BMC Bioinformatics

SOFTWARE Open Access

SimGBS: a rapid method for simulating large-scale genotyping-by-sequencing data

Jie Kang[1,5,2], Melanie K. Hess[1,4], Ken G. Dodds[1], Rudiger Brauning[1], John C. McEwan[1], Barry J. Foote[8], Judy F. Foote[8], Agnieszka Konkolewska[5,6,7], Shannon M. Clarke[1*] and Andrew S. Hess[1,3*]

*Correspondence:
Shannon M. Clarke
Shannon.Clarke@agresearch.co.nz
Andrew S. Hess
ahess@unr.edu
[1]Invermay Agricultural Centre, AgResearch, Mosgiel, New Zealand
[2]Charles Perkins Centre, The University of Sydney, Sydney, Australia
[3]Department of Agriculture, Veterinary and Rangeland Sciences, University of Nevada, Reno, USA
[4]University of Nebraska-Lincoln, Lincoln, USA
[5]Crops Research Centre, Teagasc, Oak Park, Carlow, Ireland
[6]Insight Centre for Data Analytics, University College Dublin, Belfield, Dublin 4, Ireland
[7]VistaMilk SFI Research Centre, Moorepark, Fermoy, Co. Cork, Ireland
[8]Hikurangi, New Zealand859 Russell Road, 0184

Abstract

Background Genotyping-by-sequencing (GBS) offers a cost-effective solution to access genomic information on many samples. The advent of GBS has promoted large scale genomic analyses, including linkage analysis, heritability estimation, inference of genetic relatedness and genome-wide prediction and association studies, in many non-model species. Low-depth GBS can be used as a cost-effective method to obtain genotypes; however, not all alleles will be captured during the sequencing process, which can lead to high levels of genotype uncertainty. New methods that take genotype uncertainty into account are needed, but these methods can be difficult to develop and benchmark on real data. Simulations provide a convenient way to benchmark different approaches, and therefore, are essential for assessing existing methods and guiding future method development.

Results SimGBS is a method for simulating large-scale GBS data from any real (e.g., reference) or simulated diploid genome. It is implemented in Julia but can also be accessed from R through the JuliaCall interface. Users can define populations with distinct demographic histories by modifying population size over a specified number of generations or providing a pedigree to generate structured populations. Most GBS parameters are highly customisable, including the choice of restriction enzyme and sequencing depth. SimGBS outputs both true genotype calls and DNA sequences in FASTQ format. The method is computationally efficient, capable of generating datasets for thousands of individuals; for example, simulating 172 samples required less than two hours while successfully reproducing the complexities observed in empirical GBS data.

Conclusion SimGBS is a valuable tool to assist with designing GBS experiments or evaluating bioinformatic pipelines and statistical methods for GBS data analysis in large scale population genetics studies.

Keywords Simulation, Genotyping-by-sequencing (GBS), Plant genomics, Animal genomics, Restriction enzyme

Background

Genotyping-by-sequencing (GBS) offers a cost-effective solution to obtain genomic information from hundreds of samples. Briefly, the GBS process can be described as follows [1, 2]: DNA molecules are digested into small fragments using restriction enzyme(s). A unique barcode adapter, used to distinguish fragments originating from different samples, as well as a common adapter are ligated to each GBS fragment. Samples are then combined into a library, which undergoes PCR amplification followed by fragment-size selection to select a fraction of the genome. The library is then loaded on a flow cell and sequenced. Typically, single-end short reads of approximately 100–150 bp are used which includes the barcode sequence portion of the adapter. Subsequently, the SNPs are identified and called using either a reference-free (comparing amongst GBS reads) or reference-based (comparing against the reference genome) approach. The GBS method is highly flexible, with the ability to modulate the level of genomic coverage (i.e., the proportion of regions that are covered by GBS reads) and which regions of the genome are being sequenced by changing restriction enzyme(s). Such flexibility offers a simple solution to reduce the target fraction of the genome, and therefore, makes GBS a particularly attractive method for genotyping many samples [3, 4]. Recent enhancements to GBS protocols, such as the highly multiplexed library preparation method, have enabled more samples to be sequenced at a lower (per-sample) depth to further improve the cost-effectiveness of GBS [5]. To date, GBS has been used for a variety of purposes, including understanding the evolutionary histories of non-model species [6, 7], exploring population stratifications [8, 9], and assisting plant and animal breeding [10–12].

Due to the process of sampling genomic fragments during sequencing, GBS data come with higher genotype uncertainty than conventional array-based genotyping assays (e.g., SNP-chips). Sometimes, there is a significant proportion of missing data, referring to loci where sequencing reads are present but depth is too low to reliably determine both alleles, resulting in uncertain or uncalled genotypes. However, bioinformatics workflows and statistical methods have been developed to take genotype uncertainty into account when handling GBS data [13–15], with some dedicated to tackling the issues associated with low-depth sequencing [16–19]. Properly benchmarking these methods has been quite challenging because the true genotypes are unknown. In addition, there is typically low SNP overlap between GBS and existing SNP chips. To address this scenario using real data would require comparing the results from GBS data with either SNP chips or very high-depth sequencing, both of which are cost-prohibitive on large numbers of samples.

A practical way of evaluating different analytical methods is through simulation. The development of methods for large-scale population genetics studies and genomic analyses (e.g., quantitative genetics) requires large numbers of individuals to be simulated. Most studies using genomics have focused on species with SNP chips available and therefore have focused on methods of development utilizing simulated SNP chip data [20–22], which is vastly different from GBS data. For example, whereas SNPs from arrays are traditionally uniformly distributed across the genome from a pre-defined set of loci, the markers obtained by GBS are influenced by the presence of restriction enzyme cut sites where the location of these cut sites may not be known in the species of interest. Furthermore, SNP chips often focus on common variation, so the minor allele frequency (MAF) distribution is typically less left-skewed than what is observed in GBS data. As

opposed to SNP chips, where genotypes are made available as the standard output, genotypes from GBS data are inferred based on the observed alleles and are often associated with some level of genotype uncertainty based on the observed numbers of each allele. Also, when sequencing depth is low, GBS frequently produces missing data, referring to the undersampling of genomic regions that are not sequenced or represented in some individuals. Given these differences between GBS and SNP chip data, established simulations do not adequately capture many key features of GBS data. Simulating SNP chip data for benchmarking GBS methods will fail to capture how those methods perform with real GBS data.

Versatile and high-throughput methods of simulating GBS data are needed for assessing existing bioinformatics workflows and statistical methods and guiding future method development. Although there are several tools available to simulate next-generation sequencing (NGS) data, most have focused on producing a large volume of genomic data, which often oversimplifies assumptions of population or data structure [23]. Other software packages were explicitly designed to simulate restriction-site associated (RAD) sequencing or GBS data [24–28] under realistic experimental conditions, such as mimicking allelic dropouts and PCR effects and creating populations with specified demographic histories. Among these, RADinitio [29] performs in silico restriction enzyme digestion, generates per-sample sequencing reads, and models population-level genetic variation. However, the associated computational costs make them less feasible for simulating large populations. Large-scale simulated GBS data, generated under realistic conditions, is not only useful for benchmarking the performance of bioinformatics workflows and analytic methods but also important for investigating the effectiveness of genome-wide association studies or genomic prediction using different methods for accounting for uncertainty in GBS genotypes, as well as exploring optimal strategy of distributing sequencing effort within a breeding population. Therefore, the ability to efficiently simulate GBS data for large populations, while appropriately reflecting the complexities of GBS data at the population level, is crucial to any GBS simulator.

Here we present SimGBS, a method to rapidly simulate GBS data based on Illumina HiSeq sequencing, implemented as an open-source software. Through simulation of population history, as well as the GBS process (Fig. 1), our goal is to offer an efficient solution to conduct large-scale simulations, while capturing most key features of GBS data. To showcase some key features of SimGBS, we simulated a GBS dataset using parameters extracted from real GBS data and then compared it to the real data.

Implementation

Overview

An overview of the workflow for SimGBS is presented in Fig. 2 and described in Supplementary file 1. A brief overview is provided in this section, with further details below. In the initial step, the package uses a given reference genome and restriction enzyme(s) to identify cut sites to perform virtual digestion. Next, a fragment size-selection step takes place to filter out raw GBS fragments that are too large or too small based on user-defined thresholds. The population structure is established using the gene-dropping method [30]. Briefly, a founder population is generated, and then parents are selected at random to produce offspring. This process is repeated for the number of generations defined by the user, and recombination events are randomly sampled and recorded to

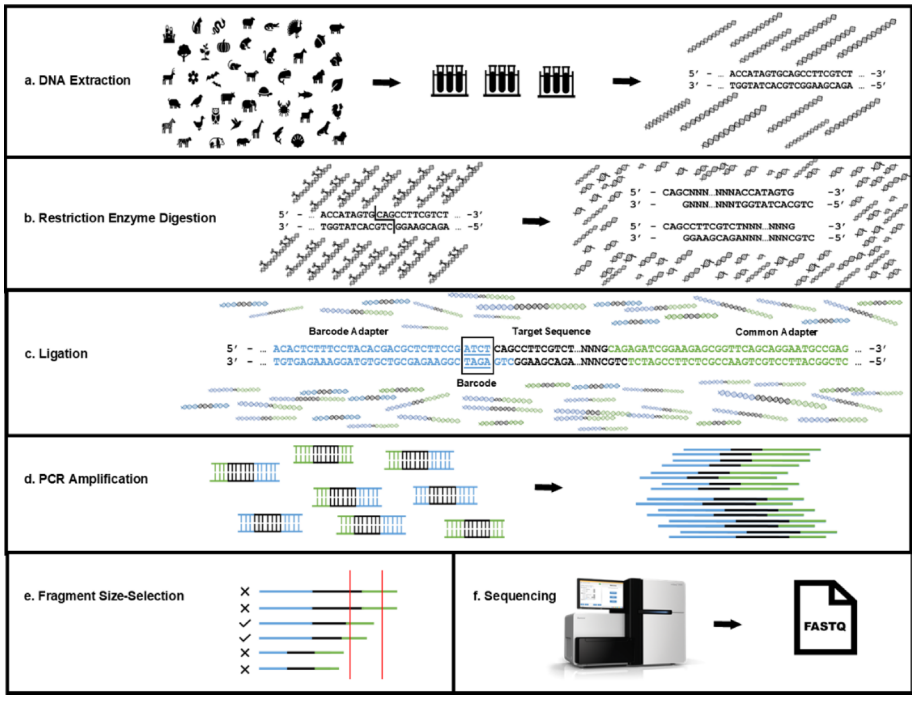

Fig. 1 Overview of genotyping-by-sequencing (GBS). GBS protocols can be broadly summarised into six steps: step (**a**) DNA extraction, DNA is isolated from the sample; step (**b**) restriction enzyme digestion, DNA molecules are digested into small fragments using restriction enzyme(s); step (**c**) Ligation, adaptors (i.e., short DNA sequences including barcodes) are ligated to the GBS fragments of each sample (with a unique barcode for each sample). This allows sequencing of hundreds of samples simultaneously, which greatly reduces the per-sample cost of sequencing. Step (**d**) PCR amplification, many thousands of GBS read templates are generated through PCR amplification. Step (**e**) Fragment size-selection, longer (e.g., >195 bp) and shorter (e.g., < 65 bp) fragments are removed; step (**f**) Sequencing, GBS libraries are then processed within a sequencer, genomic information is returned as sequencing data and stored in a FASTQ file

track cross-over events. Through this process, the passage of founder haplotypes onto descendants is tracked, allowing founder genotypes to be "dropped" down to individuals in later generations by looking up allele states from the corresponding chromosomal segments of its ancestor. Once SNP positions and genotypes are determined, alleles of each genotype are inserted into selected GBS fragments to generate the "true" genotypes for the individuals for which GBS data are being simulated. Allelic sequence depth is simulated considering sample-to-sample as well as locus-to-locus variation, in addition to random variation. Simulated GBS reads are generated that mimic the structure of a true read: a short DNA sequence serving as sample identifier (i.e., barcode) is attached to the 5′ end of each GBS read, and GBS reads are replicated on an individual sample and locus basis to reach the sampled sequencing depth level. Consistent with reads generated from a sequencer, simulated reads, along with their headers and quality scores, are written into FASTQ files.

Virtual digestion

A virtual digestion takes a genome assembly (or subset of a genome) and identifies the cut sites within that genome and fragments that would be generated from restriction enzyme digestion based on that genome. It is assumed that all cut sites of each restriction enzyme are cut with 100% efficiency. A size selection step allows users to define the lower and upper genomic fragment size limits (i.e., excluding barcode and adapter

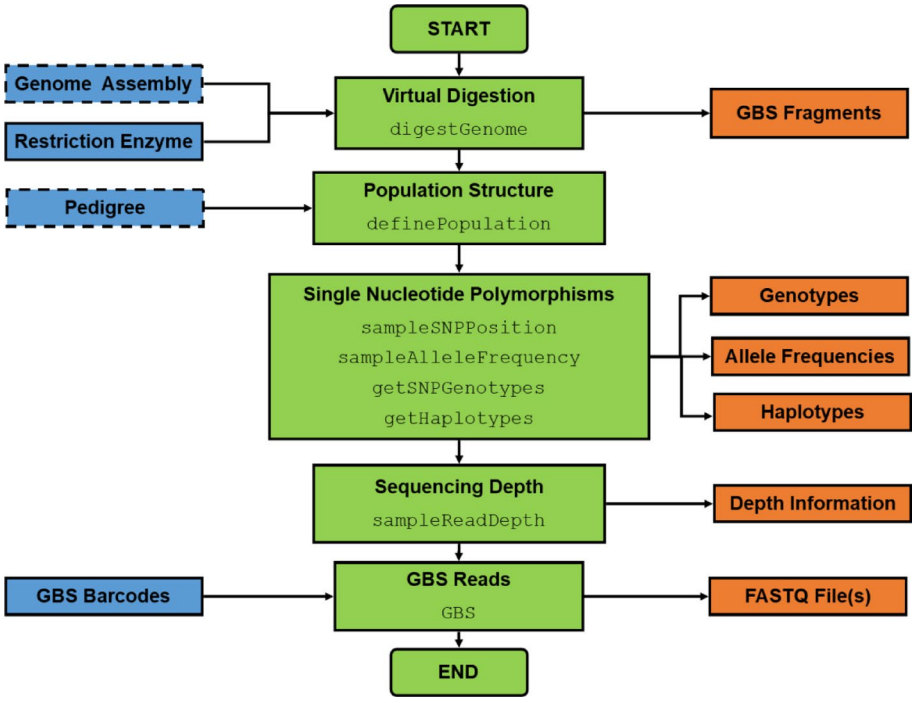

Fig. 2 Schematic flowchart of SimGBS. This flowchart provides an overview of the entire workflow of SimGBS. Blue and orange boxes represent different types of input and output of SimGBS, respectively, while green represents different functions (see Supplementary file 1). The name of each function has been labelled in non-bold text. Dotted line boxes (e.g., genome assembly and pedigree) indicate optional inputs

sequences) and select fragments within the specified range. In SimGBS, this can be used for the initial investigation into what restriction enzymes might be most suitable for use in a particular species, given its genome assembly. In the wider scope of SimGBS, this step identifies the fragments that fall in the specified range to use as the set of fragments that will be simulated. Single or double digest is possible, and users can choose from a list of commonly used restriction enzymes provided by SimGBS (e.g., *Ape*KI or *Pst*I-*Msp*I), or define their own enzyme motifs.

Population structure

SimGBS offers users great flexibility to generate different population structures, through controlling the size of the founder population and allowing increasing, decreasing or constant population size for a specified number of generations. The structure is stored, allowing the user to specify another round of population simulation or move on to the next step of the simulation process (Fig. 3). The user can also supply a pedigree file to generate an already established breeding population once the base population is simulated to form a historical LD structure. While the population structure is being established, the transmission of haplotypes is tracked across generations, but allelic states are assigned later during SNP simulation. This approach allows SimGBS to remain computationally efficient when simulating large populations with complex structures.

Single nucleotide polymorphisms

SNP densities are simulated based on either user-defined or default parameters, at both a global and a local level to capture the range of SNP densities observed in real GBS

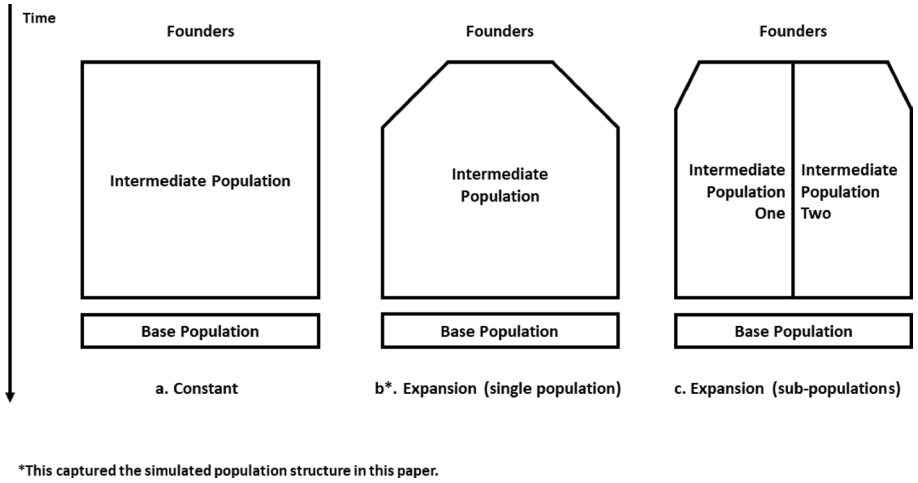

Fig. 3 Examples of different simulated population structures: **a** Constant—population size remains fixed over time; **b** Expansion (single population)—population size first increases linearly until it reaches to the targeted population size then is held constant. (Note that this population structure was implemented in this paper); **c** Expansion (sub-populations)—the size of each sub-population first increases linearly until it reaches to the targeted size then is held constant

data. Within a 1 Mb window, the SNP position is determined by performing a Bernoulli trial at each base, where the probability of an SNP occurring is equal to the sampled local density. Once SNP positions have been sampled, allele frequencies at the founder level are sampled for those SNPs that are located within GBS fragments of the appropriate size from a Beta distribution with default parameters of α and β equal to 0.62 and 0.82, respectively (Supplementary file 1). Founder haplotypes are generated by sampling allelic states from a Bernoulli distribution parameterised by the sampled allele frequency at each locus. When a SNP is introduced, the alternate allele is selected uniformly from the three nucleotides other than the reference allele. Genotypes for sequenced individuals are then propagated through the pedigree using the gene-drop method [30].

In addition to simulating the SNPs you might identify through GBS, there is also the option to simulate SNP-chip genotypes. SNPs are randomly sampled from the genome-wide set of SNPs such that within each 1 Mb window there are roughly the same number of SNPs. SNP chip genotypes often have a different allele frequency distribution to GBS genotypes, and to reflect this, the allele frequency distribution in the founder population can be established by modifying the Beta distribution parameters used to simulate the GBS allele frequencies. The ability to simulate both SNP-chip and GBS SNPs on the same individuals is required to enable comparisons between these two genotyping approaches.

Sequencing depth

We assume that variation in sequencing depth comes from three sources: sample-to-sample variation (e.g., DNA quality), fragment-to-fragment variation (e.g., methylation patterns influencing the efficiency of the restriction enzyme), and random noise (e.g., whether the fragment gets read on the sequencer). Further details regarding the choice of distributions can be found in Supplementary file 2. Expected sample depth (i.e., sequencing read depth on a per sample basis) and expected fragment depth (i.e., sequencing read depth on a per fragment basis) are sampled from independent Gamma

distributions with identical means, equal to the user-defined sequencing depth. By default, SimGBS generates fragment depth with a much larger variance than sample depth, since GBS protocols have been optimised to minimise between-sample depth variation [31]. Conversely, fragment-to-fragment variation is more difficult to reduce from a laboratory perspective, e.g., GC-rich regions are associated with PCR-amplification problems, due to the formation of secondary intramolecular structures, which may result in GC-rich regions having lower sequencing depth than other regions. To address two separate sources of variation in sequencing depth (i.e., sample-to-sample and fragment-to-fragment), the expected fragment depth for an individual at each fragment is calculated by multiplying their expected sample depth by the expected fragment depth and dividing by the user-defined sequencing depth. This yields an expected relationship between depth and call rate (i.e., the proportion of non-missing genotypes) at both the sample and SNP levels. To reflect the sampling process of fragments during sequencing, additional random variation was added to get the observed number of fragments per sample by sampling from a Negative Binomial distribution (see Supplementary file 2 for distribution parameters). In real GBS data, there is a deviation from the expected relationship between depth and call rate, therefore, the relationship between (log) SNP depth and call rates (i.e., the proportion of non-missing genotypes) was modelled using a logistic function, with extra variation introduced around the call rates at the given depth level without changing the mean depth. This provided relationships between depth and call rates that closely mimicked what is observed in real GBS data.

Generating simulated reads

Based on the known SNP positions and genotypes, SimGBS inserts variants into size-selected GBS fragments to generate GBS reads. SNPs that occur in a cut site (i.e., cut site variants) currently do not impact the depth of the read, however, this will be implemented in future versions of SimGBS. Similar to actual barcoding, a unique, short sequence is attached to the 5′ end of each GBS read to identify its origin. Each GBS read is then replicated to match with the sampled depth. Each single-end sequencing read is randomly sampled to read from the 3′ or 5′ end of the fragment. Fragments longer than 101 bp are trimmed to this length to generate reads, and fragments less than 101 bp are assumed to have been completely sequenced. Phred quality scores are assigned as "I" for all positions, corresponding to a quality score of 40 in Illumina 1.8+ or Sanger format (Phred + 33) [32]. Finally, all simulated GBS reads are written into FASTQ format [33], which comes with aheader that stores basic information about each read, including artificial sequencer name, flowcell ID, lane, coordinates on the plate, and a string of quality scores. Depending on the sample size and/or user-preferred data structure, there are multiple parameters controlling the number of samples from each (virtual) lane, as well as the number of lanes available within a (virtual) flowcell. Such information is made available to ensure compatibility with some existing bioinformatics pipelines.

Generating and processing GBS data

To showcase some key features of SimGBS, we simulated GBS data using parameters extracted from a real dataset, which is available on European Nucleotide Archive (under BioProject PRJEB54897). Briefly, DNA was extracted from tissue samples from a single herd of 172 New Zealand Saanen dairy goats (*Capra hircus*) using the high throughput

tissue extraction method as described by Clarke et al. [31]. Specifically, the samples were collected from female goats born in 2014 and 2015 and genotyped in 2016 as part of standard commercial operations. Tissue samples were obtained using Allflex Tissue Sampling Units (TSUs; https://www.allflex.co.nz/product/tsu-tissue-sampling-unit/), a minimally invasive method used routinely in the industry. Further details on the herd, including its origin, selection history, and genotyping strategy, are available in Wheeler et al. [31]. and Dodds et al. [31]. Following the GBS protocols described in Elshire et al. [1], DNA molecules were digested into fragments using the *PstI* restriction enzyme. Single-end sequencing (with read length up to 100 bp) was performed on an Illumina HiSeq2500, using v4 chemistry. The resulting data set, contained 271 Million reads after demultiplexing and trimming, with an average of 1.45 million reads per sample.

Simulated GBS data were generated based on the goat population as described in Supplementary file 1. Briefly, 100 founders were first increased to a population of 200 individuals over 20 generations. Next, individuals were randomly crossed for another 50 generations, while the population size remained constant. The last four generations were assigned as the base population and 172 of those individuals were sampled for simulated GBS sequencing. The simulated population model, consisting of an initial expansion phase followed by random mating, was used for demonstration purposes to illustrate the flexibility of SimGBS in generating populations with varying demographic histories. It was not intended to replicate the exact breeding structure of the real goat herd, for which pedigree information was unavailable. The *PstI* restriction enzyme was used to generate GBS fragments from the *ARS1* goat genome and average sequencing depth was set at 6.5.

The real and simulated GBS data were processed through the snpGBS pipeline [34]: barcode and common adapters were removed from the raw GBS data using cutadapt [35], demultiplexed GBS data were then mapped back onto the *ARS1* goat assembly [36] using bowtie2 [37], and SNPs were identified using bcftools [38]. The outputs from this bioinformatic pipeline were then used to inform the parameters used by SimGBS (Supplementary file 2). Both real GBS data (in FASTQ format) and the resulting SNP dataset (in VCF format) were uploaded to the sequence read archive (SRA) and European variation archive (EVA) and made publicly available under BioProject PRJEB54897. Analyses were carried out to evaluate the performance of SimGBS, with the focus on comparing analysis results from simulated GBS data to those obtained from the real data, using R functions provided by KGD and GUS-LD [17, 19].

Results and discussion

System requirements

This simulation of GBS data from 172 goats, as described above, was performed on Julia (Version 1.1.1), using a Linux HPC with 16 allocated cores. The total run time was less than two hours (i.e., 01:43:33), with peak memory usage at only 7.1 GB. Since there are no extra system requirements to use the software (aside from those administrated by Julia), similar jobs can also be executed on a lower-spec device with increased running time. We ran the same simulation in Julia (Version 1.6.4), using a 64-bit Windows machine with 2 cores and 8 GB RAM. We found that the most efficient way to implement SimGBS was to simulate the population structure of all chromosomes, then acquire SNP genotypes and simulate reads on chromosomes individually—this approach

reduces the computational demand of gene dropping and therefore requires less RAM and is faster. For instance, under the same settings, the PC run only took 679.5 s to complete the simulation on chromosome 1 (with a length of ~ 154 Mb). However, if this approach is to be taken it is critical to save the full pedigree file from the initial chromosome to ensure the same mating structure for all simulated chromosomes. We also attempted to conduct a similar simulation using RADinitio [29], a program to simulate RAD-Seq data, to compare runtime with SimGBS. However, using the same Linux HPC as above, our first attempt failed to complete due to excessive computational time (>7 days). The simulation was then run on a per chromosome level, with 4 CPUs allocated to each job. Across a total of 29 jobs, the shortest run time was just under 24 h (23 h 06 m), while the longest still lasted over 2 days. We concluded that SimGBS is more suited to simulating GBS data on large numbers of individuals than RADinitio.

GBS fragments

Through virtual digestion, 1,489,394 GBS fragments were generated using the *Pst*I enzyme across 29 chromosomes of the goat genome. Of these fragments, 99,266 (~ 6.7%) fell within the (genomic) size selection thresholds of 65 and 195 bases (bp). Across 2482 1-Mb fixed-size windows, on autosomal chromosomes, the average *local* genomic coverage of selected fragments was estimated to be 5225 ± 3397 bp per Mb (Fig. 4a). Overall, available *Pst*I fragments covered approximately 0.51% of the total genome assembly for these chromosomes. Surprisingly, genomic coverage of mapped reads from the real GBS data was estimated to be around 1.42%. Further investigation showed that most fragments (75.17% of "uniquely mapped" reads) were identified in both the simulated and real datasets and the observed discrepancy was mainly driven by the presence of fragments that fell just outside the size selection thresholds for the real datasets, where 22.38% and 2.32% of "uniquely mapped" reads had matching simulated GBS fragments with size below 65 bp or above 195 bp, respectively. These fragments were excluded from the simulation, but, due to the imperfect nature of size exclusion in practice, were found in the real data. Other factors, such as incomplete digestion in the real GBS data, may also contribute to the inflated coverage of real GBS data, but the occurrence was relatively low (0.12% of "uniquely mapped" reads).

SNP density

A total of 139,423 SNPs were identified in the real dataset using snpGBS, of which 109,418 were reported from KGD after filtering out those with MAF = 0 and depth < 0.1 (Table 1). With 44 ± 33 SNP being observed from each 1 Mb window (Fig. 4c), dividing this number by the *local* genomic coverage would provide the estimated *local* SNP density of, on average, 0.0086 ± 0.0030 SNP per bp (Fig. 4d); in other words, we expect to find approximately one SNP every 115 bp. Local GC sequence content can influence the size distribution of fragments and subsequently the number of SNPs observed in a region from GBS data. Since GC-rich regions are more likely to produce fragments that fall into the size-selection range of GBS (e.g., < 200), more SNPs may be observed in GC-rich regions (Fig. 4b). A strong linear relationship between GC-count and SNP count (and genomic coverage) confirmed this issue (Fig. 5). Hence, dividing the SNP count by genomic coverage to calculate *local* SNP densities can reflect the relative proportion of SNPs within each 1 Mb window. The resulting distribution of estimated *local*

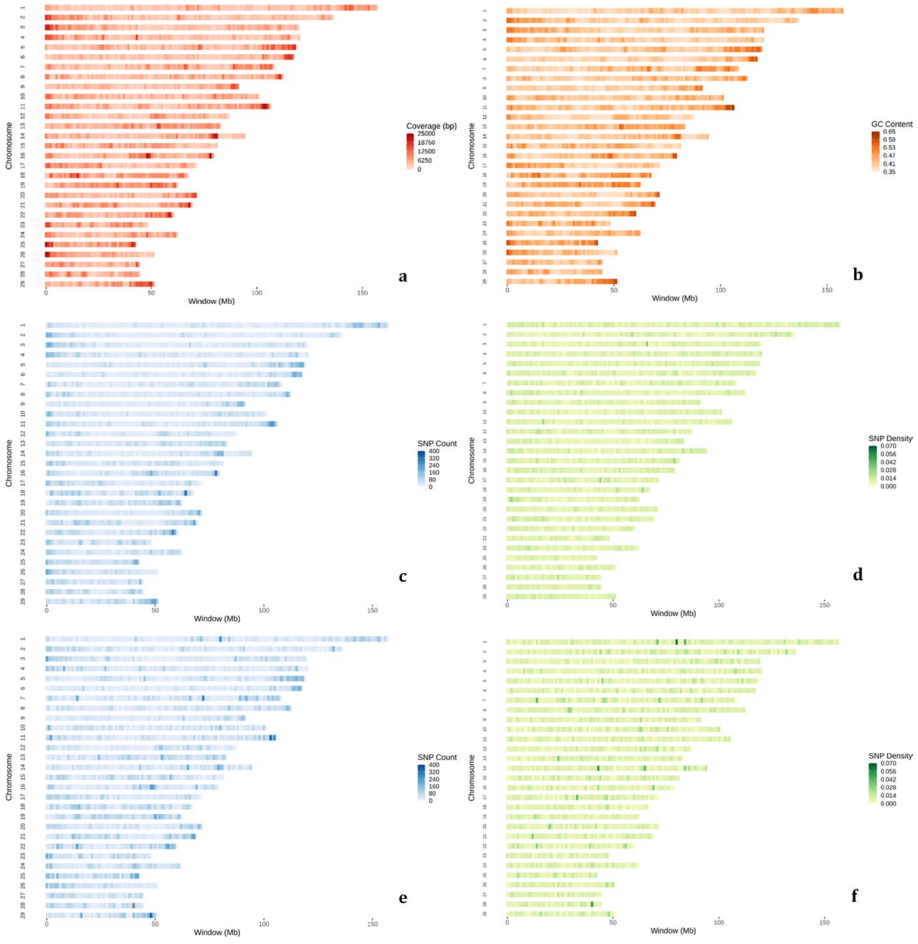

Fig. 4 Comparisons between real and simulated GBS data, including **a** Genomic regions covered by simulated *PstI* GBS fragments within each 1-Mb window; **b** Estimated GC content of the ARS1 goat assembly within each 1-Mb window; **c** the number of SNPs observed (from real GBS data) within each 1-Mb window; **d** Estimated SNP density (from real GBS data) within each 1-Mb window; **e** Number of SNPs observed (from simulated GBS data) within each 1-Mb window; **f** Estimated SNP density (from simulated GBS data) within each 1-Mb window

Table 1 Summary statistics from analysing real and simulated GBS data

	Real	SimGBS
Number of SNP	115,191	112,272
Number of samples	172	172
Mean depth	5.64 ± 10.20	5.68 ± 8.58
Mean call rates	0.68 ± 0.03	0.63 ± 0.03

Both real and simulated GBS were processed through snpGBS (incl. SNP calling), and summary statistics were extracted from KGD output.

SNP densities, or simply the underlying *global* SNP density model, is right-skewed with a heavy tail (Fig. 6). After sampling SNP positions by following the proposed method, we found that both the overall distribution of estimated (*local*) SNP densities and the number of SNP found per GBS fragment are very close between real and simulated GBS data (Figs. 4, 6 and 7).

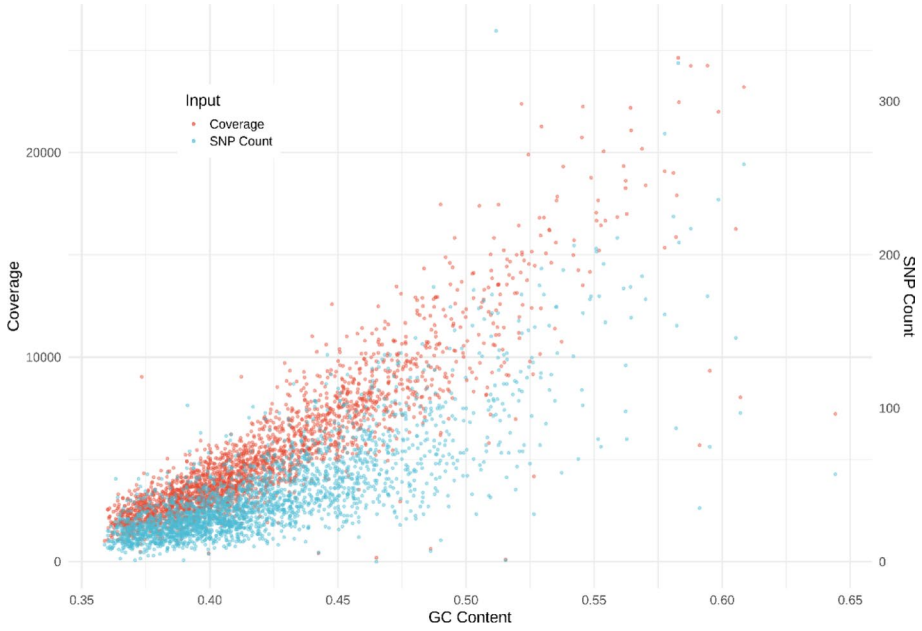

Fig. 5 Plot of estimated GC content of the *ARS1* goat assembly against genomic coverage and SNP counts across different (1-Mb) windows

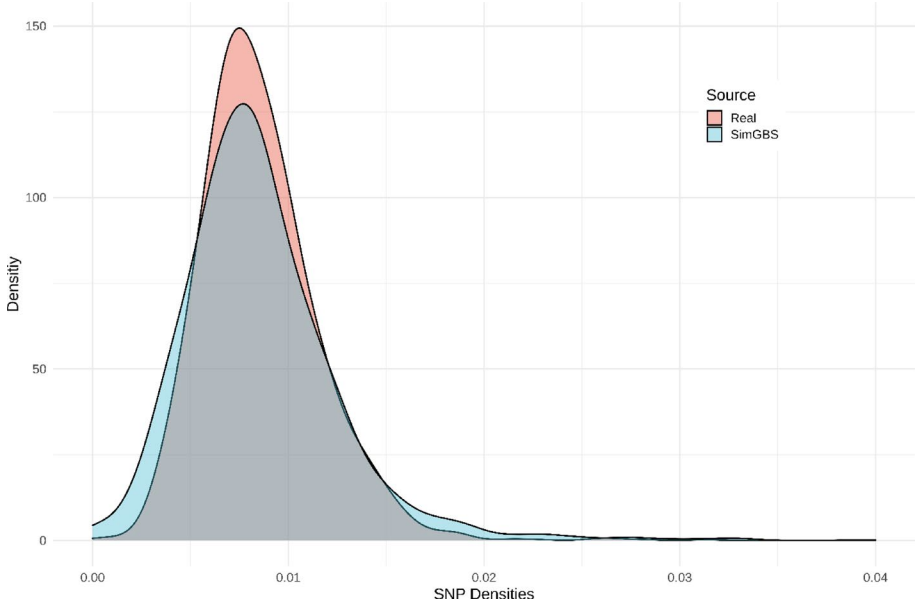

Fig. 6 Distribution of estimated *Local* SNP densities obtained from real (red) and simulated (blue) GBS data

Depth and call rates

The average sequencing depth of real GBS data was 5.64, with estimated variances of sample and SNP depth of 1.72 and 53.15, respectively. In the simulation, the average sequencing depth was assigned to be 6.5, as some reads are expected to be dropped during the bioinformatics processing. The resulting average sequencing depth of simulated data was 5.68, and the variances of sample and SNP depth were estimated to be 1.47 and 54.71, respectively. Furthermore, the average call rate from real data was 0.68 ± 0.03, whereas the average call rate of simulated data was 0.63 ± 0.03. The variances of SNP call

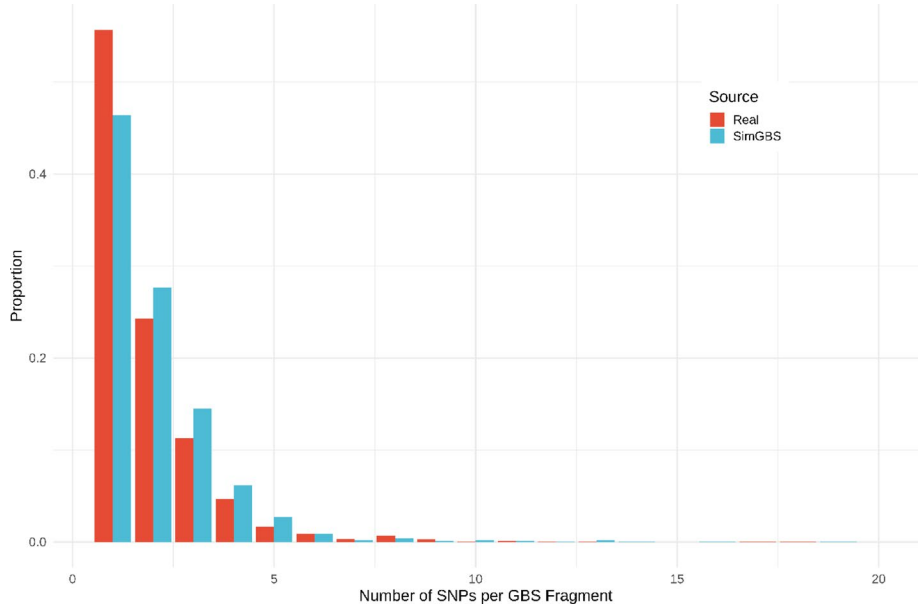

Fig. 7 Number of SNPs per GBS fragment observed from real (red) and simulated (blue) GBS data

rates from the real and simulated GBS data were 0.147 and 0.119, respectively, whereas the variances of sample call rates, were much smaller, at 0.0009 and 0.0008, respectively. The simulation successfully captured the relationship between SNP depth and call rate (Fig. 8) as well as the relationship between sample depth and call rates (Fig. 9).

Minor allele frequency

The MAF distribution from the real GBS data is slightly right-skewed, with the mean and variance estimated to be 0.21 and 0.025, respectively (Fig. 10). In comparison, the MAF distribution in the founder population was very close to the Beta distribution we selected to sample MAF (Supplementary file 2); while the MAF distribution of true simulated genotypes in the 172 GBS individuals was intermediate between the real GBS data and the simulated founders. However, the MAF distribution obtained from the simulated GBS reads was flatter than all other metrics (Fig. 10) because 27,151 (out of 139,423) SNPs were automatically removed by the default KGD filters at MAF = 0 and SNP depth < 0.01. This indicates that a significant proportion (approx. 19.5%) of the simulated SNPs are deemed to be non-polymorphic due to a relatively small population size (Supplementary file 1) or almost completely missing. The ability to accurately capture rare (low MAF) variation is a common challenge in population genetics and has frequently been discussed in the context of GWAS [39] and imputation [40]. In the context of simulating a structured population, the challenge is that the SNPs need to be specified in the founder population, and these can drift to fixation, reducing the number of SNPs that are segregating within the population. This can cause a large reduction in the number of simulated SNPs, particularly when the population size is small. Increasing population size to counter this is effective but will also increase computation time. Introducing mutations into the population will also counter this effect to some extent and while this has not yet been incorporated into SimGBS, it is planned to be included in future versions of the package. Furthermore, many commonly used restriction enzymes (e.g., *ApeKI*, *Pst*I and *Msp*I) have GC in the cut sites. GC pairs mutate at an approximately

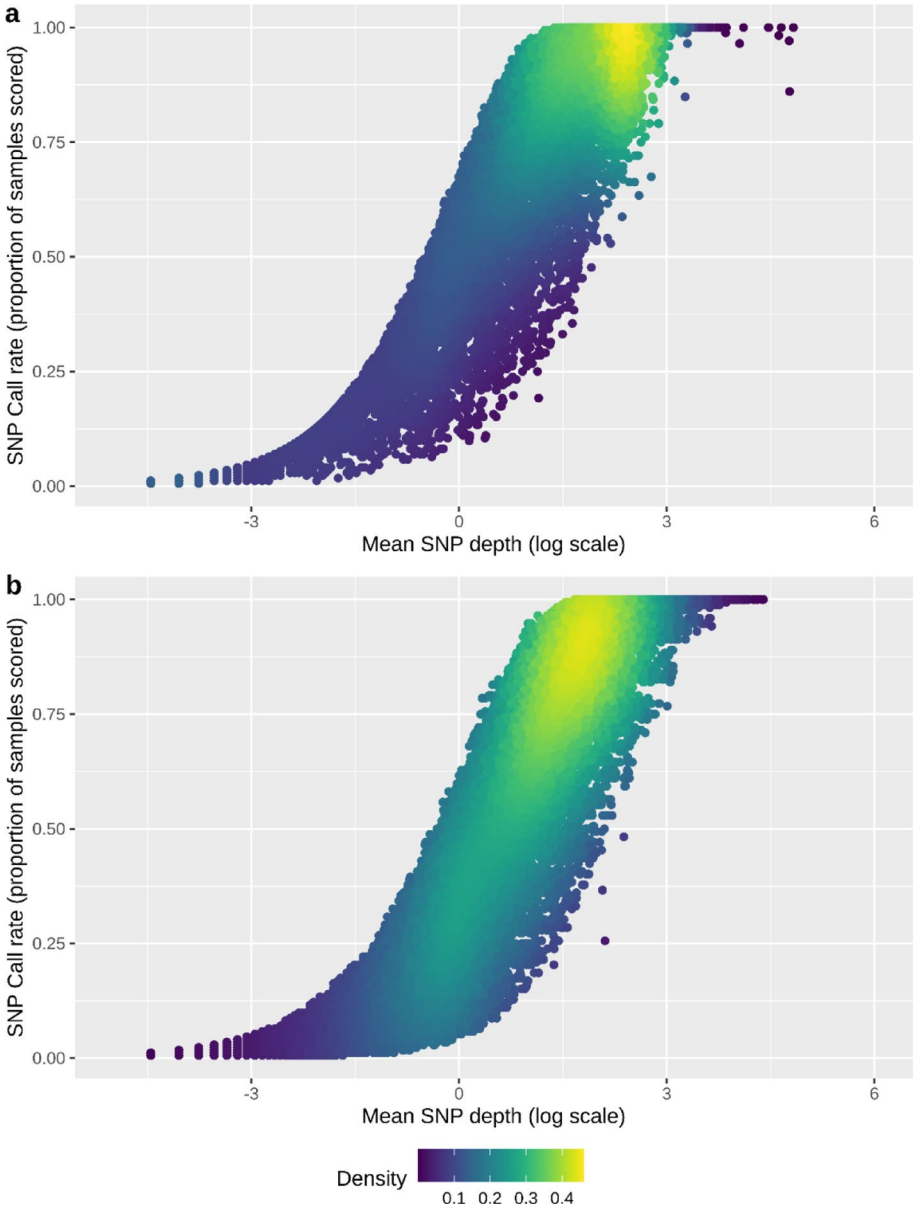

Fig. 8 SNP depth versus call rates from **a** real and **b** simulated GBS data

ten-fold higher rate than other combinations which can lead to cut-site variation (i.e., the loss or gain of restriction site) if the cut site becomes polymorphic [41]. Since a mutation occurring at the cut site is likely to be in strong LD with SNPs that are located within the GBS fragment, it can cause some shifts in the MAF distribution and ultimately the heterozygosity of SNP genotypes. For example, a heterozygous genotype can be seen as homozygous if one of the alleles is associated with the cut site variation.

Linkage disequilibrium

Linkage disequilibrium (LD) describes non-random association between alleles at different loci. In relation to the distances between loci, the shape of the LD curve often reveals useful insights into population history. For instance, if there has been continuous growth in the population, LD is expected to decrease rapidly as the distance between

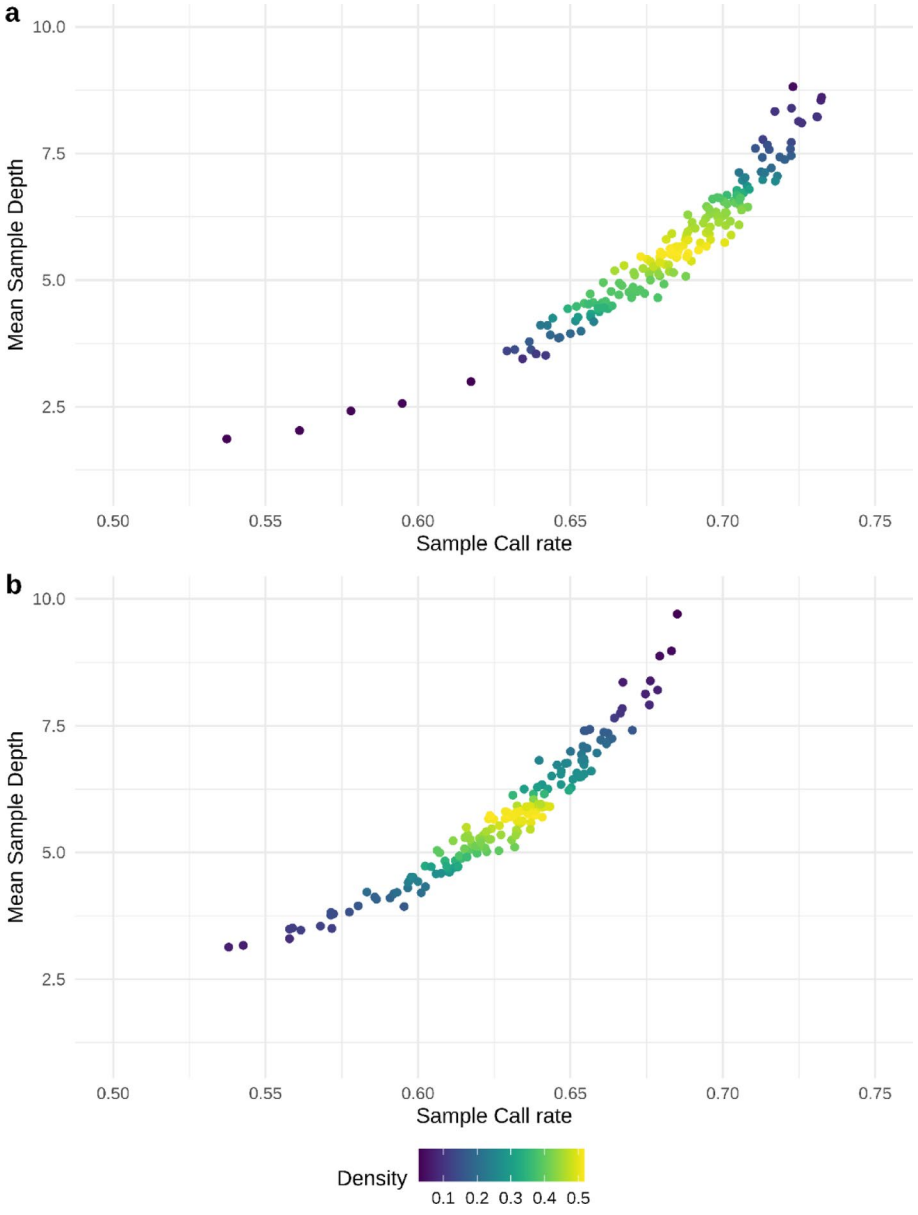

Fig. 9 Sample depth versus call rates from real (**a**) and simulated (**b**) GBS data

loci increases (i.e., LD decay). We estimated LD in the real and simulated data using both imputed genotypes (via mean imputation, after removing SNPs with more than 50% missing genotypes) and through implementing GUS-LD [19], an R package that takes genotype uncertainties into account. For the simulated data we additionally calculated LD based on the true genotypes.

Both datasets showed a slow decline of LD using all methods, whereby estimates from GUS-LD and the true genotypes were consistently higher than those calculated using imputed genotypes (Fig. 11). This was anticipated because the mean-imputation approach is more likely to depress LD estimates: mean imputation replaces missing genotypes with a constant, resulting in no variation for the imputed genotypes and no covariance between the imputed genotypes and neighbouring SNPs, thereby resulting in LD of zero for the set of individuals with imputed genotypes, and a lower estimate of

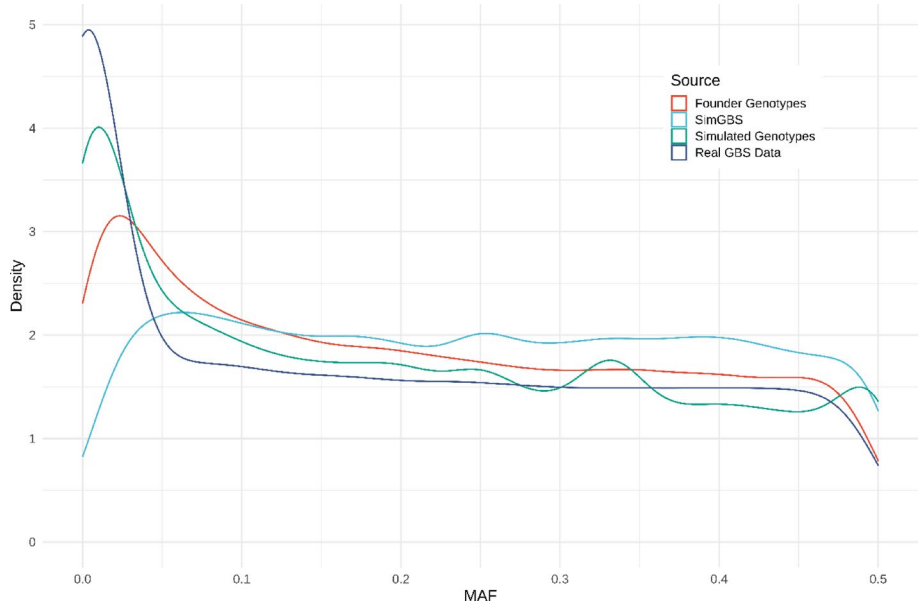

Fig. 10 Minor allele frequency (MAF) distributions comparing empirical and simulated datasets. The curves represent (i) Founder genotypes simulated from sampled allele frequencies (red), (ii) SimGBS genotypes inferred from simulated GBS reads after variant calling (teal), (iii) Simulated genotypes of the 172 individuals before read simulation (green), and (iv) Real GBS data from the goat population (purple). The close similarity between the SimGBS and real GBS data distributions demonstrates that the simulated dataset reproduces the allele frequency spectrum observed in empirical data

LD overall. LD estimated from GUS-LD was slightly inflated compared to LD calculated using true genotypes, and this is potentially driven by SNPs with extremely low depth [19]. Using our simulated dataset, we were able to show that GUS-LD captured the true LD pattern more accurately than mean imputation as it more closely followed the LD decay of the true genotypes, and therefore, GUS-LD should be used for estimating LD when using GBS-type data.

The LD decay plots for the two datasets were very similar (Fig. 11); however, the simulated dataset showed slightly less variation in LD between SNPs at a given distance than the real dataset, and the decay curve had a slightly different slope, decaying a little faster than the real dataset. These are likely due to slight differences between the demographic history of the real dataset and what we have simulated.

Applications

SimGBS was primarily designed to efficiently simulate large-scale GBS data, providing a useful resource for evaluating and developing analytical methods. A key application is the benchmarking of variant callers for GBS data. Benchmarking with empirical datasets is often limited by the absence of true SNP genotypes and positions; however, this information can be easily retrieved from simulated dat. With SimGBS, the accuracy of variant detection method can be quantified by comparing the observed SNP genotypes to the known "truth", and evaluated alongside other performance indicators, such as computational efficiency and compatibility.

SimGBS can generate datasets under various settings, including different restriction enzyme, sample sizes, population structure and sequencing depth, to assess approaches that account for genotype uncertainty (e.g., KGD) or other downstream analyses. For example, SimGBS was used to show that GUS-LD was a less biased estimator of linkage

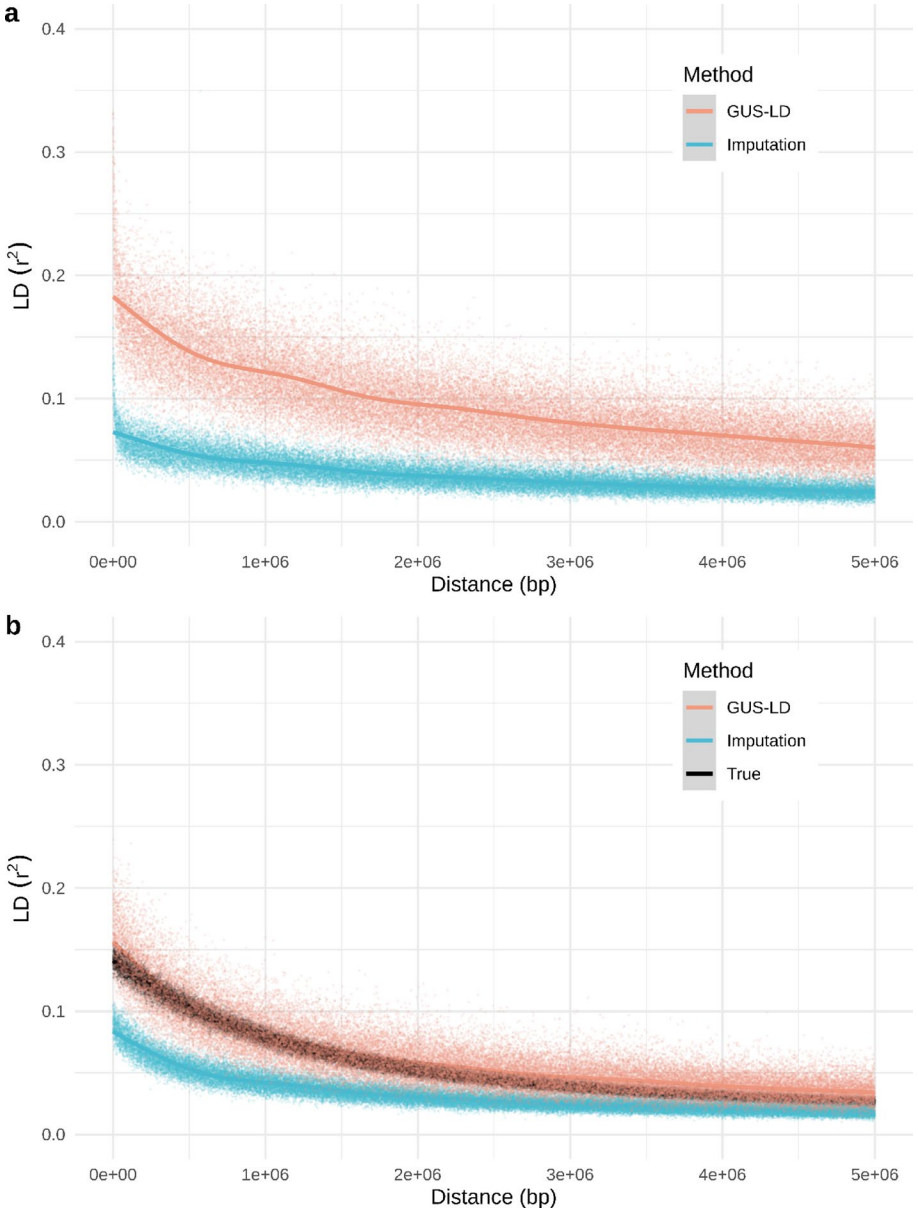

Fig. 11 Plot of linkage disequilibrium (LD) decay estimated from real (**a**) and simulated (**b**) GBS data using different methods

disequilibrium than methods using mean imputed genotypes (Fig. 11). Simulated datasetscan also guide improvements to such methods by quantifing the bias of LD estimates at lower sequencing depth and evaluating adjustments.

When designing new GBS experiments, SimGBS can be used to optimise experimental parameters (e.g., sequencing effort allocation) to balance cost and data quality. Beyond research applications, SimGBS also provides an educational tool for demonstrating the mechanism underlying reduced-representation sequencing. The simulated datasets can further facilitate the development and testing of new statistical and bioinformatics methods in genetics.

Limitations

While SimGBS offers flexible control over population structure and pedigree definition, it has so far been tested on small to moderately complex pedigrees, such as those in the example simulation (interactive tutorial) where populations were mated sequentially before applying a pedigree. Because SimGBS adopts a gene-drop approach that tracks haplotype transmission and recombination through generations, rather than explicitly resolving pedigree loops, the presence of cyclic relationships does not directly affect computation.

However, runtime and memory performance have not yet been systematically benchmarked for extremely large or highly inbred populations, which represents an area for future development. The comparison between simulated and empirical datasets in this study reflects a single representative realisation under fixed simulation parameters. Although random seeds were not fixed in this analysis, all stochastic processes in SimGBS can be reproduced by specifying a seed, allowing users to generate replicate datasets for assessing simulation variability and benchmarking performance.

Future development

SimGBS will continue to be developed to capture key features of emperical GBS data. Future versions will incorporate modelling of mutation processes, variability in restriction enzyme cut sites, enhanced flexibility in fragment size selection, including probabilistic modelling to capture the imperfect nature of this process, as well as additional forms of genetic variation and heterogeneity in sequencing quality scores. There will also be ongoing efforts to enhance computational efficiency through parallelisation and workflow optimisation.

In parallel, we plan to implement features enabling explicit modelling of sequencing error and imperfect digestion to further bridge simulated and empirical data characteristics. would enable more comprehensive benchmarking analyses, including comparisons of genotype concordance and imputation accuracy, extending previous benchmarking efforts such as those by Torkamaneh and Belzile [42] and the *GBSathon* study [43].

Conclusion

SimGBS is an open-source, data-driven computational tool to simulate GBS data that captures the complexity observed in real GBS data. It was designed to be time- and computationally efficient at simulating large numbers of individuals (> 1000), making it suitable for benchmarking different analysis methods in both population and quantitative genetics and aiding in the design of GBS experiments. SimGBS is an extremely flexible package, with many parameters that can be optimized for your purpose. It is written in Julia and can also be run from R. We demonstrated the utility of SimGBS by showing that GUS-LD, a package designed for low-depth estimation of LD, gave less biased estimates of LD than mean imputation of genotypes. SimGBS is an invaluable tool for exploring the use of low-depth GBS data in genomic analyses.

Availability and requirements

Project name: SimGBS.

Project home page: https://github.com/kanji709/SimGBS.jl

Interactive tutorial: https://colab.research.google.com/github/kanji709/SimGBS.jl/blob/master/tutorials/SimGBS_Julia_Colab_Notebook.ipynb
Operating system(s): Platform independent.
Programming language: Julia (can be accessed from R via the R package JuliaCall).
Other requirements: Julia 1.0 or higher.
License: MIT.
Any restrictions to use by non-academics: None.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-025-06343-6.

Supplementary Material 1: Tutorial for running SimGBS in Julia and R.

Supplementary Material 2: Parameter estimation for SimGBS.

Supplementary Material 3: Fig. S1. Visualisation of outputs at each stage of the SimGBS workflow.

Acknowledgements
This project was supported by the Teagasc Walsh Scholarship. GBS was performed by GenomNZ from AgResearch, Invermay, New Zealand (www.genomnz.co.nz). We thank Dr Jeanne Jacobs and Dr Roy Costilla for their helpful comments.

Author contributions
JK, MKH, and ASH developed and implemented the SimGBS software and participated in drafting the manuscript. KGD, RB, JCM, and SMC also participated in drafting the manuscript and guiding this project. BJF and JFF provided goat genotypes. AK and JK tested the software, drafted the documentation and developed an [interactive tutorial] (https://colab.research.google.com/github/kanji709/SimGBS.jl/blob/master/tutorials/SimGBS_Julia_Colab_Notebook.ipynbhttps://colab.research.google.com/github/kanji709/SimGBS.jl/blob/master/tutorials/SimGBS_Julia_Colab_Notebook.ipynb). All authors contributed to, read, and approved the final manuscript.

Funding
Not applicable.

Data availability
Demultiplexed GBS data (in FASTQ format) and SNP data (in VCF format) are made publicly available on European Nucleotide Archive (under BioProject PRJEB54897). The corresponding simulated data can be obtained by following the instructions described in Supplementary file 1.

Declarations

Ethics approval and consent to participate
This study had animal ethics approval AE15261 from the AgResearch animal ethics committee.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 30 June 2025 / Accepted: 1 December 2025
Published online: 09 December 2025

References

1. Elshire RJ, Glaubitz JC, Sun Q, Poland JA, Kawamoto K, Buckler ES, Mitchell SE. A robust, simple genotyping-by-sequencing (GBS) approach for high diversity species. PLoS ONE. 2011;6(5):19379.
2. Davey JW, et al. Genome-wide genetic marker discovery and genotyping using next-generation sequencing. Nat Rev Genet. 2011;12(7):499–510.
3. Poland JA, et al. Development of high-density genetic maps for barley and wheat using a novel two-enzyme genotyping-by-sequencing approach. PLoS ONE. 2012;7(2):e32253.
4. Brouard J-S, et al. Low-depth genotyping-by-sequencing (GBS) in a bovine population: strategies to maximize the selection of high-quality genotypes and the accuracy of imputation. BMC Genet. 2017;18(1):1–14.
5. Torkamaneh D, et al. NanoGBS: a miniaturized procedure for GBS library preparation. Front Genet. 2020;11:67.
6. Narum SR, Buerkle CA, Davey JW, Miller MR, Hohenlohe PA. Genotyping-by-sequencing in ecological and conservation genomics. Mol Ecol. 2013;22(11):2841.
7. Escudero M, Eaton DA, Hahn M, Hipp AL. Genotyping-by-sequencing as a tool to infer phylogeny and ancestral hybridization: a case study in carex (cyperaceae). Mol Phylogenet Evol. 2014;79:359–67.

8. Larson WA, Seeb LW, Everett MV, Waples RK, Templin WD, Seeb JE. Genotyping by sequencing resolves shallow population structure to inform conservation of Chinook salmon (*Oncorhynchus tshawytscha*). Evol Appl. 2014;7(3):355–69.
9. Eltaher S, Sallam A, Belamkar V, Emara HA, Nower AA, Salem KF, Poland J, Baenziger PS. Genetic diversity and population structure of f3: 6 Nebraska winter wheat genotypes using genotyping-by-sequencing. Front Genet. 2018;9:76.
10. De Donato M, Peters SO, Mitchell SE, Hussain T, Imumorin IG. Genotyping-by-sequencing (GBS): a novel, efficient and cost-effective genotyping method for cattle using next-generation sequencing. PLoS ONE. 2013;8(5):62137.
11. Hayes BJ, Cogan NO, Pembleton LW, Goddard ME, Wang J, Spangenberg GC, Forster JW. Prospects for genomic selection in forage plant species. Plant Breed. 2013;132(2):133–43.
12. He J, Zhao X, Laroche A, Lu Z-X, Liu H, Li Z. Genotyping-by-sequencing (GBS), an ultimate marker-assisted selection (MAS) tool to accelerate plant breeding. Front Plant Sci. 2014;5:484.
13. Glaubitz JC, Casstevens TM, Lu F, Harriman J, Elshire RJ, Sun Q, Buckler ES. TASSEL-GBS: a high capacity genotyping by sequencing analysis pipeline. PLoS ONE. 2014;9(2):90346.
14. Tinker NA, Bekele WA, Hattori J. Haplotag: software for haplotype-based genotyping-by-sequencing analysis. G3 Genes Genomes Genet. 2016;6(4):857–63.
15. Torkamaneh D, Laroche J, Bastien M, Abed A, Belzile F. Fast-GBS: a new pipeline for the efficient and highly accurate calling of SNPs from genotyping-by-sequencing data. BMC Bioinform. 2017;18(1):1–7.
16. Korneliussen TS, Albrechtsen A, Nielsen R. ANGSD: analysis of next generation sequencing data. BMC Bioinform. 2014;15(1):1–13.
17. Dodds KG, McEwan JC, Brauning R, Anderson RM, van Stijn TC, Kristj´ansson T, Clarke SM. Construction of relatedness matrices using genotyping-by-sequencing data. BMC Genom. 2015;16(1):1–15.
18. Bilton TP, Schofield MR, Black MA, Chagn´e D, Wilcox PL, Dodds KG. Accounting for errors in low coverage high-throughput sequencing data when constructing genetic maps using biparental outcrossed populations. Genetics. 2018;209(1):65–76.
19. Bilton TP, McEwan JC, Clarke SM, Brauning R, van Stijn TC, Rowe SJ, Dodds KG. Linkage disequilibrium estimation in low coverage high-throughput sequencing data. Genetics. 2018;209(2):389–400.
20. Karaman E, et al. An upper bound for accuracy of prediction using GBLUP. PLoS ONE. 2016;118:e0161054.
21. Zhao T, et al. Fast parallelized sampling of Bayesian regression models for whole-genome prediction. Genet Sel Evol. 2020;52(1):1–11.
22. Dekkers J, Hailin Su, and, Cheng J. Predicting the accuracy of genomic predictions. Genet Sel Evol. 2021;53(1):1–23.
23. Escalona M, Rocha S, Posada D. A comparison of tools for the simulation of genomic next-generation sequencing data. Nat Rev Genet. 2016;17(8):459–69.
24. GBS-Pacecar. A GBS read simulator to put your de Novo GBS pipeline through the paces! https://github.com/halelab/GBS-Pacecar
25. Eaton, Deren AR, et al. Misconceptions on missing data in RAD-seq phylogenetics with a deep-scale example from flowering plants. Syst Biol. 2017;66(3):399–412.
26. Lepais O, Jason T, Weir. SimRAD: an R package for simulation-based prediction of the number of loci expected in RAD seq and similar genotyping by sequencing approaches. Mol Ecol Resour. 2014;14(6):1314–21.
27. Mora-Márquez F, et al. Ddradseqtools: a software package for in Silico simulation and testing of double-digest RAD seq experiments. Mol Ecol Resour. 2017;17(2):230–46.
28. Timm H, et al. DDRAGE: A data set generator to evaluate DdRADseq analysis software. Mol Ecol Resour. 2018;18(3):681–90.
29. Rivera-Col´on AG, Rochette NC, Catchen JM. Simulation with radinitio improves Radseq experimental design and sheds light on sources of missing data. Mol Ecol Resour. 2021;21(2):363–78.
30. Cheng H, Garrick D, Fernando R. XSim: simulation of descendants from ancestors with sequence data. G3 Genes Genomes Genet. 2015;5(7):1415–7.
31. Clarke SM, Henry HM, Dodds KG, Jowett TW, Manley TR, Anderson RM, McEwan JC. A high throughput single nucleotide polymorphism multiplex assay for parentage assignment in New Zealand sheep. PLoS ONE. 2014;9(4):93392.
32. Ewing B. Base-calling of automated sequencer traces using phred. II. Error probabilities Genome Res. 1998;8(3):186–94.
33. Cock, Peter JA, et al. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Res. 2010;38(6):1767–177.
34. Kang J, Dodds K, Byrne S, Faville M, Black M, Hess A, Hess M, McCulloch A, Jacobs J, Milbourne D, Wilcox P, Brauning R. 2021. snpGBS: a simple and flexible bioinformatics workflow to identify SNPs from genotyping-by-sequencing data. In: S. Hartmann, S. Bachmann-Pfabe, S. Byrne, U. Feuerstein, B. Julier, R. Kölliker, D. Kopecky, I. Roldan-Ruiz, T. Ruttink, J.-P. Sampoux, B. Studer, T. Vleugels, pp. 67-70ang J, Dodds K, Byrne S, Faville M, Black M, Hess A, Hess M, McCulloch A, Jacobs J, Milbourne D, Wilcox P, Brauning R, Bachmann-Pfabe S, Byrne S, Feuerstein U, Julier B, Kölliker R, Kopecky D, Roldan-Ruiz I, Ruttink T, Sampoux J-P, Studer B, Vleugels T, editors. Exploiting genetic diversity of forages to fulfil their economic and environmental roles. Proceedings of the 34th meeting of the EUCARPIA Fodder Crops and Amenity Grasses Section in cooperation with the EUCARPIA Festulolium Working Group, Freising September, 2021, pp. 67–70. https://doi.org/10.5507/vup.21.24459677.16
35. Martin M. Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet J. 2011;17(1):10–2.
36. Bickhart DM, Rosen BD, Koren S, Sayre BL, Hastie AR, Chan S, Lee J, Lam ET, Liachko I, Sullivan ST, et al. Single-molecule sequencing and chromatin conformation capture enable de Novo reference assembly of the domestic goat genome. Nat Genet. 2017;49(4):643–50.
37. Langmead B, Salzberg SL. Fast gapped-read alignment with bowtie 2. Nat Methods. 2012;9(4):357–9.
38. Li H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. Bioinformatics. 2011;27(21):2987–93.
39. Gibson G. Rare and common variants: twenty arguments. Nat Rev Genet. 2012;13(2):135–45.
40. Hoffmann TJ. Witte. Strategies for imputing and analyzing rare variants in association studies. Trends Genet. 2015;31(10):556–63.
41. DaCosta JM, Sorenson MD. Amplification biases and consistent recovery of loci in a double-digest RAD-seq protocol. PLoS ONE. 2014;9(9):e106713.
42. Torkamaneh D. Scanning and filling: ultra-dense SNP genotyping combining genotyping-by-sequencing, SNP array and whole-genome resequencing data. PLoS ONE. 2015;10:7.

43. Ashby R, Brauning R, Baird B, Jauregui R, Vallender M, Laugraud A et al. GBSathon: benchmarking GBS analysis workflows through comparison with SNP chip and pedigree data. 2019. https://doi.org/10.13140/RG.2.2.19098.18884

Publisher's note