Pérez-Enciso *et al. Genet Sel Evol* (2020) 52:7
https://doi.org/10.1186/s12711-020-0530-2

GSE Genetics Selection Evolution

SOFTWARE Open Access

SeqBreed: a python tool to evaluate genomic prediction in complex scenarios

Miguel Pérez-Enciso[1,2*], Lino C. Ramírez-Ayala[1] and Laura M. Zingaretti[1,3]

Abstract

Background: Genomic prediction (GP) is a method whereby DNA polymorphism information is used to predict breeding values for complex traits. Although GP can significantly enhance predictive accuracy, it can be expensive and difficult to implement. To help design optimum breeding programs and experiments, including genome-wide association studies and genomic selection experiments, we have developed SeqBreed, a generic and flexible forward simulator programmed in python3.

Results: SeqBreed accommodates sex and mitochondrion chromosomes as well as autopolyploidy. It can simulate any number of complex phenotypes that are determined by any number of causal loci. SeqBreed implements several GP methods, including genomic best linear unbiased prediction (GBLUP), single-step GBLUP, pedigree-based BLUP, and mass selection. We illustrate its functionality with Drosophila genome reference panel (DGRP) sequence data and with tetraploid potato genotype data.

Conclusions: SeqBreed is a flexible and easy to use tool that can be used to optimize GP or genome-wide association studies. It incorporates some of the most popular GP methods and includes several visualization tools. Code is open and can be freely modified. Software, documentation, and examples are available at https://github.com/miguelperezenciso/SeqBreed.

Background

Genomic prediction (GP) is a method whereby DNA polymorphism information is used to predict the breeding value of individuals for complex traits. The availability of high-throughput single nucleotide polymorphism (SNP) genotyping in a cost-effective manner has led GP to become a standard tool in the analysis and improvement of complex traits [1]. GP has revolutionized breeding programs in plants and animals and, today, GP methods are also widely used in human genetics or ecology. Nevertheless, GP is more expensive than traditional pedigree-based breeding. GP can be difficult to implement in practical scenarios, due in part to the difficulty of optimizing genotyping strategies and to uncertainty about the genetic basis of complex traits. Thus, it is highly advisable to evaluate its potential advantages and expected performance in advance. GP accuracy depends on a large number of factors. Several of these can be controlled by the practitioner, to some extent, such as the number of SNPs, number of individuals, selection intensity, and the evaluation method. Other factors cannot be modified, such as linkage disequilibrium and are even unknown (genetic architecture). Although several approximations of the accuracy of GP have been developed, e.g. [2, 3], it remains difficult to analytically assess the influence of these factors in practical scenarios across generations. For this purpose, stochastic computer simulation is the most reliable option. Although critical factors such as the detailed genetic architecture of complex traits are unknown, the main genetic parameters are reasonably well known for most complex traits, such as heritability and the distribution of genetic effects, which can

*Correspondence: miguel.perez@uab.es
[1] Centre for Research in Agricultural Genomics (CRAG), CSIC-IRTA-UAB-UB, 08193 Bellaterra, Barcelona, Spain
Full list of author information is available at the end of the article

BMC

© The Author(s) 2020. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

be approximated by a gamma distribution [4–6]. Thus, a simulation study can be performed to evaluate the effect of the number of causal loci quantitative trait nucleotides (QTN) and of their location to assess the robustness of predictions.

Here, we present a versatile python3 forward simulation tool, SeqBreed, to evaluate GP performance in generic scenarios and with any genetic architecture (i.e., number of QTN, their effects and location, and the number of traits). The purpose of SeqBreed is to generate phenotype and genotype data of individuals under different (genomic) selection strategies. SeqBreed is inspired by a previous pSBVB fortran software program [7], but the code has been rewritten in python3 and many new options have been added. Python can be much slower than compiled languages, but is much easier and friendlier to use, allowing direct interaction with the user to, e.g., make plots or control selection and breeding decisions. In addition, many libraries in python, such as 'numpy' (https://numpy.org/) or 'pandas' (https://pandas.pydata.org/), are wrappers on compiled languages, such that careful programming significantly alleviates the limited speed of native python. Thus, SeqBreed is much more versatile than pSBVB and incorporates many new options, such as genome-wide association studies (GWAS) and principal component analysis (PCA). Most importantly, it allows automatic implementation of standard genomic selection procedures. Usage details and the main features of SeqBreed are described in the following and in the accompanying GitHub site https://github.com/miguelperezenciso/SeqBreed.

Implementation

Outline

Broadly, SeqBreed takes genotype/sequence data from a founder population and simulates phenotypes according to a predetermined genetic architecture. Offspring genomes and phenotypes can be simulated under selection or random drift. By default, selection is simulated across a predetermined number of generations and selection intensities. SeqBreed offers extensive flexibility to the user. For example, accuracy of GP with several SNP arrays can be simultaneously compared using the same data; offspring of specific pairs of parents can be generated; and dihaploid offspring can be simulated. SeqBreed can be run using scripts or interactively, where the user can, say, obtain plots for each generation or generate genotype data of a given set of individuals. Examples of the program's usage are in the GitHub's jupyter notebook https://github.com/miguelperezenciso/SeqBreed/blob/master/SeqBreed_tutorial.ipynb and in the python script https://github.com/miguelperezenciso/SeqBreed/blob/master/main.py.

SeqBreed is programmed in python3 using an object-oriented paradigm. The generic SeqBreed flowchart is visualized in Fig. 1. As input, SeqBreed minimally requires a genotype file from the founder base population in vcf [8] or plink-like format [9]. A typical SeqBreed run consists of the following steps:

1. Upload founder sequence genotypes in vcf or plink format. The program automatically determines ploidy and the number of chromosomes and SNPs.
2. Specify genome characteristics. Sex-linked SNPs and/or recombination rates can be specified.
3. Specify desired heritabilities and causal SNPs (QTN) and their effects for every trait. Environmental variances are inferred given founder genomes, QTN effects and heritabilities.
4. Offspring genomes and phenotypes are simulated by gene-dropping along a predetermined pedigree or by implementing selection.

PCA plots or GWAS options are also implemented. The main python classes are:

- Population: This class contains the main attributes for running selection experiments and is a container for Individual objects. It includes methods to add new individuals generated by mating two parents or randomly shuffling founder genomes in order to increase the number of base population animals (see [10]). It also prints basic population data and summary plots.
- Individual: It allows generation, manipulation, and printing of the genotypes and phenotypes of individuals. Internally, an individual's genome is represented by contiguous non recombining blocks rather than by the list of all SNP alleles, which allows dramatic savings in memory and increases in efficiency (see Figure 1 in Pérez-Enciso et al. [10]).
- Genome: All genome characteristics are stored and can be accessed by methods in this class. It specifies ploidy, number and class of chromosomes, and recombination rates or SNP positions.
- GFounder: SeqBreed requires as minimum input the genotypes of the so-called 'founder population', which comprises the parents of the rest of the individuals to be generated. This class stores these genotypes and automatically retrieves main genome features such as SNP positions, number of chromosomes, etc. Initial genotypes can be filtered by minimum allele frequency (MAF).
- QTN: This class determines the genetic architecture for each trait simulated. It has methods to determine the environmental variance given a desired heritabil-

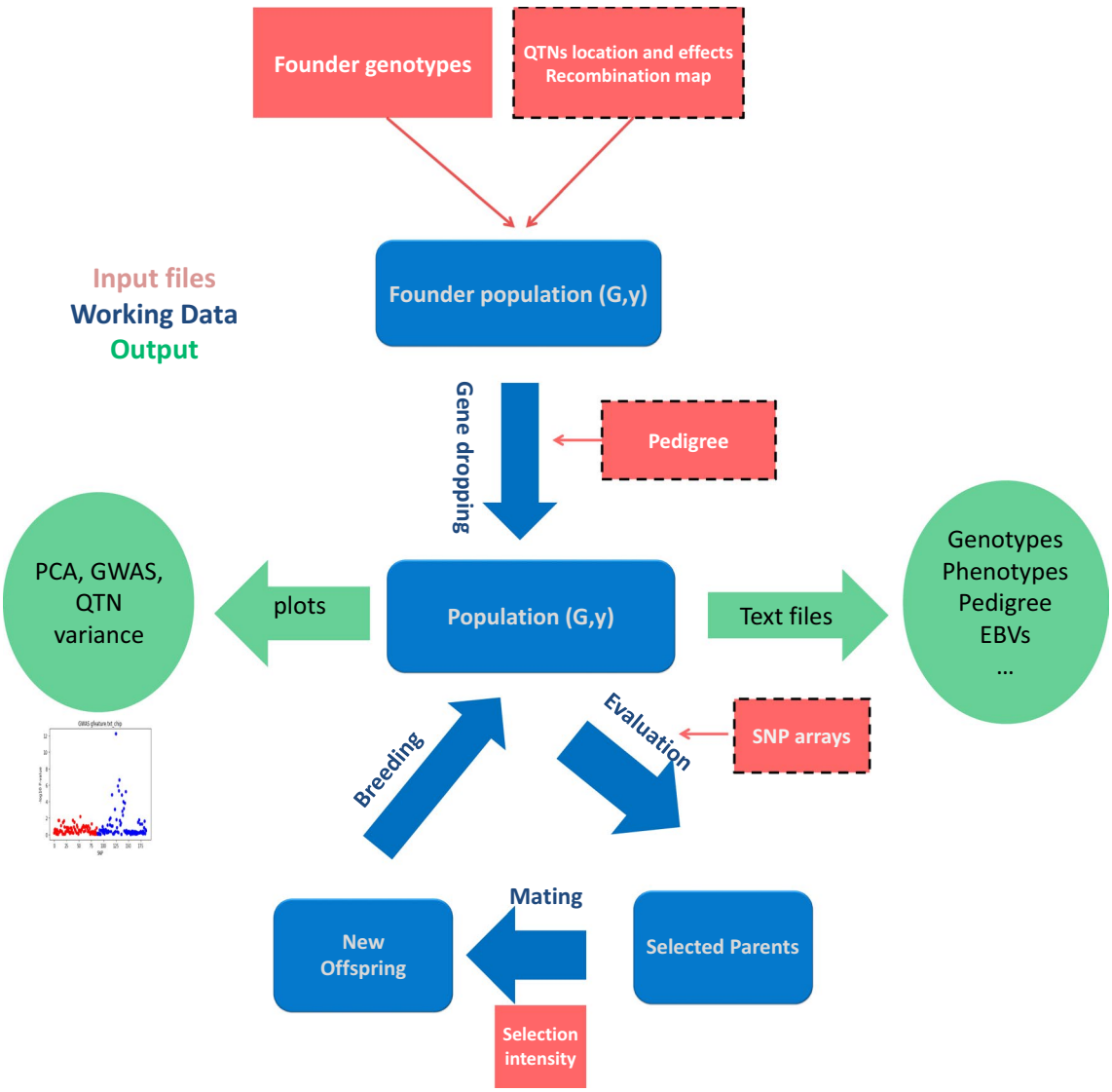

Fig. 1 Outline of the SeqBreed pipeline. Inputs are shown in red squares, dashed border rectangles represent optional input, internal data are in blue rounded squares, main operations are indicated in blue, and outputs are in green circles; G and y refer to genotypes and phenotypes, respectively. The program starts with an optional gene dropping step following an input pedigree. No selection is performed at this stage. The bottom loop represents selection, where new offspring are generated based on the genotypes of selected parents. A list of SNPs in the genotyping array must be determined when using GBLUP and BLUP. A new cycle starts when these new offspring are added to the existing population. Plots can be performed at several stages

ity, and to plot variance components for QTN. In its current version, SeqBreed allows for dominance and additive actions, but not epistasis.

- Chip: This class is basically a container for the list of SNPs that are included in a genotyping array. It allows easy comparison of different genotyping strategies in genomic selection.

Specifying genome features and genetic architecture

By default, SeqBreed assumes that all loci are autosomal and a recombination rate of 1 cM = 1 Mb throughout the genome. It includes options to specify sex or mitochondrial chromosomes, and local and sex specific recombination maps. A pseudo-autosomal region (PAR) is not accommodated for sex chromosomes, i.e., the whole Y chromosome is assumed to be non-recombining. A

mitochondrial chromosome is a non-recombining chromosome that is transmitted maternally. SeqBreed allows for autopolyploidy of any level, which is automatically detected from vcf files. Accurate modeling of meiosis in polyploids is notoriously difficult [11, 12] and SeqBreed implements a simplified algorithm:

1. For each chromosome id, homologs are randomly paired.
2. Within each pair of homologs, cross-over events are generated as for diploids, i.e., no interaction between homologous chromosomes is modeled and the number of cross-over events is simulated following a Poisson distribution with a rate equal to chromosome length in Morgans.
3. Sex chromosomes are modeled with a maximum ploidy of 2.

Therefore, our algorithm does not fully model the interaction of preferential pairing of homologous chromosomes and double reduction arising from multivalent formation [13]. For the purposes of this software (i.e. comparison of GP strategies over a limited number of generations), it is unlikely that this approximation has a dramatic effect.

SeqBreed allows the simulation of any number of phenotypic traits, regardless of ploidy. For each trait, broad-sense heritability must be specified. There are three options to specify the number of QTN and their effects (https://github.com/miguelperezenciso/SeqBreed#3-specifying-genetic-architecture): (i) a random number of QTN positions are sampled genome-wide and additive effects are sampled from a gamma distribution Γ (shape = 0.2 and scale = 5), as suggested by Caballero et al. [5]; (ii) the positions of the QTN are specified in a file and additive effects are sampled from a gamma distribution; and (iii) QTN positions and additive and dominant effects for each trait are specified in an external file. By default, QTN are not removed from the sequence data to perform genomic evaluation. To remove QTN from evaluation, a SNP chip can be defined that excludes the QTN. Options (i) and (ii) can only be used with one trait and without dominance. SeqBreed adjusts the environmental variance $Var(e)$ to retrieve the desired broad-sense heritabilities (H^2) from $Var(e) = Var(g) \times (1 - H^2)/H^2$, where $Var(g)$ is the variance of the genotypic values of individuals in the founder population. The genotypic value for individual i is defined as:

$$g_i = \sum_{j=1}^{nQTN} \gamma_{ij} a_j + \sum_{j=1}^{nQTN} \delta_{ij} d_j,$$

where $nQTN$ is the number of QTN, a_j is the additive effect of the j-th QTN, that is, half the expected difference between homozygous genotypes, with γ_{ij} taking the values -1, 0 and 1 for homozygous, heterozygous, and alternative homozygous genotypes, respectively, d_j is the dominance effect of the j-th QTN, with δ_{ij} taking the value 1 if the genotype is heterozygous and 0 otherwise. In the case of polyploids:

$$g_i = \sum_{j=1}^{nQTN} \eta_{ij} a_j + \sum_{j=1}^{nQTN} \varphi_{ij} d_j,$$

where η_{ij} is the number of copies of the alternative allele (coded as 1) minus half the ploidy for the j-th QTN and the i-th individual, and a_j is therefore the expected change in phenotype per copy of allele '1' at the j-th QTN. In polyploids, technically as many dominance coefficients as ploidy levels (h) minus two can be defined, which is not practical. As in pSBVB [7], we define φ_{ij} as the minimum number of copies of allele '1' such that the expected phenotype is d (see Figure 1 in Zingaretti et al. [7]). SeqBreed uses $\varphi_{ij} = 1$, that is, all heterozygous individuals have the same genotype value as the complete homozygous '1'. SeqBreed computes genotypic values for each individual and simulate phenotypes from $y_i = \mu + g_i + e_i$, where μ is a constant and e is a normal deviate $e \sim N(0, Var(e))$.

For multiple traits, the user needs to specify additive and dominant QTN effects separately for each trait. This is done via an external text file, where additive and dominant QTN effects are specified for each trait (option 3 in https://github.com/miguelperezenciso/SeqBreed#3-specifying-genetic-architecture). There is no specific assumption on genetic correlations between traits. To simulate no pleiotropy, QTN for each trait have zero effects for all other traits. Note that this does not prevent a non-zero genetic correlation arising from linkage disequilibrium. The program does not automatically adjust desired genetic correlations and, thus, different QTN values may need to be tested to fit desired correlations. Environmental correlations are always zero.

It is typically difficult to find real sequence data to generate a reasonably sized founder population. To accommodate this, SeqBreed can generate 'dummy' founder individuals by randomly combining recombinant haplotypes. This can be done in two ways, either by generating a random pedigree and simulating a new founder individual by gene-dropping along this pedigree, or by directly simulating a number of recombining breakpoints and assigning random founder genotypes to each block between recombination breakpoints (https://github.com/miguelperezenciso/SeqBreed/blob/master/README.md#breeding-population).

Gene dropping and selection implementation

Seqbreed can be run along a predetermined pedigree or using a combination of options (several examples are provided in the GitHub site). It is also possible to generate new individuals interactively, including dihaploids. To speed up computations and avoid unnecessary memory usage, only recombination breaks and ancestor haplotype ids are stored for each individual (see Figure 1 in [14]).

SeqBreed allows computing estimated breeding values using several GP methods. It also allows several lists of SNPs (SNP chips) to be defined, such that GP performance can be easily compared across chips. From a methodological point of view, most GP implementations are based on penalized linear methods (e.g., de los Campos et al. [15]). SeqBreed includes some of the most popular GP options, including pedigree BLUP [16], GBLUP [17], and single-step GBLUP [18]. Only single trait GP algorithms are implemented so far. Mass selection is also implemented. For GBLUP and single-step GBLUP, the genomic relationship matrix **G** is obtained using VanRaden [17] as:

$$\mathbf{G} = \frac{\mathbf{X}\mathbf{X}'}{2\sum_{j=1}^{nSNP} p_j(1-p_j)},$$

where **X** is a $N \times n$SNP matrix containing genotypes (coded 0,1,2 deviated from the mean) for SNPs on the chip, p_j is the allele frequency of the j-th SNP, N is the number of individuals, and nSNP is the number of SNPs. To avoid potential singularity problems, diagonal elements of **G** are multiplied by 1.05. SeqBreed requires that heritabilities to be used in BLUP or GBLUP are provided (i.e., they are not estimated). The program allows the incorporation of other custom GP methods based on a user python function or by exporting SNP data and phenotypes from SeqBreed, running a genetic evaluation, externally and then importing the resulting estimated breeding values.

Selection can be automatically configured and run, as documented in the GitHub examples (https://github.com/miguelperezenciso/SeqBreed). Running a selection scheme requires specifying the number of generations, the numbers of females and males to be selected, and the number of offspring per female. SeqBreed splits the selection process in three steps, which allows a fine control over the breeding program. First, breeding values are predicted using the chosen evaluation method and marker information. By default, the data from all individuals across the current and previous generations are used, but this can be changed by specifying the subset of individuals to be used. Second, a function is used to generate offspring from selected parents. This function requires specifying the candidates for selection (allowing for continuous or discrete generations), selection intensity, family size, and either assortative or random mating between selected parents. Hierarchical mating between females and males is employed by default (https://github.com/miguelperezenciso/SeqBreed#7-implementing-selection). Assortative and random mating schemes are implemented; more sophisticated mating schemes, such as those based on optimal contributions [19, 20], have to be specified manually by modifying the function 'ReturnNewPed' in the selection module (https://github.com/miguelperezenciso/SeqBreed/blob/master/src/selection.py).

Visualization

A novel feature of SeqBreed, as compared to our previous software pSBVB, is the capability of graphical outputs. Figure 2 illustrates some of the plots that can be performed automatically. Figure 2a shows the results of the QTN.plot() function, which plots the individual QTN variance as a function of MAF, the histogram of QTN variances, and the cumulative variance when QTN are sorted by MAF. This is performed for each phenotype and for both additive and dominance variances, based on allele substitution effects $\alpha = a + d(1-2p)$, where p is the minimum allele frequency, and assuming complete equilibrium. In addition, PCA plots using all sequence or custom defined SNP sets (Fig. 2b) are available, as well as GWAS plots showing p-values or false discovery rate (FDR) values (Fig. 2c). Genotype and phenotype data can also be exported in text files.

Usage and examples

The basic functioning of SeqBreed is illustrated by the main.py script that is available at https://github.com/miguelperezenciso/SeqBreed/blob/master/main.py. This script, or its equivalent jupyter notebook (SeqBreed_tutorial.ipynb), shows the basic commands to run SeqBreed and import the required modules. First, SeqBreed modules are imported as:

```
from SeqBreed import genome as gg
from SeqBreed.selection import selection as sel
```

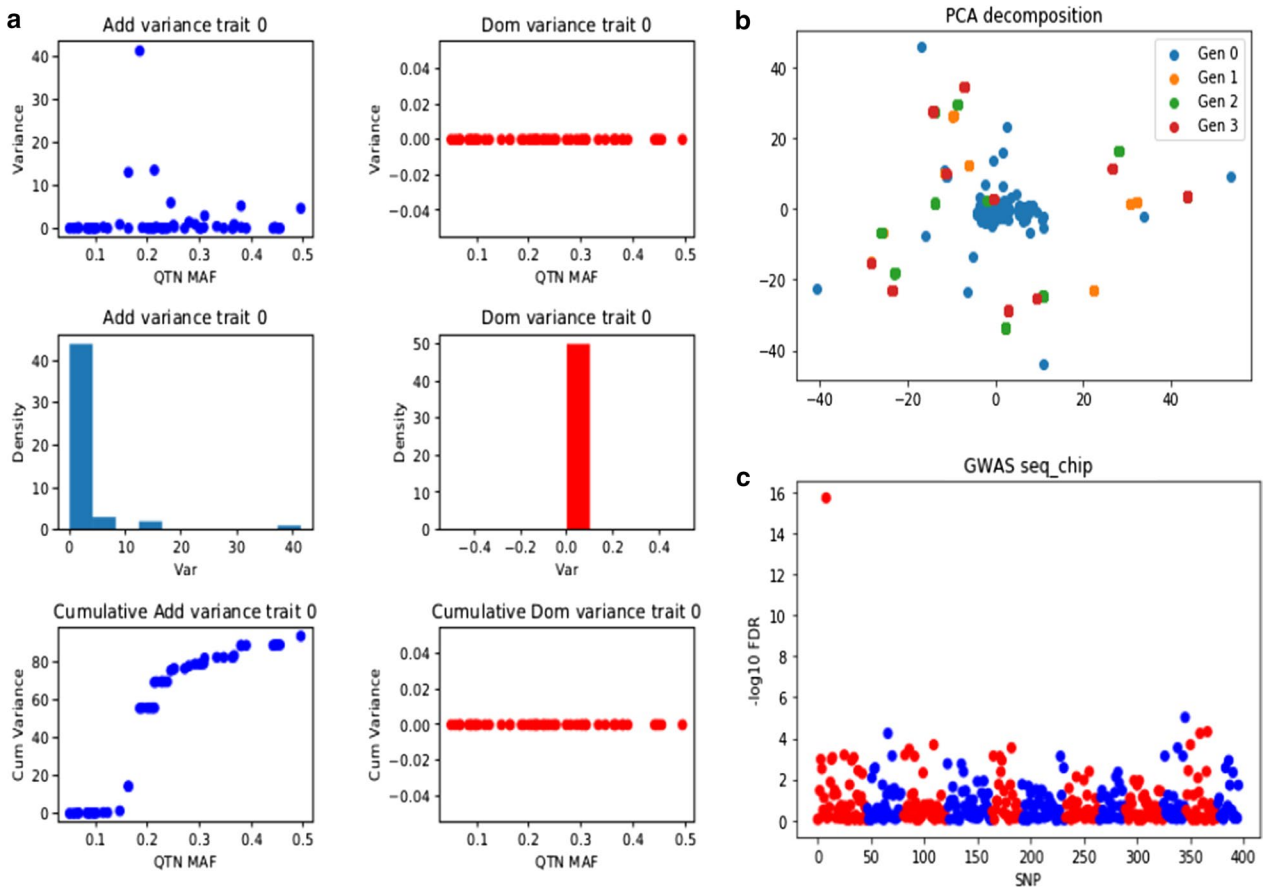

Fig. 2 Example plots produced by SeqBreed. **a** Contribution of each QTN to total variance. Top, individual QTN variances as a function of minimum allele frequency (MAF); middle, histogram of QTN variances; bottom, cumulative variance when QTN are sorted by MAF. In blue, additive variances; in red, dominance variances. The figure shows a fully additive phenotype, thus dominance variances are zero. **b** Principal component analysis plot; individuals of different generations are in different colors. **c** Genome-wide association study showing false discovery rate values (− log10 scale). SNPs on different chromosomes are represented in alternate colors

Founder population genotypes are uploaded from the vcf file, using the command:

```
gbase = gg.GFounder(vcfFile=vcffile, snpFile=seqfile)
```

which generates a GFounder object that contains founder genotypes, vcffile is the file containing genotypes and seqfile is generated by the program and contains information about SNP positions, which are used in the next step.

Next, the main genome features are specified. The following command creates a Genome object that assumes that the 'X' chromosome is the sex X chromosome, while SNPs on the chromosome named 'MT' are mitochondrial.

```
# seqfile is a file obtained in previous step
gfeatures = gg.Genome(snpFile=seqfile, mapFile=mapfile, ploidy=gbase.ploidy, XChr='X',
MTChr='MT')
```

Genetic architecture can be specified in different ways. The simplest is to generate nqtn QTN randomly distributed along the genome with effects sampled from a gamma distribution, where h2 is the desired heritability.

```
# 10 QTNs are simulated, h2 of the trait is 0.7
qtn = gg.QTNs(h2=[0.7], genome=gfeatures, nqtn=10)
# environmental variances are computed
qtn.get_var(gfeatures, gbase)
```

Selection is implemented in cycles, the number of generations, the numbers of males and females selected, and family size must be specified. The following is an example with GBLUP selection, random mating, and continuous generations.

```
ngen = 5          # no. of selection generations
nsel = [5, 10]    # no. of males and females selected
noffspring = 10 # no. offspring per female

# selection cycles
for t in range(ngen):
   # STEP 0: generate marker data for evaluation, stored in X matrix
   # pop is a Population object containing individual genomes and phenotypes
   # chip0 is a Chip object containing SNPs to be used in genomic evaluation
   X = gg.do_X(pop.inds, gfeatures, gbase, chip0)

   # STEP 1: estimate breeding values using criterion GBLUP, assuming h2=0.3
   # criterion can take values 'random', 'phenotype', 'blup', 'gblup' or 'sstep'
   sel.doEbv(pop, criterion='gblup', X=X, h2=0.3, nh=gfeatures.ploidy)

   # STEP 2: pedigree with offspring of selected individuals
   # mating can be ‘assortative’ or ‘random’
   # generation indicates that individuals from generation onwards are considered as selection
candidates
   newPed = sel.ReturnNewPed(pop, nsel, famsize=noffspring, mating='random',  generation=0)

   # STEP 3: generates new offspring (this function adds QTN genotypes, true bvs and
phenotypes)
   # New individuals are added to current Population
   pop.addPed(newPed, gfeatures, qtn, gbase)
```

We illustrate the software with sequence data from the Drosophila genome reference panel (DGRP, [21]), parsed and filtered as explained in [22], and genotype data from tetraploid potato [23], parsed as described in [7]. Data and scripts are in https://github.com/miguelperezenciso/SeqBreed/tree/master/DGRP and in https://github.com/miguelperezenciso/SeqBreed/tree/master/POTATO for the Drosophila and potato examples, respectively. The DGRP scripts illustrate the specific recombination map of Drosophila, where males do not recombine, as shown

in the 'dgrp.map' file. The example provided in GitHub consists of a small experiment to compare genomic and mass selection. Plots in the jupyter notebook are implemented to track phenotypic changes by generation. The potato scripts illustrate how to generate an F2 cross between extreme lines and to perform a GWAS experiment in polyploids. GWAS results using PCA corrected phenotypes are also shown.

Conclusions and future developments

Several other programs have been developed for similar purposes as SeqBreed, including our own pSBVB [7], AlphaSim [24] and its successor AlphaSimR (https://alphagenes.roslin.ed.ac.uk/wp/software-2/alphasimr/), PedigreeSim [13], simuPOP [25], and QMSim [26]. However, SeqBreed offers a unique combination of features for simulation of GP of complex traits, including built-in implementation of several GP methods, the possibility of simulating polyploid genomes, and several options to specify QTN and SNP arrays. It also allows new individuals to be generated interactively and provides graphical plots of results. It is easy to use, easy to install, and software options are illustrated with several examples in the GitHub site. Given the interactive nature of python and its graphical features, SeqBreed is especially suited for educational purposes. However, for large-scale simulations SeqBreed will not be as efficient as some Fortran counterparts such as AlphaSim or pSBVB.

Note that SeqBreed was designed to evaluate the performance of GP or GWAS over a short time horizon, i.e., new mutations are not generated. SeqBreed is not designed to investigate the long-term effects of demography or selection on DNA variability because new mutations are not generated. For these purposes, Slim [27] or similar tools are more appropriate. To investigate realistic scenarios, the recommended input for SeqBreed is real sequence data.

Plans for further development of SeqBreed include additional features to generalize available genetic architectures (e.g., imprinting, epistasis), integration with machine-learning tools (scikit, keras) for genetic evaluation, development of an educational tool with an html-based interface, and improving output and plotting features.

Authors' contributions

MPE conceived research. MPE and LMZ wrote software and documentation. MPE, LMZ and LCRA tested and validated the program. All authors read and approved the final manuscript.

Funding

This work was supported by a PhD grant from the Ministry of Economy and Science (MINECO, Spain) to LMZ, by MINECO grant AGL2016-78709-R and from the EU through the BFU2016-77236-P (MINECO/AEI/FEDER, EU) to MPE and the "Centro de Excelencia Severo Ochoa 2016–2019" award SEV-2015-0533. LCRA is funded by "Don Carlos Antonio López" Graduate program (BECAL) from Paraguay.

Availability of data and materials

https://github.com/miguelperezenciso/SeqBreed

Availability and requirements: Project name: SeqBreed. Project home page: https://github.com/miguelperezenciso/SeqBreed. Operating systems: Tested in linux and mac. It should also run in windows python. Programming language: Python. License: GNU GPLv3. Any restrictions to use by non-academics: None.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Centre for Research in Agricultural Genomics (CRAG), CSIC-IRTA-UAB-UB, 08193 Bellaterra, Barcelona, Spain. [2] ICREA, Passeig de Lluís Companys 23, 08010 Barcelona, Spain. [3] Universidad Nacional de Villa María, IAPBCyA-IAPCH Villa María, Córdoba, Argentina.

Received: 27 August 2019 Accepted: 29 January 2020
Published online: 10 February 2020

References

1. Meuwissen T, Hayes B, Goddard M. Accelerating improvement of livestock with genomic selection. Annu Rev Anim Biosci. 2013;1:221–37.
2. Daetwyler HD, Villanueva B, Woolliams JA. Accuracy of predicting the genetic risk of disease using a genome-wide approach. PLoS One. 2008;3:e3395.
3. Goddard M. Genomic selection: prediction of accuracy and maximisation of long term response. Genetica. 2009;136:245–57.
4. Hayes B, Goddard ME. The distribution of the effects of genes affecting quantitative traits in livestock. Genet Sel Evol. 2001;33:209–29.
5. Caballero A, Tenesa A, Keightley PD. The nature of genetic variation for complex traits revealed by GWAS and regional heritability mapping analyses. Genetics. 2015;201:1601–13.
6. Eyre-Walker A, Keightley PD. The distribution of fitness effects of new mutations. Nat Rev Genet. 2007;8:610–8.
7. Zingaretti ML, Monfort A, Pérez-Enciso M. pSBVB: a versatile simulation tool to evaluate genomic selection in polyploid species. G3 (Bethesda). 2019;9:327–34.
8. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The sequence alignment/map format and SAMtools. Bioinformatics. 2009;25:2078–9.
9. Chang CC, Chow CC, Tellier LC, Vattikuti S, Purcell SM, Lee JJ. Second-generation PLINK: rising to the challenge of larger and richer datasets. Gigascience. 2015;4:7.
10. Pérez-Enciso M, Forneris N, de Los Campos G, Legarra A. Evaluating sequence-based genomic prediction with an efficient new simulator. Genetics. 2017;205:939–53.
11. Baduel P, Bray S, Vallejo-Marin M, Kolář F, Yant L. The, "Polyploid Hop": Shifting challenges and opportunities over the evolutionary lifespan of genome duplications. Front Ecol Evol. 2018;6:117.
12. Jighly A, Lin Z, Forster JW, Spangenberg GC, Hayes BJ, Daetwyler HD. Insights into population genetics and evolution of polyploids and their ancestors. Mol Ecol Resour. 2018;18:1157–72.
13. Voorrips RE, Maliepaard CA. The simulation of meiosis in diploid and tetraploid organisms using various genetic models. BMC Bioinformatics. 2012;13:248.
14. Pérez-Enciso M, Varona L, Rothschild MF. Computation of identity by descent probabilities conditional on DNA markers via a Monte Carlo Markov Chain method. Genet Sel Evol. 2000;32:467–82.

15. de los Campos G, Hickey JM, Pong-Wong R, Daetwyler HD, Calus MPL. Whole-genome regression and prediction methods applied to plant and animal breeding. Genetics. 2013;193:327–45.
16. Henderson CR. Applications of linear models in animal breeding. Guelph: University of Guelph; 1984.
17. VanRaden PM. Efficient methods to compute genomic predictions. J Dairy Sci. 2008;91:4414–23.
18. Legarra A, Aguilar I, Misztal I. A relationship matrix including full pedigree and genomic information. J Dairy Sci. 2009;92:4656–63.
19. Sánchez L, Bijma P, Woolliams JA. Minimizing inbreeding by managing genetic contributions across generations. Genetics. 2003;164:1589–95.
20. Sonesson AK, Meuwissen THE. Mating schemes for optimum contribution selection with constrained rates of inbreeding. Genet Sel Evol. 2000;32:231–48.
21. Huang W, Massouras A, Inoue Y, Peiffer J, Ràmia M, Tarone AM, et al. Natural variation in genome architecture among 205 *Drosophila melanogaster* genetic reference panel lines. Genome Res. 2014;24:1193–208.
22. Forneris NS, Vitezica ZG, Legarra A, Pérez-Enciso M. Influence of epistasis on response to genomic selection using complete sequence data. Genet Sel Evol. 2017;49:66.
23. Enciso-Rodriguez F, Douches D, Lopez-Cruz M, Coombs J, de Los Campos G. Genomic selection for late blight and common scab resistance in tetraploid potato (*Solanum tuberosum*). G3. (Bethesda). 2018;8:2471–81.
24. Faux A-M, Gorjanc G, Gaynor RC, Battagin M, Edwards SM, Wilson DL, et al. AlphaSim: software for breeding program simulation. Plant Genome. 2016;9:1–14.
25. Peng B, Kimmel M. simuPOP: a forward-time population genetics simulation environment. Bioinformatics. 2005;21:3686–7.
26. Sargolzaei M, Schenkel FS. QMSim: a large-scale genome simulator for livestock. Bioinformatics. 2009;25:680–1.
27. Messer PW. SLiM: simulating evolution with selection and linkage. Genetics. 2013;194:1037–9.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions