line plots
288 of the estimated genotype ratios (Figure 5 and Supplemental Figure 6). Chromosomes 1,
289 3, and 7 were selected to demonstrate the negative impact of allele read biases (Figure 5).
290 The line plots indicated that all of the algorithms estimated quite similar genotypes for
291 chromosome 1 and 3, while clearer differences were observed in the plots for
292 chromosome 7. As fewer markers in chromosome 1 and 3 showed sever allele read biases,

293 while chromosome 7 showed relatively higher number of markers preferentially
294 accumulated reference allele reads, chromosome 1 and 3 might have less error prone
295 markers (Supplemental Figure 2). These chromosomes with relatively less error prone do
296 not seem to have made a sever negative impact on the estimation even by LB-Impute and
297 magicImpute, and resulted in the similar estimation with GBScleanR (Figure 5A-F).
298 Since all algorithms resulted in similar distortion patterns around 30 Mb of chromosome
299 1, the distortion on chromosome 1 was not likely to be due to the misestimation but the
300 true genetic characteristic of this population (Figure 5A-C). On the other hand, slight
301 distortions at 5, 10, 35 Mb of chromosome 3 that were observed in the genotypes
302 estimated by LB-Impute and magicImpute, but not by GBScleanR, might be artifacts
303 produced by biased estimation at error prone markers (Figure 5D-F). In the case of
304 chromosome 7, however, LB-Impute and magiImpute obviously failed in estimating the
305 genotypes and produced artificially distorted genotype ratio patterns (Figure 5G-I). In
306 contrast, GBScleanR showed a smooth and less fluctuating change of the genotype ratio,
307 which was an acceptable pattern as the genotype ratio of a F_2 population, throughout the
308 chromosome although severe distortion was observed probably due to true segregation
309 distortion (Figure 6I). The other chromosomes also exhibited similar results to those
310 observed for chromosomes 1, 3, and 7 (Supplemental Figure S14). LB-Impute and

311 magicImpute always resulted in highly biased genotype estimation at many loci and
312 fluctuated genotype ratio throughout the chromosomes. In clear contrast to those results,
313 GBScleanR showed relatively smooth and gradual change of genotype ratio throughout
314 the chromosomes. Even though some loci showed distortion of genotype ratio, majority
315 of chromosomal region had ratio relatively near to 1:2:1 in almost all chromosomes in the
316 result of GBScleanR than those of the other algorithms. Overall, the performance
317 evaluation on the real data demonstrated the superiority and robustness of GBScleanR
318 against error prone markers.
319
320 *Comparing the running times*
321 In addition to the comparisons of the algorithms in the aspects of accuracy and robustness,
322 we also compared the running times using the simulation datasets. As shown in Table 2,
323 GBScleanR completed all processes of genotype estimation faster than LB-Impute and
324 magicImpute for the simulation datasets of biparental F_2 populations including
325 homoP2_F2 and hetP2_F2. In the case of the largest homoP2_F2 dataset that consists of
326 1000 samples, GBScleanR took only 15 seconds even with four cycles of IPO, while
327 magicImpute and LB-Impute consumed 6.5 and 122.3 times as long. Although the
328 magicImpute algorithm could only work properly for the hetP2_F2 datasets with detailed

329 pedigree information given by a pedigree file, while the algorithm should accept simple
330 specification of pedigree information given by a string like “Pairing, Sibling” indicating
331 mating scheme of a given population (See *Input parameter settings* in the Materials and
332 Methods section), we evaluated running times of magicImpute with or without pedigree
333 files for hetP2_F2. Even for the hetP2_F2 datasets, which require a larger number of
334 calculations than homoP2_F2, GBScleanR with IPO outperformed magicImpute (Table
335 2). Detailed pedigree information given as a pedigree file made magicImpute quite slower
336 than that with simple pedigree information. On the other hand, a 2.5-times longer
337 calculation at maximum was observed in GBScleanR with four cycles of IPO compared
338 to magicImpute for the homoP8_RIL datasets (Table 2). Nevertheless, IPO was not
339 required basically for RIL populations as shown in Figure 2, and GBScleanR without IPO
340 could finish the calculations 35% faster than magicImpute even in the homoP8_RIL
341 dataset with 1000 samples.
342

Discussion

Advantage of GBScleanR

The present study introduced an 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 techniques, has a chance to contain such error prone markers even after trying to filter out erroneous markers. 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 of noisy GBS data by incorporating the unevenness of allele reads into the model. Although all of the algorithms resulted in poor estimation accuracy for the simulation datasets with less than 1x of the offspring read depth, in which correct genotype call rates were smaller than 80%, especially in the hetP2_F2 datasets, GBScleanR with IPO basically outperformed the other algorithms in any situation and resulted in the most stable and accurate genotype estimation. When the offspring reads

362 more than 3x could be provided for a population derived from inbred founders, including
363 homoP2_F2 and homoP8_RIL, GBScleanR made nearly 95-99% accurate estimation
364 (Figure 1 and Supplemental Data 1). On the other hand, biparental populations derived
365 from outbred founders (hetP2_F2) seemed to require a larger number of reads for precise
366 estimation. In addition, the number of samples in the given population exhibited a
367 relatively large effect on the estimation of founder genotypes and the accuracy of founder
368 genotype estimation affected on the accuracy of offspring genotype estimation (Figure 1
369 and 2). Our simulation study presented the difficulty of genotype estimation of a
370 population derived from outbred founders compared with that derived from inbred
371 founders. Interestingly, the increase of offspring reads could not improve genotype
372 estimation accuracy for nosy GBS data by LB-Impute and magicImpute and rather
373 decreased accuracy with the increase of miscall rates (Figure 1 and Supplemental Figure
374 2). Since GBScleanR without IPO also showed similar results with LB-Impute and
375 magicImpute, those unexpected increases of miscalls were caused by biased genotype
376 estimation on error prone markers that had uneven read acquisition ratio of reference and
377 alternative alleles. The larger number of reads per maker would be expected to increase
378 the reliability of genotype calling. However, in the case of genotype data with error prone
379 markers, our study demonstrated that the algorithms without concern for allele read biases

380 would mess up the genotype estimation.

381 The algorithm evaluation using the real data that potentially contains error prone
382 markers also exhibited the superiority of GBSclenaR. The result clearly illustrated the
383 negative impact of error prone markers having sever allele read biases and how the
384 iterative estimation of error pattern by GBScleanR outperforms the existing algorithms
385 (Figure 5 and Supplemental Figure 6). The estimated genotype data for the real data by
386 LB-Impute resulted in the lowest missing rate and the highest recombination frequency
387 compared with the other algorithms (Figure 4). Together with the relatively poor correct
388 call rate in the simulation data (Figure 2), the LB-Impute algorithm tends to try filling up
389 genotype calls even those are relatively less reliable and results in biased genotype
390 estimation around error prone markers. Therefore, LB-Impute may deteriorate genotype
391 data quality by enhancing wrong genotype calls using erroneous information of biased
392 read counts if given genotype data contains error prone markers. Unlike LB-Impute, the
393 magicImpute algorithm resulted in relatively small numbers of recombinations, while it
394 scored the highest missing rates in all chromosomes (Figure 4). Although magicImpute
395 seems to set missing to unreliable estimation, biased allele read ratios at error prone
396 markers cannot be properly handled in the algorithm. As the result of biased genotype
397 estimation around the error prone markers, magicImpute also resulted in highly

398 fluctuating patterns of genotype ratio throughout almost all chromosomes, like LB-
399 Impute did. On the other hand, GBScleanR could score the best concordance rates in all
400 chromosomes (Figure 4). In addition, although the numbers of recombinations found in
401 each chromosome did not fit the expected numbers according to the genetic map of rice
402 estimated in a previous study[32], the lowest number of the improbable double
403 recombinations within short stretches was observed in the genotype data estimated by
404 GBScleanR (Table 1). These results, together with the higher accuracy in the simulation
405 datasets, indicates that GBScleanR is able to precisely incorporate the information of
406 biased read counts at error prone markers and give robust and accurate estimation. The
407 plots of the estimated genotype ratios also illustrate that GBScleanR can largely reduce
408 the negative impacts of biased read counts and minimize artificial distortions even in the
409 chromosomes containing a relatively large number of error prone markers (Figure 5 and
410 Supplemental Figure 6).

411 The distortion level and the recombination frequency have large effects on downstream
412 genetic analyses and also those values themselves are research targets. Any genetic and
413 genomic study relies on the given genotype data and the accuracy of genotyping has a
414 crucial effect on the quality of such research. The accuracy of genotyping can be
415 evaluated by three criteria: correct calls, miscalls, and missing calls. Usually, miscalls

416 have a more destructive effect on post-genotyping analyses than missing calls, because
417 post-genotyping analyses regularly suppose all genotype calls to be true even if those
418 include miscalls, and potentially generate misleading results. Furthermore, if the wrongly
419 genotyped markers are clustered in a particular region, all statistical calculations for that
420 region will be unreliable. Our study demonstrated the reliability of GBScleanR as the
421 support tool for genetic studies that handle genotype data even with or without error prone
422 markers.
423
424 *GBScleanR requires no strict filtering*
425 In this study, we conducted a filtering based on a quantile of read counts for the real data
426 to filter out genotype calls supported by overrepresented sequences that are possibly
427 derived from paralogous and repetitive sequences. No one is able to exactly distinguish
428 which reads were wrongly mapped. The error correction algorithms basically "believe"
429 that the genotype calls supported by the higher number of reads were more probable,
430 while fewer reads give the algorithms more space to guess the true genotype. The large
431 number of reads with overrepresented mismapped reads does not increase the reliability
432 of the genotype calls but rather causes misestimation of homozygous calls as
433 heterozygous. Although GSBcleanR tries to detect mismapped reads, we recommend to

434 filter out genotype calls supported by overrepresented sequences to reduce their negative
435 impact on estimation. Nevertheless, GBScleanR accepts the error prone markers in a
436 given dataset and recognizes the error pattern via IPO. Therefore, no strict filtering of
437 markers based on allele frequency and missing rate is required, which is regularly
438 performed for RRS-based genotyping data[33,34]. These filtering steps usually remove a
439 large portion of the detected SNP markers[35–38]. In the case of our real data, filtering to
440 remove possible error prone markers based on, for example, minor allele frequency > 5%
441 and missing rate < 20% left only 162 out of 5035 SNPs for the 12 chromosomes in the
442 real data. Markers that would be filtered out may possibly contain information about the
443 true genotype at the markers even if the data are partial and skewed. GBScleanR is able
444 to integrate those less informative markers into the error correction process by tweaking
445 the HMM. Since there is no clear guideline for filtering strength, which should differ
446 depending on the dataset used, filtering criteria are left in the hands of researchers.
447 GBScleanR allows users to skip such an arbitrary step and improves the cost efficiency
448 of the RRS-based genotyping system by reducing the number of unused reads obtained
449 via costly NGS.
450
451 *Limitations and perspectives of GBScleanR*

452 The current implementation of GBScleanR treats all chromosomes as diploid autosomes.
453 Although sex chromosomes can be processed by our algorithm, the calculation of
454 transition probabilities between autosomes and sex chromosomes would be expected to
455 differ, in which any recombination does not occur in either male or female individuals.
456 The algorithm therefore overestimates the number of recombinations in the sex
457 chromosomes, while Supplemental Figure 8B shows that an overrated recombination
458 frequency does not produce any severe effects on genotype estimation. A previously
459 published paper introduced a model that can be used to infer the number of
460 recombinations in the sex chromosomes of experimental populations[39]. We plan to further
461 implement this model into GBScleanR in a future update.
462 In addition, there are obvious demands for applying GBScleanR to polyploid organisms.
463 Several dosage estimation algorithms have already been implemented some tools[40–42].
464 Nevertheless, all of these infer the dosage at each marker independently and do not
465 incorporate information at adjacent markers, as performed in an HMM. The dosage
466 estimation using an HMM may increase the robustness against error prone markers, while
467 increasing the complexity of the algorithm at the same time. The codes of GBScleanR
468 have been basically organized to allow us to expand its functions for polyploid organisms.
469 Thus, GBScleanR will support polyploids in the future.

470 The recent remarkable development in third-generation sequencing, named Sequel by
471 Pacific Bioscience and MinION by Oxford Nanopore Technologies, enables us to obtain
472 high quality genome sequences at reasonable cost[43,44]. Reference genome information for
473 any organism is expected to become available soon. Thus, RRS-based genotyping
474 systems that are underpinned by a flexible robust error correction tool such as GBScleanR
475 will become practical platforms for use in breeding programs that require a prompt and
476 efficient scheme under limited periods and budgets. Distant relatives including the
477 ancestral wild species of domesticated species are important genetic resources,
478 particularly in plant breeding, that can improve abiotic and biotic stress resistance[45–47].
479 For such genetic studies and breeding projects using wild species, GBScleanR could be
480 the first choice as a fundamental tool for genotype identification coupled with an RRS-
481 based genotyping system.
482

483 **Methods**

484 *Overview of the GBScleanR package*

485 GBScleanR is an R package and consists of a main tool that conducts error correction and
486 utility functions that handle genotype data in the variant call format (VCF). Supplemental
487 Figure 7 presents the data analysis work-flow and the related functions that are
488 implemented in GBScleanR. The algorithm is designed to estimate the true genotype calls
489 along chromosomes from the given allele read counts in a VCF file that is generated by
490 SNP callers such as GATK and TASSEL-GBS[48,49]. The current implementation supports
491 genotypic data for mapping populations that are derived from two or more diploid
492 founders followed by selfings, sibling crosses, or random crosses, for example F_2 and 8-
493 way RILs. Our method assumes that markers are biallelic and ordered along the
494 chromosomes, and the reads are mapped onto a reference genome sequence. To access
495 the large amounts of genotype data required, an input VCF file is first converted into a
496 genomic data structure (GDS) file[50]. This conversion automatically filter out non-biallelic
497 markers.

498 GBScleanR provides functions for data visualization, filtering, and loading/writing a
499 VCF file. Furthermore, the data structure of the GDS file created via this package is
500 compatible with that used in the SeqArray package and can be further converted to a

501 derivative GDS format used in the SNPRelate, GWASTools and GENESIS packages,
502 which are designed to handle large amounts of variant data and conduct regression
503 analysis[50,52–54]. Therefore, GBScleanR can be built in a pipeline for an genetic study that
504 allows us smooth access to the given large genotype data for visualization, filtering, error
505 correction, and regression analysis without the requirement for extra data format
506 conversion steps.
507
508 *Modeling for error correction*
509 The error correction algorithm of GBScleanR is based on the HMM and treats the
510 observed allele read counts for each SNP marker along a chromosome as outputs from a
511 sequence of latent true genotypes. Our model supposes that a population of $N^{\mathrm{o}} \geq 1$
512 sampled offspring is originally derived from crosses between $N^{\mathrm{f}} \geq 2$ founder individuals.
513 The founders can be inbred lines with homozygotes at all markers or outbred lines in
514 which markers would be heterozygous. Only one chromosome is considered for modeling
515 due to the independence of chromosomes. Let $\boldsymbol{Y}^{\mathrm{o}} = \{y_{mi}^{\mathrm{o}}\}_{m=1\ldots M, i=1\ldots N^{\mathrm{o}}}$ denote the
516 observed allele read counts at marker *m* in offspring *i*. The element y_{mi}^{o} consists of two
517 values, y^{ref} and y^{alt}, which represent the reference read count and the alternative read
518 count, respectively. Similarly, the observed allele read counts at marker *m* in founder *j* are

represented by $\boldsymbol{Y}^{\mathrm{f}} = \{y_{mj}^{\mathrm{f}}\}_{m=1\ldots M, j=1\ldots N^{\mathrm{f}}}$. On the other hand, the matrices for hidden true offspring genotypes and hidden true founder genotypes are represented by $\boldsymbol{X}^{\mathrm{o}} = \{x_{mi}^{\mathrm{o}}\}_{m=1\ldots M, i=1\ldots N^{\mathrm{o}}}$ and $\boldsymbol{X}^{\mathrm{f}} = \{x_{mj}^{\mathrm{f}}\}_{m=1\ldots M, j=1\ldots N^{\mathrm{f}}}$, respectively. The element x_{mi}^{o} takes a value of 0, 1, or 2 to indicate the reference homozygote, heterozygote, and alternative homozygote genotype without phasing information. Unlike the offspring genotype matrix, the founder genotype x_{mj}^{f} stores the phased genotype as $x_{mj}^{\mathrm{f}} = (x^1, x^2)$ 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 x_i^{o}, does not follow a Markov process, while the sequence of the descendent haplotypes h_i^{o} along a pair of chromosomes that is derived from N^{f} founders does. To enable our estimation problem to be solved in the HMM framework, $\boldsymbol{H}^{\mathrm{o}} = \{h_{mi}^{\mathrm{o}}\}_{m=1\ldots M, i=1\ldots N^{\mathrm{o}}}$ is introduced to represent the matrix of the phased descendent haplotypes. The element h_{mi}^{o} 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 $\leq N^{\mathrm{f}}$ if all founders are inbred or $\leq 2N^{\mathrm{f}}$ if outbred, to indicate the origins of the

descendent haplotypes.

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

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

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

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

$$P(\boldsymbol{X}^{\mathrm{f}}, \boldsymbol{Y}^{\mathrm{f}}) = P(\boldsymbol{Y}^{\mathrm{f}} | \boldsymbol{X}^{\mathrm{f}}) P(\boldsymbol{X}^{\mathrm{f}})$$

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

$$P(\boldsymbol{H}^{\mathrm{o}}, \boldsymbol{Y}^{\mathrm{o}} | \boldsymbol{X}^{\mathrm{f}}, \boldsymbol{Y}^{\mathrm{f}}) = \prod_{i=1}^{N^{\mathrm{o}}} P\left(y_i^{\mathrm{o}} | h_i^{\mathrm{o}}, \boldsymbol{X}^{\mathrm{f}}\right) P(h_i^{\mathrm{o}} | \boldsymbol{X}^{\mathrm{f}}).$$

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

$$P(\boldsymbol{H}^{\mathrm{o}}, \boldsymbol{X}^{\mathrm{f}}, \boldsymbol{Y}^{\mathrm{o}}, \boldsymbol{Y}^{\mathrm{f}}) = \prod_{i=1}^{N^{\mathrm{o}}} \left\{ \prod_{m=1}^{M} P(y_{mi}^{\mathrm{o}} | h_{mi}^{\mathrm{o}}, \boldsymbol{x}_m^{\mathrm{f}}) \prod_{m=2}^{M} P(h_{mi}^{\mathrm{o}} | h_{m-1,i}^{\mathrm{o}}, \boldsymbol{x}_{m-1}^{\mathrm{f}}) \, P(h_{1i}^{\mathrm{o}} | \boldsymbol{x}_1^{\mathrm{f}}) \right\}$$

$$\times \prod_{m=1}^{M} P\left(\boldsymbol{y}_m^{\mathrm{f}} \middle| \boldsymbol{x}_m^{\mathrm{f}}\right) P\left(\boldsymbol{x}_m^{\mathrm{f}}\right),$$

where $P\left(y_{mi}^{\mathrm{o}} \middle| h_{mi}^{\mathrm{o}}, \boldsymbol{x}_m^{\mathrm{f}}\right)$, $P\left(h_{mi}^{\mathrm{o}} \middle| h_{m-1,i}^{\mathrm{o}}, \boldsymbol{x}_{m-1}^{\mathrm{f}}\right)$ and $P\left(h_{1i}^{\mathrm{o}} \middle| \boldsymbol{x}_1^{\mathrm{f}}\right)$ correspond to the emission probability, the transition probability, and the initial probability of the HMM, respectively, and $P\left(\boldsymbol{y}_m^{\mathrm{f}} \middle| \boldsymbol{x}_m^{\mathrm{f}}\right)$ is the probability of observing read counts for the founders at marker m when the combination of the true genotypes of the founders is $\boldsymbol{x}_m^{\mathrm{f}}$. We assume that all SNP markers are biallelic and that $P\left(\boldsymbol{x}_m^{\mathrm{f}}\right)$ 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 $\boldsymbol{h}_{mi}^{\mathrm{o}}$ 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_{mi}^{o} from x_{mi}^{o}, but not from h_{mi}^{o} directly. Therefore, to obtain the emission probability $P\left(y_{mi}^{\mathrm{o}} \middle| h_{mi}^{\mathrm{o}}, \boldsymbol{x}_m^{\mathrm{f}}\right)$, three additional probabilities are introduced; $P\left(y_{mi}^{\mathrm{o}} \middle| x'^{\mathrm{o}}_{mi}, e^{\mathrm{seq}}, w_m\right)$, $P\left(x'^{\mathrm{o}}_{mi} \middle| x_{mi}^{\mathrm{o}}, e_m^{\mathrm{map}}\right)$, and $P\left(x_{mi}^{\mathrm{o}} \middle| h_{mi}^{\mathrm{o}}, \boldsymbol{x}_m^{\mathrm{f}}\right)$, where e^{seq} and e_m^{map} 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_m^{map}. Combining these parameters allows the emission probability to be replaced with $l^{\mathrm{o}}_{mi} = P\left(y^{\mathrm{o}}_{mi} \middle| h^{\mathrm{o}}_{mi}, \boldsymbol{x}^{\mathrm{f}}_m, e_m^{\mathrm{map}}, e^{\mathrm{seq}}, w_m\right)$ and it holds that

$$l^{\mathrm{o}}_{mi} = \sum_{x^{\mathrm{o}}_{mi}} P\left(y^{\mathrm{o}}_{mi} \middle| x^{\mathrm{o}}_{mi}, e_m^{\mathrm{map}}, e^{\mathrm{seq}}, w_m\right) P\left(x^{\mathrm{o}}_{mi} \middle| h^{\mathrm{o}}_{mi}, \boldsymbol{x}^{\mathrm{f}}_m\right),$$

$$P\left(y^{\mathrm{o}}_{mi} \middle| x^{\mathrm{o}}_{mi}, e_m^{\mathrm{map}}, e^{\mathrm{seq}}, w_m\right) = \sum_{x'^{\mathrm{o}}_{mi}} P\left(y^{\mathrm{o}}_{mi} \middle| x'^{\mathrm{o}}_{mi}, e^{\mathrm{seq}}, w_m\right) P\left(x'^{\mathrm{o}}_{mi} \middle| x^{\mathrm{o}}_{mi}, e_m^{\mathrm{map}}\right),$$

where $P\left(x^{\mathrm{o}}_{mi} \middle| h^{\mathrm{o}}_{mi}, \boldsymbol{x}^{\mathrm{f}}_m\right)$ 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 $h^{\mathrm{o}}_{1,1} = (h^1 = 1, h^2 = 4)$ at marker 1, indicating that the haplotypes descended from the first chromosome of founder 1 and the second chromosome of founder 2 and if the founder genotypes are $x^{\mathrm{f}}_{1,1} = (0,0)$ and $x^{\mathrm{f}}_{1,2} = (0,1)$, the genotype can only be heterozygotic $x^{\mathrm{o}}_{1,1} = 1$.

We assume that the observed read counts $y^{\mathrm{o}}_{mi} = \left\{y^{\mathrm{ref}}, y^{\mathrm{alt}}\right\}$ follow binomial distribution

$$P\left(y^{\mathrm{o}}_{mi} \middle| x'^{\mathrm{o}}_{mi} = 0, e^{\mathrm{seq}}, w_m\right) \propto (1 - e^{\mathrm{seq}})^{y^{\mathrm{ref}}} (e^{\mathrm{seq}})^{y^{\mathrm{alt}}},$$

$$P\left(y^{\mathrm{o}}_{mi} \middle| x'^{\mathrm{o}}_{mi} = 1, e^{\mathrm{seq}}, w_m\right) \propto (1 - w_m)^{y^{\mathrm{ref}}} (w_m)^{y^{\mathrm{alt}}},$$

$$P\left(y_{mi}^{\mathrm{o}} \middle| x'^{\mathrm{o}}_{mi} = 2, e^{\mathrm{seq}}, w_m\right) \propto (e^{\mathrm{seq}})^{y^{\mathrm{ref}}}(1 - e^{\mathrm{seq}})^{y^{\mathrm{alt}}},$$

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[24,26,27]. The normalization constant for $P\left(y_{mi}^{\mathrm{o}} \middle| x'^{\mathrm{o}}_{mi}, e^{\mathrm{seq}}, w_m\right)$ 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 homozygous. This type of error is accounted for by summing over x'^{o}_{mi} in the joint probability $P\left(y_{mi}^{\mathrm{o}} \middle| x'^{\mathrm{o}}_{mi}, e^{\mathrm{seq}}, w_m\right) P\left(x'^{\mathrm{o}}_{mi} \middle| x_{mi}^{\mathrm{o}}, e_m^{\mathrm{map}}\right)$. We assume that e_m^{map} can take two values $\left(e^{\mathrm{ref}}, e^{\mathrm{alt}}\right)$, 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 $P\left(x'^{\mathrm{o}}_{mi} \middle| x_{mi}^{\mathrm{o}}, e_m^{\mathrm{map}} = \left(e^{\mathrm{ref}}, e^{\mathrm{alt}}\right)\right)$ takes the values listed in Table 3.

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

$$\boldsymbol{l}_m^{\mathrm{f}} \propto \prod_{j}^{N^{\mathrm{f}}} P\left(y_{mj}^{\mathrm{f}} \middle| x_{mj}^{\mathrm{f}}, e^{\mathrm{seq}}, w_m\right),$$

where the mismapping of reads in the founders is not considered. The normalization constant for $\boldsymbol{l}_m^{\mathrm{f}}$ is not shown. To omit the less probable founder genotypes, the probability $P\left(y_{mj}^{\mathrm{f}} \middle| x_{mj}^{\mathrm{f}}, e^{\mathrm{seq}}, w_m\right)$ is set at 0 if the result is less than 0.01.

Transition probability

The transition probability $P\left(h_{mi}^{\mathrm{o}} \middle| h_{m-1,i}^{\mathrm{o}}, \boldsymbol{x}_{m-1}^{\mathrm{f}}\right)$ is used to describe the probability that a crossover occurs between adjacent markers. The sequences of the descendent haplotypes are modelled using a continuous-time Markov chain, as described previously[55,56]. The probability that the haplotype state transitions from $h_{m-1,i}^{\mathrm{o}}$ to h_{mi}^{o} between the markers can be obtained using

$$P\left(h_{mi}^{\mathrm{o}} \middle| h_{m-1,i}^{\mathrm{o}}\right) = e^{\boldsymbol{Q} d_m}, m \neq 1,$$

where $\mathbf{Q}$ denotes the transition-rate matrix that defines the rates at which a state transition occurs per unit of time.

The parameter $\mathbf{Q}$ is calculated based on the idea of junction densities following the modelling framework described in Zheng *et al.*, 2014 and Zheng *et al*, 2015[55,56]. The junction densities define the expected numbers of junction types per Morgan. If $\mathrm{j}(abcd)$ represent a junction density of the type $(abcd)$ where ab indicates a pair of descendent

625 haplotypes at a marker and cd denotes that of the next marker located at the other side
626 of the junction. Each of a, b, c, d can take one of the integers indicating the origin of
627 the haplotype. $\mathrm{j}(1112)$ indicates one junction from haplotype 1 to 2 on one chromosome,
628 with the same haplotype maintained on the other chromosome. In the case of a biparental
629 population derived from inbred founders, possible junction types are $\mathrm{j}(1111)$, $\mathrm{j}(1212)$,
630 $\mathrm{j}(2222)$, $\mathrm{j}(1112)$, $\mathrm{j}(1211)$, $\mathrm{j}(1221)$, $\mathrm{j}(1122)$, and the junction types in which the
631 order of numbers are swapped, e.g $\mathrm{j}(2121)$. As genetic markers are considered
632 sufficiently dense in RRS-based genotype data, only one junction between adjacent
633 markers is expected at most. Therefore, the rate in the transition-rate matrix $\mathbf{Q}$ was set
634 to 0 for the junction types with more than one junction e.g., $\mathrm{j}(1221)$ and $\mathrm{j}(2112)$.
635 Although $\mathrm{j}(1122)$ also has two junctions, these could be observed in a later generation
636 because the chromosomes are both derived from one identical chromosome that had a
637 junction between haplotypes 1 and 2 in a preceding generation. The possible junctions
638 would be selected based on pedigree information. There is no chance that a third
639 haplotype, e.g., $\mathrm{j}(1213)$, can be obtained in a biparental population, but it is possible in
640 a population that is derived from four or more founders, such as 8-way RILs.
641 The matrix exponential is obtained by Higham's algorithm, which is implemented in the
642 "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, $\mathbf{Q}$ 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, d_m is calculated based on the physical distance and the expected genetic distance:

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

where p_m represents the physical position of marker *m* and E^{d} is the expected genetic distance per mega base pairs that should be specified as input parameter.

For m = 1, the initial probability $P\left(h_{1i}^{\mathrm{o}} \middle| \boldsymbol{x}_1^{\mathrm{f}}\right)$ 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 can be rewritten as shown below.

$$P\left(\boldsymbol{H}^{\mathrm{o}}, \boldsymbol{X}^{\mathrm{f}}, \boldsymbol{Y}^{\mathrm{o}}, \boldsymbol{Y}^{\mathrm{f}}\right) = \prod_{m=1} l_m^{\mathrm{f}} P\left(\boldsymbol{x}_m^{\mathrm{f}}\right) \prod_{i=1} \left\{ \prod_{m=1} l_{mi}^{\mathrm{o}} \prod_{m=2} P\left(h_{mi}^{\mathrm{o}} \middle| h_{m-1,i}^{\mathrm{o}}, \boldsymbol{x}_{m-1}^{\mathrm{f}}\right) P\left(h_{1i}^{\mathrm{o}} \middle| \boldsymbol{x}_1^{\mathrm{f}}\right) \right\}$$

The Viterbi algorithm is used to maximize the joint probability and estimate the most probable haplotype sequences[57]. 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 $v_{mi}^{\mathrm{o}}(h_{mi}^{\mathrm{o}}|\boldsymbol{x}_m^{\mathrm{f}})$ and founders $v_m^{\mathrm{f}}(\boldsymbol{x}_m^{\mathrm{f}}|\boldsymbol{x}_{m-1}^{\mathrm{f}})$, and records the Viterbi paths for the sequences of offspring haplotypes $\psi_{mi}^{\mathrm{o}}(h_{mi}^{\mathrm{o}}|\boldsymbol{x}_m^{\mathrm{f}})$ and founder genotypes $\psi_m^{\mathrm{f}}(\boldsymbol{x}_m^{\mathrm{f}})$, respectively. The recursive calculation via an integrated Viterbi algorithm proceeds as follows.

Initialization at m = 1:

The initialization starts with simply calculating the probability to have h_{1i}^{o} when the combination of the founder genotypes $\boldsymbol{x}_1^{\mathrm{f}}$ was given.

$$v_{1i}^{\mathrm{o}}(h_{1i}^{\mathrm{o}}|\boldsymbol{x}_1^{\mathrm{f}}) = l_{1i}^{\mathrm{o}} P(h_{1i}^{\mathrm{o}}|\boldsymbol{x}_1^{\mathrm{f}}),$$

where $P(h_{1i}^{\mathrm{o}}|\boldsymbol{x}_1^{\mathrm{f}})$ is the initial probability that the ith offspring has the combination of haplotypes $h_{1i}^{\mathrm{o}} = (h^1, h^2)$ at marker m.

Recursion at $2 \leq \mathrm{m} \leq \mathrm{M}$:

The recursive calculation can be separated into several steps. As the first step, the algorithm calculates the probabilities of state transitions from marker m-1 to m as shown below.

$$\tau_{mi}\left(h_{mi}^{\mathrm{o}}\middle|h_{m-1,i}^{\mathrm{o}},\boldsymbol{x}_{m-1}^{\mathrm{f}}\right)=P\left(h_{mi}^{\mathrm{o}}\middle|h_{m-1,i}^{\mathrm{o}}\right)v_{m-1,i}^{\mathrm{o}}\left(h_{m-1,i}^{\mathrm{o}}\middle|\boldsymbol{x}_{m-1}^{\mathrm{f}}\right),$$

$$\tau'_{mi}\left(h_{mi}^{\mathrm{o}}\middle|\boldsymbol{x}_{m}^{\mathrm{f}},\boldsymbol{x}_{m-1}^{\mathrm{f}}\right)=l_{mi}^{\mathrm{o}}\max_{h_{m-1,i}^{\mathrm{o}}}\left[\tau_{mi}\left(h_{mi}^{\mathrm{o}}\middle|h_{m-1,i}^{\mathrm{o}},\boldsymbol{x}_{m-1}^{\mathrm{f}}\right)\right],$$

where $\tau_{mi}\left(h_{mi}^{\mathrm{o}}\middle|h_{m-1,i}^{\mathrm{o}},\boldsymbol{x}_{m-1}^{\mathrm{f}}\right)$ is a matrix each element of that indicates the probability to observe the transition from $h_{m-1,i}^{\mathrm{o}}$ to h_{mi}^{o} when $h_{m-1,i}^{\mathrm{o}}$ and $\boldsymbol{x}_{m-1}^{\mathrm{f}}$ are given, while $\tau'_{mi}\left(h_{mi}^{\mathrm{o}}\middle|\boldsymbol{x}_{m}^{\mathrm{f}},\boldsymbol{x}_{m-1}^{\mathrm{f}}\right)$ indicates the probabilities to observe h_{mi}^{o} when the most probable transition from $h_{m-1,i}^{\mathrm{o}}$ to h_{mi}^{o} occurred with given $\boldsymbol{x}_{m}^{\mathrm{f}}$ and $\boldsymbol{x}_{m-1}^{\mathrm{f}}$.

Then, calculate the Viterbi scores and the Viterbi paths for founders as

$$v_{m}^{\mathrm{f}}\left(\boldsymbol{x}_{m}^{\mathrm{f}}\middle|\boldsymbol{x}_{m-1}^{\mathrm{f}}\right)=P\left(\boldsymbol{x}_{m-1}^{\mathrm{f}}\right)l_{m-1}^{\mathrm{f}}\prod_{i}\sum_{h_{mi}^{\mathrm{o}}}\tau'_{mi}\left(h_{mi}^{\mathrm{o}}\middle|\boldsymbol{x}_{m}^{\mathrm{f}},\boldsymbol{x}_{m-1}^{\mathrm{f}}\right),$$

$$\psi_{m}^{\mathrm{f}}\left(\boldsymbol{x}_{m}^{\mathrm{f}}\right)=\operatorname*{argmax}_{\boldsymbol{x}_{m-1}^{\mathrm{f}}}\left[v_{m}^{\mathrm{f}}\left(\boldsymbol{x}_{m}^{\mathrm{f}}\middle|\boldsymbol{x}_{m-1}^{\mathrm{f}}\right)\right].$$

Using $\psi_{m}^{\mathrm{f}}\left(\boldsymbol{x}_{m}^{\mathrm{f}}\right)$ that stores the most probable combination of founder genotypes at marker m-1 when $\boldsymbol{x}_{m}^{\mathrm{f}}$ is given, we can get the Viterbi scores and the Viterbi paths for offspring as following.

$$\psi_{mi}^{\mathrm{o}}\left(h_{mi}^{\mathrm{o}}\middle|\boldsymbol{x}_{m}^{\mathrm{f}}\right)=\arg\max_{h_{m-1,i}^{\mathrm{o}}}\left[\tau_{mi}\left(h_{mi}^{\mathrm{o}}\middle|h_{m-1,i}^{\mathrm{o}},\boldsymbol{x}_{m-1}^{\mathrm{f}}=\psi_{m}^{\mathrm{f}}\left(\boldsymbol{x}_{m}^{\mathrm{f}}\right)\right)\right],$$

$$v_{mi}^{\mathrm{o}}\left(h_{mi}^{\mathrm{o}}\middle|\boldsymbol{x}_{m}^{\mathrm{f}}\right)=\tau'_{mi}\left(h_{mi}^{\mathrm{o}}\middle|\boldsymbol{x}_{m}^{\mathrm{f}},\boldsymbol{x}_{m-1}^{\mathrm{f}}=\psi_{m}^{\mathrm{f}}\left(\boldsymbol{x}_{m}^{\mathrm{f}}\right)\right).$$

Termination at m = M:

Finally, we obtain the probability to observe $\boldsymbol{x}_{M}^{\mathrm{f}}$ and the most probable combination of founder genotypes at marker M via the following calculation.

$$v'^{\mathrm{f}}_{M}\left(\boldsymbol{x}_{M}^{\mathrm{f}}\right)=P\left(\boldsymbol{x}_{M}^{\mathrm{f}}\right)l_{M}^{\mathrm{f}}\prod_{i}\sum_{h_{mi}^{\mathrm{o}}}v_{Mi}^{\mathrm{o}}\left(h_{Mi}^{\mathrm{o}}\middle|\boldsymbol{x}_{M}^{\mathrm{f}}\right),$$

$$\hat{\boldsymbol{x}}_M^{\mathrm{f}} = \underset{\boldsymbol{x}_M^{\mathrm{f}}}{\operatorname{argmax}}\left[v'^{\mathrm{f}}_M\left(\boldsymbol{x}_M^{\mathrm{f}}\right)\right].$$

Once $\hat{\boldsymbol{x}}_M^{\mathrm{f}}$ was determined, we can also get the most probable offspring haplotypes at marker M:

$$\hat{h}_{Mi}^{\mathrm{o}} = \underset{h_{Mi}^{\mathrm{o}}}{\operatorname{argmax}}\left[v_{Mi}^{\mathrm{o}}\left(h_{Mi}^{\mathrm{o}} \middle| \hat{\boldsymbol{x}}_M^{\mathrm{f}}\right)\right].$$

Backtracking from M-1 to 1:

To retrieve the most probable sequences of founder genotypes and offspring haplotypes, we trace back the Viterbi paths as shown below.

$$\hat{\boldsymbol{x}}_m^{\mathrm{f}} = \psi_{m+1}^{\mathrm{f}}\left(\hat{\boldsymbol{x}}_{m+1}^{\mathrm{f}}\right),$$

$$\hat{h}_{mi}^{\mathrm{o}} = \psi_{m+1,i}^{\mathrm{o}}\left(\hat{h}_{m+1,i}^{\mathrm{o}} \middle| \hat{\boldsymbol{x}}_{m+1}^{\mathrm{f}}\right).$$

If multiple arguments have the same maximum score, one of them 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 $\hat{\boldsymbol{x}}_M^{\mathrm{f}}$ 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 $\left(m < \frac{M}{2}\right)$ from the reversed direction round with the last half from the forward direction round.

The estimated genotype $\hat{x}_{mi}^{\mathrm{o}}$ at each marker in each sample can now be obtained based on $\hat{h}_{mi}^{\mathrm{o}}$ and $\hat{\boldsymbol{x}}_m^{\mathrm{f}}$. To evaluate the reliability of the estimated genotypes, the marginal

probability is also calculated:

$$P\left(x_{mi}^{\mathrm{o}} \middle| \boldsymbol{Y}^{\mathrm{o}}, \widehat{\boldsymbol{X}}^{\mathrm{f}}\right) = \sum_{\widehat{h}_{mi}^{o}} P\left(x_{mi}^{\mathrm{o}} \middle| h_{mi}^{\mathrm{o}}, \widehat{\boldsymbol{X}}^{\mathrm{f}}\right) P\left(h_{mi}^{\mathrm{o}} \middle| \boldsymbol{Y}^{\mathrm{o}}, \widehat{\boldsymbol{X}}^{\mathrm{f}}\right),$$

where $P\left(h_{mi}^{\mathrm{o}} \middle| \boldsymbol{Y}^{\mathrm{o}}, \widehat{\boldsymbol{X}}^{\mathrm{f}}\right)$ can be obtained using the forward-backward algorithm[57]. The estimated genotype $\hat{x}_{mi}^{\mathrm{o}}$ is set as missing if $P\left(\hat{x}_{mi}^{\mathrm{o}} \middle| \boldsymbol{Y}^{\mathrm{o}}, \widehat{\boldsymbol{X}}^{\mathrm{f}}\right)$ is smaller than a user specified threshold P_{call}, with 0.9 as the default.

Estimation of allele read bias and mismapping rate

GBScleanR requires four parameters E^{d}, e^{seq}, e_{m}^{map}, 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 as input parameters, while the mismap rates e_{m}^{map} and the allele read biases w_{m} can be estimated via the iterative parameter optimization (IPO). 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, $e_{m}^{\mathrm{map}} = \left(e^{\mathrm{ref}}, e^{\mathrm{alt}}\right)$, and w_{m} are initialized 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_{mi}^{ref} and y_{mi}^{alt} be the number of reference reads and alternative reads at marker *m* in sample *i* including both offspring and founders. The mismapping rate at marker *m* can be obtained using:

$$e^{\mathrm{ref}} = \frac{\sum_i \left[\delta\left(y_{mi}^{\mathrm{alt}} > 0\right)\delta(x_{mi}^{\mathrm{o}} = 0)\right]}{\sum_i [\delta(x_{mi}^{\mathrm{o}} = 0)]},$$

$$e^{alt} = \frac{\sum_i \left[\delta\left(y_{mi}^{\mathrm{ref}} > 0\right)\delta(x_{mi}^{\mathrm{o}} = 2)\right]}{\sum_i [\delta(x_{mi}^{\mathrm{o}} = 2)]},$$

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 following:

$$w_m = \frac{E_m^{\mathrm{ref}}}{E_m^{\mathrm{ref}} + E_m^{\mathrm{alt}}},$$

$$E_m^{\mathrm{ref}} = \frac{r_m^{\mathrm{ref}}(0) + r_m^{\mathrm{ref}}(1)}{n_m(0) + n_m(1)},$$

$$E_m^{\mathrm{alt}} = \frac{r_m^{\mathrm{alt}}(2) + r_m^{\mathrm{alt}}(1)}{n_m(2) + n_m(1)},$$

$$r_m^{\mathrm{x}}(g) = \sum_i [y_{mi}^{x}\delta(x_{mi}^{\mathrm{o}} = g)], \mathrm{x} \in \{\mathrm{ref}, \mathrm{alt}\},$$

$$n_m(g) = 2^{\delta(g \neq 1)} \sum_i [\delta(x_{mi}^{\mathrm{o}} = g)],$$

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

751 prepared and sequenced together. To confirm if the given data meets this assumption, our
752 algorithm first calculates $\boldsymbol{w} = \{w_m\}_{m=1\ldots M}$ with $g = 1$ and $g = \{0, 2\}$ separately,
753 and then evaluates the correlation between $\boldsymbol{w}$ obtained in both calculations. If the
754 correlation coefficient is greater than or equal to 0.7, the equation above is employed;
755 otherwise, $\boldsymbol{w}$ used with g = 1, although all of the datasets used in this paper meet the
756 assumption of bias uniformity, even the real dataset.
757
758 *Preparation of real data*
759 The genotype data of an F_2 population that was derived from a cross between *O. sativa*
760 and *O. longistaminata* were used in this study. The data potentially include a large number
761 of error prone markers with mismapping and allele read biases, as described in our
762 previous paper[58]. In brief, the F_2 population was produced via the self-pollination of F_1
763 plants that were derived from a cross between *O. sativa* ssp. *japonica* cv. Nipponbare and
764 *O. longistaminata* acc. IRGC110404. GBS was performed with a *Kpn*I-*Msp*I restriction
765 enzyme pair using MiSeq with 75 bp paired-end sequencing. Sequencing runs with nine
766 96-multiplex libraries generated an average $134{,}447 \pm 50{,}788$ reads per sample. We
767 obtained a total of 2,539,459 and 3,481,218 reads for *O. sativa* and *O. longistaminata*
768 from the independent runs, respectively, which should be sufficient for fairly accurate

769 determination of the founder genotypes. The obtained reads were then processed via the
770 TASSEL-GBS pipeline v2 by following the instructions in the manual with the default
771 parameters (https://bitbucket.org/tasseladmin/tassel-5-
772 source/wiki/Tassel5GBSv2Pipeline)[49]. The obtained SNP markers were then filtered to
773 retain only those markers that were homozygous in each founder but biallelic between
774 founders. Only the first SNP was selected if multiple SNPs were present within a 75 bp
775 stretch. Considering the distant cross that was used to obtain the F_2 population, our
776 genotype data might include mismap-prone markers due to the repetitive sequences in the
777 two founder genomes. The mismapping patterns can differ between samples which have
778 different descendent genome compositions. Therefore, to filter out the erroneous
779 genotype calls that were caused by mismapping reads from repetitive sequences, both the
780 reference and the alternative allele read counts were set to 0 for genotype calls where
781 either the reference or the alternative allele reads were higher than the 90th percentile for
782 each genotype call in each sample, respectively. The resulting genotype data had 5035
783 SNP markers for 814 F_2 individuals with an average read depth of 0.85x.
784
785 *Generation of simulation data*
786 Genotype datasets were simulated for three scenarios: a 2-way F_2 population from inbred

787 founders (homoP2_F2), a 2-way F_2 population from outbred founders (hetP2_F2), and 8-
788 way RILs from inbred founders (homoP8_RIL). For each scenario, we simulated
789 populations that have 10, 100, and 1000 individuals, respectively. To generate the
790 descendent haplotype and genotype data for each dataset in each scenario, the breeding
791 pedigree was first simulated via the following. The homoP2_F2 scenario assumes that the
792 two inbred founders have homozygotes at all markers and are biallelic between founders,
793 the founders are crossed and then self-pollinate, as usually conducted in the genetic
794 analysis of self-fertilizing plants. The hetP2_F2 scenario assumes two outbred founders
795 with independently random genotypes at any marker with the minor allele observed in at
796 least one of the four possible haplotypes. These outbred founders are subjected to mating
797 to produce 1000 progenies and then 100 sibling combinations are used to produce 10
798 progenies in each. The last generation of hetP2_F2 has 1000 offspring in total. This
799 scenario assumes the case of genetic analyses for non-self-fertilizing organisms. The
800 pedigree of homoP8_RIL consists of three pairing generations and five self-fertilized
801 generations. In the pairing stage, four pairs are selected from the eight founders with no
802 duplication, and one descendant is created from each pair. The four descendants are then
803 further paired to produce 100 progenies from each of the two pairs. At this step, we have
804 two sibling families with 100 individuals in each. At the last pairing, 50 pairs are produced

805 by selecting one individual per sibling family to produce pairs from each of which 50
806 progenies are generated. The resultant 2500 individuals are then self-fertilized five times
807 so that one offspring is produced per individual in order to maintain the population size.
808 This kind of pedigree is usually found in populations that have been subjected to genetic
809 survey and breeding and known as Multiparent Advanced Generation Inter-Cross
810 (MAGIC) population. A given number of offspring is sampled from the last generation to
811 prepare each dataset.

812 Descendent haplotype data from each population was generated by introducing
813 crossovers into the gametes that have descended from the parent(s) of each progeny at
814 each mating step. Only one pair of 50 Mb chromosomes as diploid genome was simulated
815 for all datasets and the expected genetic distance per mega base pairs was set to 0.04.
816 Markers were located randomly throughout the chromosome. The occurrence of
817 crossovers between adjacent markers in a gamete follows Poisson distribution with a
818 mean value that is equal to the genetic distance of the given pair of adjacent markers. The
819 genetic distances between the markers were calculated by multiplying the physical
820 distances by 0.04×10^{-6}. If an odd number of crossovers was assigned to a marker pair,
821 gamete switching descendent haplotypes were generated, while the assigned crossover
822 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. The expected offspring read depths were set at 0.1x, 0.25x, 0.5x, 0.75x, 1x, 2x, 3x, 10x, and 20x, respectively. The founder read depth was set at 5x for all datasets, because usually founders are sequenced relatively deep even in the low coverage GBS. For each of the three scenarios, the read count data was simulated by mimicking the allele read biases observed in the real data. 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, $\mathrm{Exp}(\lambda)$. The probability of a read at marker *m* can be obtained by the following:

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

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 $(1 - p^{\mathrm{ref}})$, respectively, where p^{ref} represents the

841 probability that the read is a reference allele. The probability p^{ref} takes different values

842 depending on the genotype at marker m in sample i, x_{mi}. When the genotype is the

843 homozygous reference, $p^{\mathrm{ref}} = (1 - e^{\mathrm{seq}})$, where e^{seq} represents a sequencing error.

844 Similarly, $p^{\mathrm{ref}} = 0.5$ and $p^{\mathrm{ref}} = e^{\mathrm{seq}}$ for heterozygous and homozygous alternatives.

845 To incorporate allele read bias, the genotype-dependent read observation probability

846 $P(z|w_m, x_{mi}), z \in \{\mathrm{ref}, \mathrm{alt}\}$ was introduced, which takes the values shown in Table 4.

847 The joint probability of having a read for allele z at marker m in sample i can then be

848 calculated:

849 $$P(z, m, i|w_m, x_{mi}) = P(z|w_m, x_{mi})P(m)P(i), P(i) = \frac{1}{N},$$

850 where N is the number of samples. Mismapping error is introduced by replacing the

851 genotypes with the probability $P\left(x'_{mi}\middle|x_{mi}, e_m^{\mathrm{map}} = \left(e^{\mathrm{ref}}, e^{\mathrm{alt}}\right)\right)$, taking the values

852 listed in Table 3. The number of total reads is calculated by multiplying the number of

853 samples by the expected read depth and the number of markers. We then allocate those

854 reads based on the joint probability $P(z, m, i|w_m, x'_{mi})$, which is not conditional on the

855 true genotype x_{mi}. The read count data for the founders and the offspring are simulated

856 separately with given expected read depths for each dataset.

857 The simulation datasets were generated by mimicking only the real data of chromosome

858 7, which showed relatively severe bias (Supplemental Figure 1). Each dataset was

859 therefore simulated to consist of 620 markers on one chromosome with physical lengths
860 of 50 Mb and genetic lengths of 2 Morgans. $P(m)$ was also obtained from the read data
861 by calculating the total number of reads per marker y_m^T by summing up the read counts
862 for each marker. The values w_m were obtained by calculating $y_m^{\mathrm{ref}}/(y_m^{\mathrm{ref}} + y_m^{\mathrm{alt}})$, where
863 y_m^{ref} and y_m^{alt} denote the total numbers of reference and alternative reads per marker
864 observed in the real data, respectively. The mismapping rate was randomly assigned to
865 markers by sampling the probabilities from an exponential distribution with the mean at
866 0.05, since the mismapping pattern in the real data is unknown. The sequencing error rate
867 e^{seq} was set to 0.0025. Since typical genotyping pipeline filter out markers at which no
868 read was observed in either of founders, we allocated at least one read to all markers of
869 founders and then allocated the remaining reads by following the join probability
870 ("nonzero" datasets). In addition to them, we also simulated datasets in which all reads
871 were allocated by following the join probability and, as a result, founders have no read at
872 some markers and missing genotypes ("allowzero" datasets).
873
874 *Input parameter settings*
875 GBScleanR was tested and compared with LB-impute and magicImpute[26,27]. GBScleanR
876 includes some arguments that can be used to tweak this tool: the expected genetic distance

877 per mega base pairs E^d, sequencing error rate e^{seq}, threshold for the probability of calling
878 estimated genotype P_{call}, and the number of iterations required for genotype estimation.
879 To decide the default values for these arguments, we tested the algorithm using the
880 simulation dataset with 100 individuals and a 3x offspring read depth in the "nonzero"
881 dataset of the homoP2_F2 scenario under various settings. As shown in Supplemental
882 Figure S7, the tests indicated that P_{call} have relatively large impacts on the genotype
883 estimation accuracy, while the differences associated with the number of iterations (if
884 more than one), E^d, and e^{seq} were negligible. The values of E^d = 0.04 and e^{seq} =
885 0.0025 that were used in the data simulation seem to be good choices when set for
886 genotype estimation. Increasing P_{call} from 0.8 to 0.95 improved the correct call rate in an
887 almost proportional manner, while the miscall rate worsened at the same time. Therefore,
888 we set P_{call} at 0.9 for the algorithm evaluation as used in the default setting for
889 magicImpute. In the case of iteration number, we found that a larger number leads to
890 better correct call and miscall rates, which reached a plateau at four or more iterations.
891 Thus, we decided to set the number of iterations to four. To evaluate GBScleanR, we
892 tested the algorithm with and without the iterative parameters optimization (IPO) w_m and
893 e_m^{map}.
894 For fair comparison among the three tools, similar input parameters were used if LB-

895 Impute and magicImpute have corresponding parameters to those given in GBScleanR.
896 In the case of LB-impute, the "impute" option was set to the "-method" argument and the
897 "-offspringimpute" flag was used to run it in the offspring impute mode. A vector of
898 strings stating the names of the founder samples was specified as "-parents". “-recombdist”
899 and “-readerr” were set at 12500000 and 0.0025, which correspond to E_d and e^{seq}. While
900 the parameter related to read mismapping is estimated by IPO and not tunable in
901 GBScleanR, LB-Impute requires to set the parameter “-genotypeerr” that is related to e^{map}
902 of GBScleanR. We set 0.005 to “-genotypeerr” as magicImpute also requires similar
903 parameters and has 0.005 as default. The arguments in magicImpute include the
904 mandatory arguments "model" and "popdesign" in addition to the genotype data in the
905 "magicsnp" format. The simulated data and real data were converted to the magicsnp
906 format. Marker positions in centimorgan, which are required for magicImpute, were
907 calculated by multiplying the physical positions of the markers by the expected genetic
908 distance per mega base of 0.04, which was used to generate the simulation data. The
909 "jointModel" option was set to the "model" argument for all datasets. The "popdesign"
910 argument that affects on the state transition-rates was specified based on the pedigree in
911 each scenario as follows. The “popdesign” was set to ("Pairing", "Selfing") for the
912 scenario homoP2_F2, while ("Pairing", "Pairing", "Pairing", "Selfing", "Selfing",

913 "Selfing", "Selfing", "Selfing") was specified for homoP8_RIL. Although the popdesign
914 for hetP2_F2 was supposed to be ("Pairing", "Sibling") based on the pedigree and the
915 instructions in the manual, magicImpute did not accept it and returned an error message
916 saying "invalid popdesign". Therefore, pedigree information files were created to indicate
917 population design for all datasets of the hetP2_F2 scenario and provided to execute
918 magicImpute as “popdesign”. As a note, we found that the execution of magicImpute with
919 a pedigree information file takes longer calculation than that with simple “popdesign”
920 specification by a string vector. Additionally, magicImpute has “minPhredQualScore”,
921 “imputingThreshold”, and “detectingThreshold” arguments. The former corresponds to
922 e^{seq} and was set at 26.0206 that is equal to e^{seq} = 0.0025. The latter two specify the
923 parameter similar with P_{call} of GBScleanR and both of them was set at 0.9. Both of
924 “founderAllelicError” and “offspringAllelicError” that corresponds to e^{map} of GBScleanR
925 and “-genotypeerr” of LB-Impute were set to 0.005. The estimated genotype data
926 generated by magicImpute includes incomplete genotype estimations represented by 1N,
927 N1, 2N, and N2, where N points to an unknown allele and the numbers 1 and 2 indicate
928 reference and alternative alleles, respectively. Although these incomplete estimations
929 provide partial information about the genotype at the markers, we treat them as
930 completely missing because downsream genetic analyses generally cannot handle such

931 incomplete information.

932

933 *Evaluation of algorithms*

934 The genotype estimation accuracy of each algorithm for the simulated datasets was
935 evaluated by calculating the proportions of correctly estimated genotypes (correct call
936 rate), wrongly estimated genotypes (miscall rate), and missing genotype calls (missing
937 call rate) for each offspring in the datasets. The means and standard deviations of these
938 values were then obtained for each dataset processed by each algorithm. The resultant
939 scores are shown in Supplemental Data 1. In the case of the real data, we first selected
940 relatively reliable genotype calls supported by more than 6 reads at markers that have less
941 than 20% of missing rate and more than 40% of minor allele frequency. Those selected
942 genotype calls were then set to missing by removing all read counts observed for them.
943 This masked real data was subjected to genotype estimation using GBScleanR, LB-
944 Impute, and magicImpute. The resultant estimated genotype data was evaluated based on
945 the concordance rate that is the proportion of estimated genotype calls matching with the
946 raw genotype calls at the masked genotype calls. Furthermore, we also calculated the
947 averages of missing rate, the number of recombinations, and segregation distortion level
948 in each chromosome. The segregation distortion level at each marker was obtained by

949 dividing the chi-square value, which was calculated based on the assumption that the
950 genotype ratio to be 0.25, 0.5, and 0.25 for three possible genotypes, by the number of
951 non-missing genotype calls. The segregation distortion level of each chromosome was
952 then obtained by summing up the segregation distortion levels of all markers in each
953 chromosome. To identify recombination breakpoints of genome segments in the estimated
954 genotype data, we searched the markers that show genotypes switched between adjacent
955 non-missing markers. The middle points of those markers were considered as
956 recombination break points.
957

958 **Data availability**

959 The stable version of GBScleanR is available on Bioconductor
960 (https://bioconductor.org/packages/GBScleanR/). The latest developmental version is
961 available on GitHub (https://github.com/tomoyukif/GBScleanR). This package and its
962 source codes are distributed under the license GNU General Public License version 3.
963 The vcf file of the real data and R scripts for the data analyses performed in this study are
964 also available on GitHub (https://github.com/tomoyukif/GBSR_SupFiles). Further
965 description of the sequencing reads for the real data can be found in Furuta *et al.*, 2017[58].
966

Acknowledgements

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.].

Contributions

T.F. conceived and designed the study. T.F. also developed the R package, conducted bioinformatics analysis, and drafted the manuscript. M.A. and T.Y. supervised the study. All authors read and approved the final version of the manuscript.

References

1. Goodwin, S., McPherson, J. D. & McCombie, W. R. Coming of age: Ten years of next-generation sequencing technologies. *Nature Reviews Genetics* vol. 17 333–351 Preprint at https://doi.org/10.1038/nrg.2016.49 (2016).
2. Unamba, C. I. N., Nag, A. & Sharma, R. K. Next generation sequencing technologies: The doorway to the unexplored genomics of non-model plants. *Frontiers in Plant Science* vol. 6 1074 Preprint at https://doi.org/10.3389/fpls.2015.01074 (2015).
3. Davey, J. W. *et al.* Genome-wide genetic marker discovery and genotyping using next-generation sequencing. *Nature Reviews Genetics* vol. 12 499–510 Preprint at https://doi.org/10.1038/nrg3012 (2011).
4. Torkamaneh, D., Boyle, B. & Belzile, F. Efficient genome-wide genotyping strategies and data integration in crop plants. *Theoretical and Applied Genetics* vol. 131 499–511 Preprint at https://doi.org/10.1007/s00122-018-3056-z (2018).
5. Bhat, J. A. *et al.* Genomic selection in the era of next generation sequencing for complex traits in plant breeding. *Frontiers in Genetics* vol. 7 221 Preprint at https://doi.org/10.3389/fgene.2016.00221 (2016).
6. Scheben, A., Batley, J. & Edwards, D. Genotyping-by-sequencing approaches to characterize crop genomes: choosing the right tool for the right application. *Plant Biotechnol J* **15**, 149–161 (2017).
7. Baird, N. A. *et al.* Rapid SNP Discovery and Genetic Mapping Using Sequenced RAD Markers. *PLoS One* **3**, e3376 (2008).
8. Peterson, B. K., Weber, J. N., Kay, E. H., Fisher, H. S. & Hoekstra, H. E. Double Digest RADseq: An Inexpensive Method for De Novo SNP Discovery and Genotyping in Model and Non-Model Species. *PLoS One* **7**, e37135 (2012).
9. Fu, Z. *et al.* Using NextRAD sequencing to infer movement of herbivores among host plants. *PLoS One* **12**, e0177742 (2017).
10. Toonen, R. J. *et al.* ezRAD: A simplified method for genomic genotyping in non-model organisms. *PeerJ* **2013**, e203 (2013).
11. Wang, S., Meyer, E., Mckay, J. K. & Matz, M. v. 2b-RAD: A simple and flexible method for genome-wide genotyping. *Nat Methods* **9**, 808–810 (2012).
12. Ali, O. A. *et al.* Rad capture (Rapture): Flexible and efficient sequence-based genotyping. *Genetics* **202**, 389–400 (2016).
13. Elshire, R. J. *et al.* A Robust, Simple Genotyping-by-Sequencing (GBS) Approach for High Diversity Species. *PLoS One* **6**, e19379 (2011).
14. Poland, J. A., Brown, P. J., Sorrells, M. E. & Jannink, J.-L. Development of High-

1015 Density Genetic Maps for Barley and Wheat Using a Novel Two-Enzyme Genotyping-
1016 by-Sequencing Approach. *PLoS One* **7**, e32253 (2012).
1017 15. de Donato, M., Peters, S. O., Mitchell, S. E., Hussain, T. & Imumorin, I. G.
1018 Genotyping-by-Sequencing (GBS): A Novel, Efficient and Cost-Effective Genotyping
1019 Method for Cattle Using Next-Generation Sequencing. *PLoS One* **8**, e62137 (2013).
1020 16. Morris, G. P. *et al.* Population genomic and genome-wide association studies of
1021 agroclimatic traits in sorghum. *Proc Natl Acad Sci U S A* **110**, 453–458 (2013).
1022 17. Kujur, A. *et al.* Ultra-high density intra-specific genetic linkage maps accelerate
1023 identification of functionally relevant molecular tags governing important agronomic
1024 traits in chickpea. *Sci Rep* **5**, (2015).
1025 18. Spindel, J. *et al.* Genomic Selection and Association Mapping in Rice (Oryza sativa):
1026 Effect of Trait Genetic Architecture, Training Population Composition, Marker
1027 Number and Statistical Model on Accuracy of Rice Genomic Selection in Elite,
1028 Tropical Rice Breeding Lines. *PLoS Genet* **11**, e1004982 (2015).
1029 19. Parker, C. C. *et al.* Genome-wide association study of behavioral, physiological and
1030 gene expression traits in outbred CFW mice. *Nat Genet* **48**, 919–926 (2016).
1031 20. Litt, M. & Luty, J. A. A hypervariable microsatellite revealed by in vitro amplification
1032 of a dinucleotide repeat within the cardiac muscle actin gene. *Am J Hum Genet* **44**,
1033 397–401 (1989).
1034 21. Syvanen, A. C. Toward genome-wide snp genotyping. *Nat Genet* **37**, S5–S10 (2005).
1035 22. Browning, S. R. & Browning, B. L. Rapid and accurate haplotype phasing and missing-
1036 data inference for whole-genome association studies by use of localized haplotype
1037 clustering. *Am J Hum Genet* **81**, 1084–1097 (2007).
1038 23. Huang, X. *et al.* High-throughput genotyping by whole-genome resequencing.
1039 *Genome Res* **19**, 1068–1076 (2009).
1040 24. Swarts, K. *et al.* Novel Methods to Optimize Genotypic Imputation for Low-Coverage,
1041 Next-Generation Sequence Data in Crop Plants. *Plant Genome* **7**,
1042 plantgenome2014.05.0023 (2014).
1043 25. Rowan, B. A., Patel, V., Weigel, D. & Schneeberger, K. Rapid and inexpensive whole-
1044 genome genotyping-by-sequencing for crossover localization and fine-scale genetic
1045 mapping. *G3: Genes, Genomes, Genetics* **5**, 385–398 (2015).
1046 26. Fragoso, C. A., Heffelfinger, C., Zhao, H. & Dellaporta, S. L. Imputing genotypes in
1047 biallelic populations from low-coverage sequence data. *Genetics* **202**, 487–495 (2016).
1048 27. Zheng, C., Boer, M. P. & van Eeuwijk, F. A. Accurate genotype imputation in
1049 multiparental populations from low-coverage sequence. *Genetics* **210**, 71–82 (2018).
1050 28. Eddy, S. R. What is a hidden Markov model? *Nature Biotechnology* vol. 22 1315–1316

1051 Preprint at https://doi.org/10.1038/nbt1004-1315 (2004).
1052 29. Davey, J. W. *et al.* Special features of RAD Sequencing data: Implications for
1053 genotyping. *Mol Ecol* **22**, 3151–3164 (2013).
1054 30. DaCosta, J. M. & Sorenson, M. D. Amplification Biases and Consistent Recovery of
1055 Loci in a Double-Digest RAD-seq Protocol. *PLoS One* **9**, e106713 (2014).
1056 31. Wijnker, E. *et al.* The genomic landscape of meiotic crossovers and gene conversions
1057 in Arabidopsis thaliana. *Elife* **2013**, (2013).
1058 32. Chen, M. *et al.* An Integrated Physical and Genetic Map of the Rice Genome. *Plant*
1059 *Cell* **14**, 537 (2002).
1060 33. Scheben, A., Batley, J. & Edwards, D. Genotyping-by-sequencing approaches to
1061 characterize crop genomes: choosing the right tool for the right application. *Plant*
1062 *Biotechnology Journal* vol. 15 149–161 Preprint at https://doi.org/10.1111/pbi.12645
1063 (2017).
1064 34. Kim, C. *et al.* Application of genotyping by sequencing technology to a variety of crop
1065 breeding programs. *Plant Science* vol. 242 14–22 Preprint at
1066 https://doi.org/10.1016/j.plantsci.2015.04.016 (2016).
1067 35. Gardner, K. M. *et al.* Fast and cost-effective genetic mapping in apple using next-
1068 generation sequencing. *G3: Genes, Genomes, Genetics* **4**, 1681–1687 (2014).
1069 36. Pootakham, W. *et al.* Genome-wide SNP discovery and identification of QTL
1070 associated with agronomic traits in oil palm using genotyping-by-sequencing (GBS).
1071 *Genomics* **105**, 288–295 (2015).
1072 37. Celik, I., Bodur, S., Frary, A. & Doganlar, S. Genome-wide SNP discovery and genetic
1073 linkage map construction in sunflower (Helianthus annuus L.) using a genotyping by
1074 sequencing (GBS) approach. *Molecular Breeding* **36**, 1–9 (2016).
1075 38. Boyles, R. E. *et al.* Quantitative Trait Loci Mapping of Agronomic and Yield Traits in
1076 Two Grain Sorghum Biparental Families. *Crop Sci* **57**, 2443–2456 (2017).
1077 39. Zheng, C. Modeling X-linked ancestral origins in multiparental populations. *G3:*
1078 *Genes, Genomes, Genetics* **5**, 777–801 (2015).
1079 40. Gerard, D., Ferrão, L. F. V., Garcia, A. A. F. & Stephens, M. Genotyping polyploids
1080 from messy sequencing data. *Genetics* **210**, 789–807 (2018).
1081 41. Zych, K., Gort, G., Maliepaard, C. A., Jansen, R. C. & Voorrips, R. E. FitTetra 2.0 -
1082 Improved genotype calling for tetraploids with multiple population and parental data
1083 support. *BMC Bioinformatics* **20**, 148 (2019).
1084 42. Clark, L. v., Lipka, A. E. & Sacks, E. J. polyRAD: Genotype calling with uncertainty
1085 from sequencing data in polyploids and diploids. *G3: Genes, Genomes, Genetics* **9**,
1086 663–673 (2019).

43. Rhoads, A. & Au, K. F. PacBio Sequencing and Its Applications. *Genomics, Proteomics and Bioinformatics* vol. 13 278–289 Preprint at https://doi.org/10.1016/j.gpb.2015.08.002 (2015).
44. Lu, H., Giordano, F. & Ning, Z. Oxford Nanopore MinION Sequencing and Genome Assembly. *Genomics, Proteomics and Bioinformatics* vol. 14 265–279 Preprint at https://doi.org/10.1016/j.gpb.2016.05.004 (2016).
45. Atwell, B. J., Wang, H. & Scafaro, A. P. Could abiotic stress tolerance in wild relatives of rice be used to improve Oryza sativa? *Plant Science* vols 215–216 48–58 Preprint at https://doi.org/10.1016/j.plantsci.2013.10.007 (2014).
46. Kole, C. *et al.* Application of genomics-assisted breeding for generation of climate resilient crops: Progress and prospects. *Frontiers in Plant Science* vol. 6 563 Preprint at https://doi.org/10.3389/fpls.2015.00563 (2015).
47. Brozynska, M., Furtado, A. & Henry, R. J. Genomics of crop wild relatives: Expanding the gene pool for crop improvement. *Plant Biotechnology Journal* vol. 14 1070–1085 Preprint at https://doi.org/10.1111/pbi.12454 (2016).
48. McKenna, A. *et al.* The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res* **20**, 1297–1303 (2010).
49. Glaubitz, J. C. *et al.* TASSEL-GBS: A High Capacity Genotyping by Sequencing Analysis Pipeline. *PLoS One* **9**, e90346 (2014).
50. Zheng, X. *et al.* A high-performance computing toolset for relatedness and principal component analysis of SNP data. *Bioinformatics* **28**, 3326–3328 (2012).
51. Li, H. & Barrett, J. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *BIOINFORMATICS* **27**, 2987–2993 (2011).
52. Zheng, X. *et al.* SeqArray—a storage-efficient high-performance data format for WGS variant calls. *Bioinformatics* **33**, 2251–2257 (2017).
53. Gogarten, S. M. *et al.* Genetic association testing using the GENESIS R/Bioconductor package. *Bioinformatics* **35**, 5346–5348 (2019).
54. Gogarten, S. M. *et al.* GWASTools: an R/Bioconductor package for quality control and analysis of genome-wide association studies. *Bioinformatics* **28**, 3329–3331 (2012).
55. Zheng, C., Boer, M. P. & van Eeuwijk, F. A. A general modeling framework for genome ancestral origins in multiparental populations. *Genetics* **198**, 87–101 (2014).
56. Zheng, C., Boer, M. P. & van Eeuwijk, F. A. Reconstruction of genome ancestry blocks in multiparental populations. *Genetics* **200**, 1073–1087 (2015).
57. Rabiner, L. R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE* **77**, 257–286 (1989).

58. Furuta, T., Ashikari, M., Jena, K. K., Doi, K. & Reuscher, S. Adapting genotyping-by-sequencing for rice F2 populations. *G3: Genes, Genomes, Genetics* **7**, 881–893 (2017).

Tables

Table 1 The number of genome segments

		Segment length	
		0-1 Mb	1-2 Mb
Total[a]	GBScleanR	261	668
	magicImpute	627	1716
	LB-Impute	2164	4166
Double crossovers[b]	GBScleanR	58	78
	magicImpute	235	558
	LB-Impute	948	1395

[a] The total number of genome segments that have a same genotype continuously with the length of 0-1 or 1-2 Mb.
[b] The number of genome segments that have a heterozygous genotype continuously with the length of 0-1 or 1-2 Mb, and flanked by homozygous segments of a same allele.

Table 2 Running time

Dataset	No. of samples	GBScleanR[a]	magicImpute[b]	LB-Impute
homoP2_F2	10	2 (1)	2	20
homoP2_F2	100	3 (1)	13	178
homoP2_F2	1000	15 (5)	97	1834
hetP2_F2	10	4 (2)	3 (10)	NA
hetP2_F2	100	18 (5)	20 (140)	NA
hetP2_F2	1000	143 (41)	165 (7447)	NA
homoP8_RIL	10	44 (12)	98	NA
homoP8_RIL	100	235 (62)	336	NA
homoP8_RIL	1000	2087 (535)	829	NA

Running times in seconds for each dataset are shown.

[a] The numbers in parenthesis indicate the running times of GBScleanR without the iterative parameter optimization.

[b] The numbers in parenthesis indicate the running times of magicImpute with pedigree files.

1129

Table 2 Probabilities associated with mismapping

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

1130

Table 4 Genotype-dependent read observation probabilities

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

1131

Figures

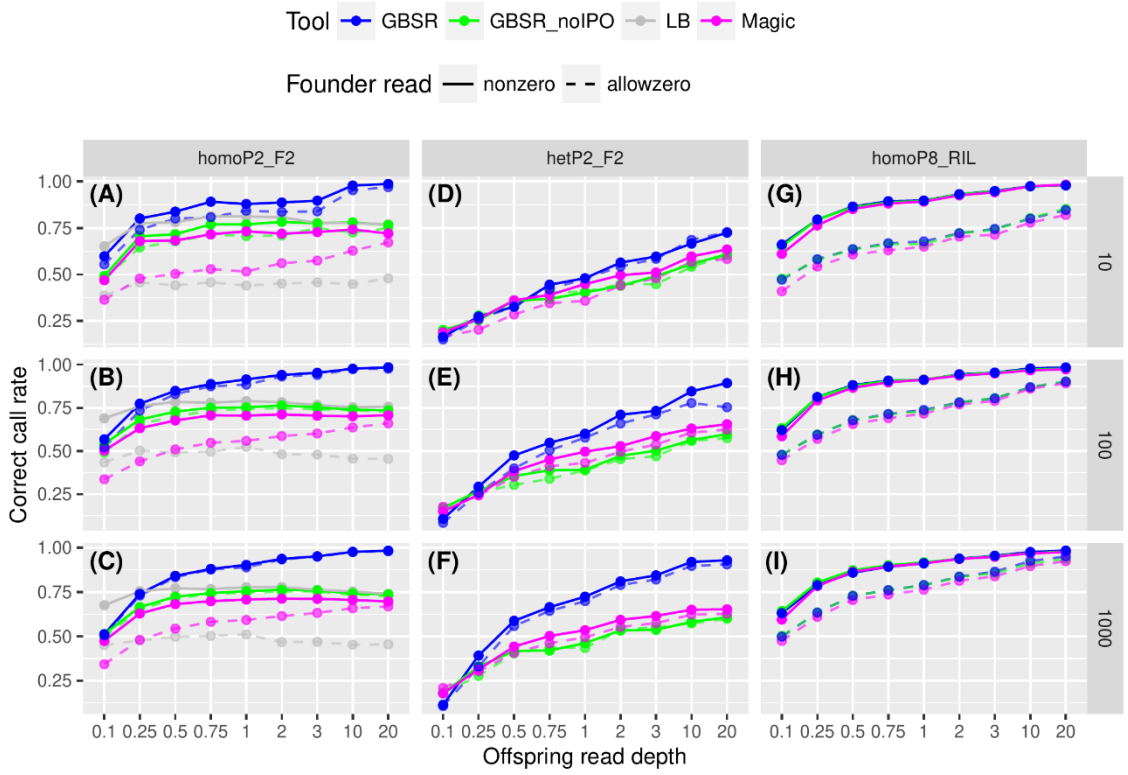

Figure 1 Offspring genotype estimation accuracies for the simulation data.

The plots show correct call rates for the datasets with given read depths for offspring (x axis) in the homoP2_F2 (A-C), hetP2_F2 (D-F), and homoP8_RIL (G-I) scenarios. The rows of the panels indicate the differences in the number of samples in the simulated data as shown in the strips on the right. Solid lines and dashed lines represent the results for the datasets with (allowzero) or without (nonzero) missing of founder reads. “GBSR” and “GBSR_noIPO” represent GBScleanR with and without IPO. “LB” and “Magic” indicate LB-Impute and magicImpute, respectively.

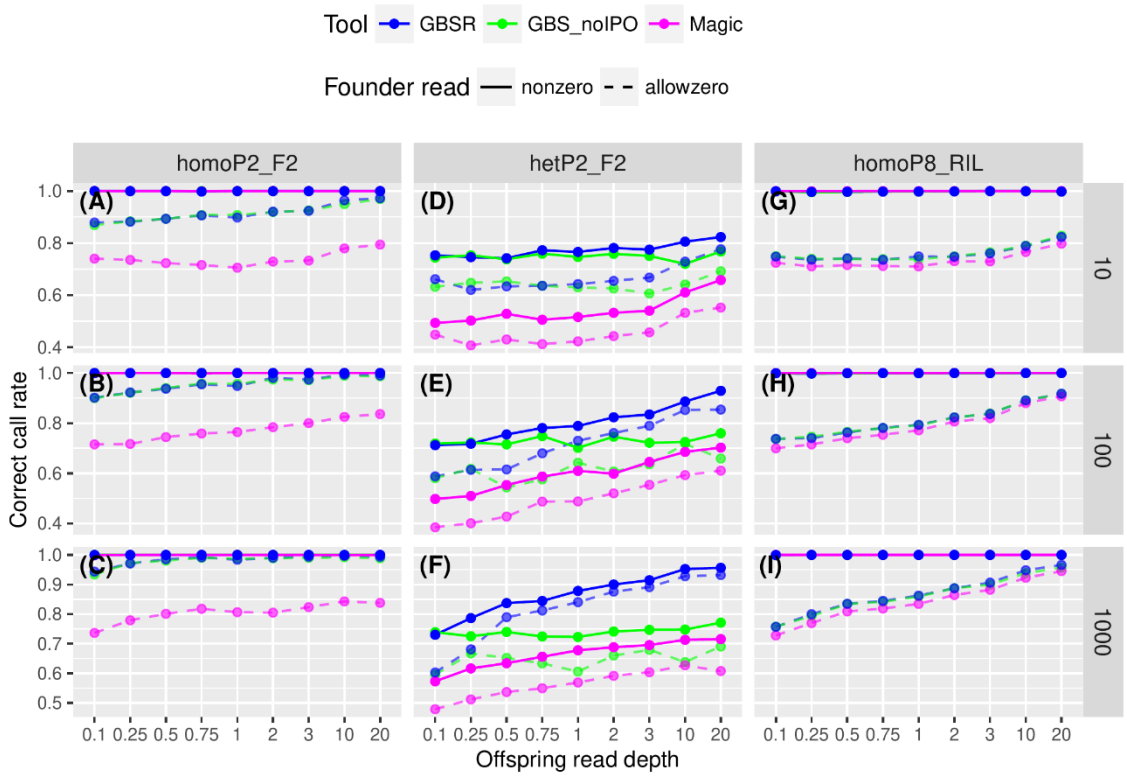

Figure 2 Founder genotype estimation accuracies for the simulation data.

The plots show correct call rates for the datasets with given read depths for offspring (x axis) in the homoP2_F2 (A-C), hetP2_F2 (D-F), and homoP8_RIL (G-I) scenarios. The rows of the panels indicate the differences in the number of samples in the simulated data as shown in the strips on the right. Solid lines and dashed lines represent the results for the datasets with (allowzero) or without (nonzero) missing of founder reads. “GBSR” and “GBSR_noIPO” represent GBScleanR with and without IPO. “LB” and “Magic” indicate LB-Impute and magicImpute, respectively.

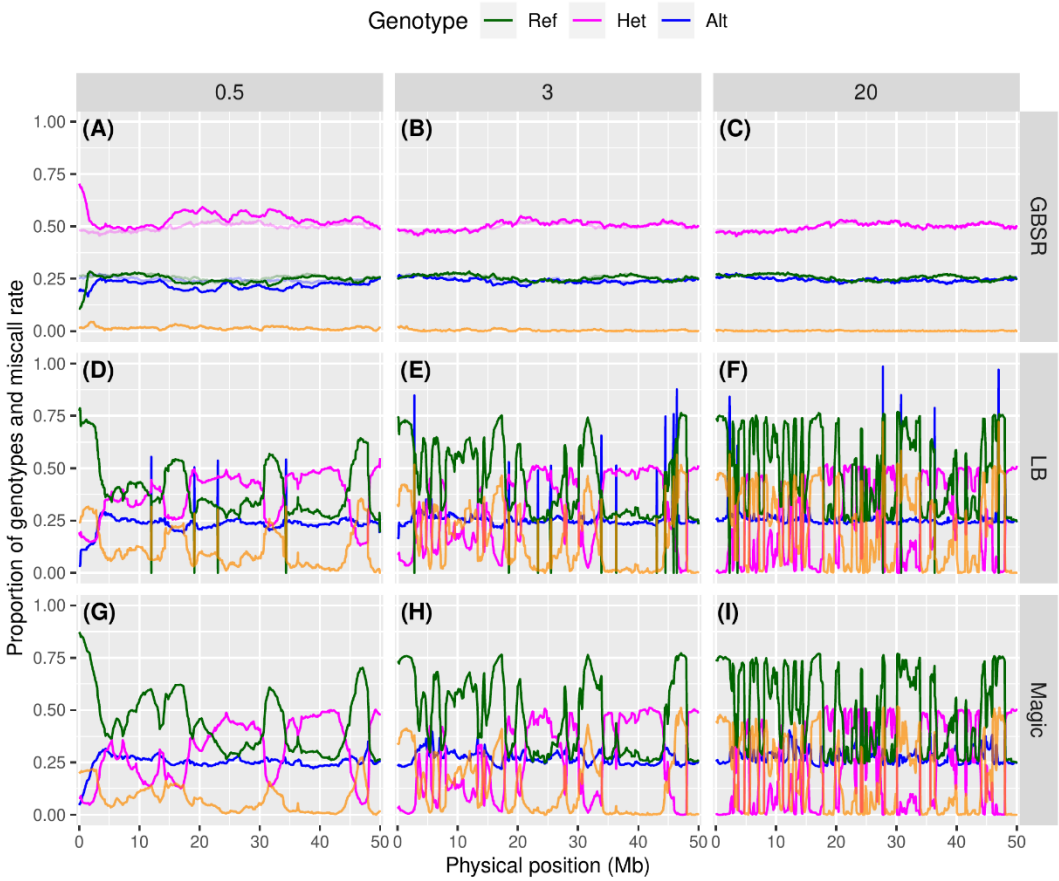

Figure 3 Genotype ratios at markers ordered along a simulated chromosome. The plots show genotype ratios calculated from the estimated genotype data for the homoP2_F2 dataset with 1000 offspring and 620 markers with offspring read depths of 0.5x, 3x, and 20x and without allowance of no read in founders. ”GBSR” (A-C), ”LB” (D-F), and ”Magic” (G-I) 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. True genotype ratios are indicated by transparent lines in the panels for GBScleanR (A-C).

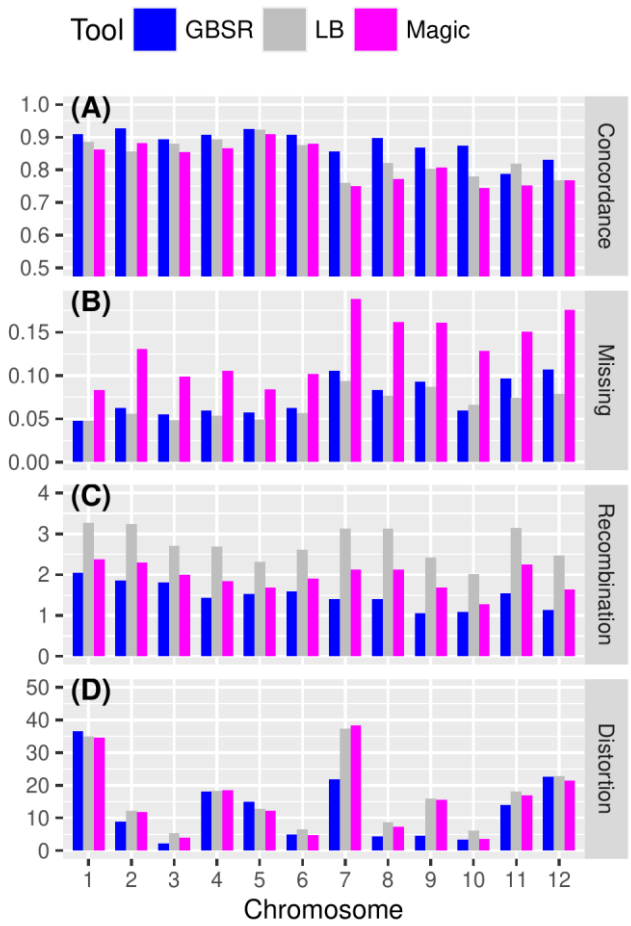

Figure 4 Evaluation of genotype estimation accuracy using the real data.

(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. Blue, gray, and magenta bars indicate the values of GBScleanR (GBSR), LB-Impute (LB), and magicImpute (Magic), respectively

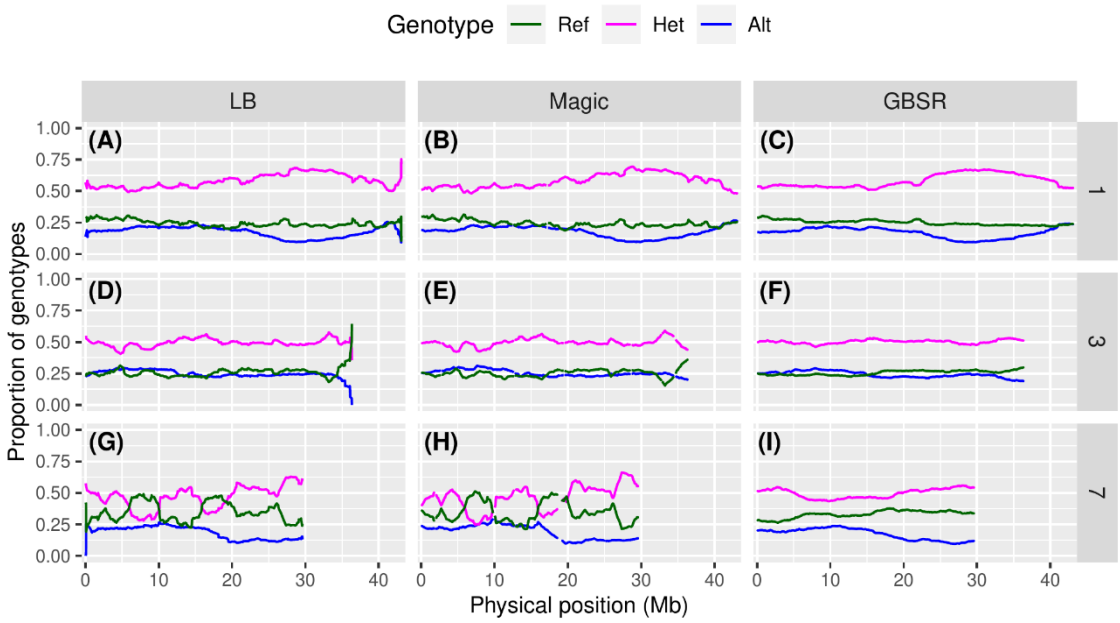

Figure 5 Genotype ratio of estimated offspring genotypes in the real data.

Genotype ratios at markers along chromosome 1 (A-C), 3 (D-F), and 7 (G-I) 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.