pSBVB: polyploid Sequence Based Virtual Breeding

A flexible gene dropping algorithm to simulate sequence based population data and complex traits in polyploids

Miguel Pérez-Enciso,
(miguel.perez@uab.es)

With collaborations from L. Zingaretti, N. Forneris, G. de los Campos and A. Legarra

Purpose

pSBVB is a sequence-based population simulator that can accommodate polyploid genomes. It is based on SBVB (https://github.com/mperezenciso/sbvb0) (Pérez-Enciso et al. 2017) that allows simulating traits of an arbitrary genetic complexity in polyploids. Its goal is to simulate complex traits and genotype data starting with a vcf file that contains the genotypes of founder individuals and following a given pedigree. The main output are the genotypes of all individuals in the pedigree and/or molecular relationship matrices (GRM) using all sequence or a series of SNP lists, together with phenotype data. The program implements very efficient algorithms where only the recombination breakpoints for each individual are stored, therefore allowing the simulation of thousands of individuals very quickly. The vcf file cannot contain missing genotypes and is assumed to be phased. A python, easier to use version, is currently under development. Manual: Link to full html help: https://lauzingaretti.github.io/pSBVB/.

Main features

- ✓ Any number of traits.
- ✓ Any number of QTNs, trait specific.
- ✓ Either auto or alloploid genomes are allowed.
- ✓ Specific modeling for additive and dominant coefficients in polyploids.
- ✓ Molecular relationship matrices specifically tailored for polyploids
- ✓ Can generate correlated allelic effects and frequencies.
- ✓ Efficient algorithms to generate haplotypes and sample SNP genotypes.
- ✓ Computes genomic relationship matrices for any number of SNP arrays simultaneously.
- ✓ Any number of chromosomes, allows for varying local recombination rates.
- ✓ In contrast to SBVB, epistasis cannot be modeled.

Installation

The source code, manual and examples can be obtained from

```
https://github.com/mperezenciso/psbvb
```

To compile:

```
gfortran -O3 kind.f90 ALliball.f90 aux_sub11.f90 sbvb.f90 -o sbvb -lblas
```

or

```
make
```

To install in /usr/local/bin

```
sudo make install
```

The program requires `blas` libraries, but these are standard in any Unix or OS mac system. We have tested pSBVB only in linux with gfortran compiler; intel ifort seems not working, but gfortran in mac OS looks ok. To Ubuntu installing:

```
sudo apt-get install BLAS
```

Or a more minimalist set of packages can be installed with:

```
sudo apt-get install libblas-dev liblapack-dev
```

Usage

To run (assuming vcf is compressed):

```
zcat file.vcf.gz | perl vcf2tped2.pl -hap | cut -d ' ' -f 1,4- | \
sbvb -i sbvb.par
```

Where sbvb.par is the parameter file (details follow). The intermediate steps are simply for SBVB to read genotypes in suitable format, that is,

```
allele1_snp1_ind1 allele2_snp1_ind1 allele1_snp1_ind2 allele2_snp1_ind2 …
allele1_snp2_ind1 allele2_snp2_ind1 allele1_snp2_ind2 allele2_snp2_ind2 …
```

with alleles coded as 0/1. To run SBVB with the same random seed:

```
… | sbvb -i sbvb.par -seed iseed
```

where `iseed` is an integer number.

Parameter file

The parameter file controls all pSBVB behavior. It consists of a list of sections in UPPER CASE (in any order) followed in the next line by the required data, e.g.,

```
QTNFILE
sbvb.qtl
```

tells the program that QTN specifications are in `sbvb.qtl` file. Comments can be mixed starting with # or ! A full list of options in the parameter file is in APPENDIX. In the following, we list the main ones.

Specifying genetic architecture

If more than one trait is generated, then use

```
NTRAIT
ntraits
```

in the parameter file. Otherwise this section is not needed. SBVB requires the user to provide the list of causal SNPs (QTNs) as specified in QTNFILE section. The format of the QTN file is

```
ichr  ipos
```

or

```
ichr  ipos  add_eff_T1  add_eff_T2  …  add_Eff_Tn
```

or

```
ichr  ipos  add_eff_T1  dom_eff_T1  add_eff_T2  dom_eff_T2  …  add_eff_Tn  dom_eff_Tn
```

separated by spaces, and where ichr is chromosome and ipos is position in base pair, add_eff is additive effect (ie, half the difference between 11 and 00 homozygotes), and dom_eff is the heterozygous effect.

For polyploids, the add_eff specifies the change in phenotype by each copy of the alternative allele (see below). For several traits, first are printed all add effects for every trait, next add+dom.

WARNING: QTN position must coincide with one SNP position in the vcf file, otherwise it is not considered.

If QTN effects are not provided, they can be simulated specifying

```
QTNDISTA

u lower_bound upper_bound | n mu var | g s b
```

and

```
QTNDISTD
u lower_bound upper_bound | n mu var | g s b
```

In parameter file. Where 'u' means effects are sampled from a uniform $U(lower\_bound, upper\_bound)$, 'n' from a normal distribution $N(mu, \mathrm{var})$ and 'g' from a gamma $\gamma(s, b)$. For a gamma distribution, the probability p that a derived allele decreases the phenotype can be specified as.

```
PSIGNQTN
p
```

The default value is 50%. By default, effects are sampled independently of frequency, but it is possible to generate a correlation (rho) with

```
RHOQA
rho
```

This option can be useful to simulate past selection (see Pérez-Enciso et al. doi: 10.1534/genetics.116.194878). The narrow sense heritability is specified as

```
H2
h2
```

or alternatively, the broad sense heritability (using `H2G`). **Only the genotypes from the base population (in the vcf file) are used to adjust heritability.** The phenotype of individual *i* (y_i) is simulated from

$$y_i = \mu + \sum_{j=1}^{Q} \gamma_{ij} a_j + \sum_{j=1}^{Q} \delta_{ij} d_j + \varepsilon_i, \qquad (1)$$

where μ is the general mean, a_j is the additive effect of *j*-th locus, that is, half the expected difference between homozygous genotypes, γ_{ij} takes values -1, 0 and 1 for homozygous, heterozygous and alternative homozygous genotypes, respectively, d_j is the dominance effect of *j*-th locus, and δ_{ij} takes value 1 if the genotype is heterozygous, 0 otherwise, and ε_i is a normal residual of the *i*- observation.

For polyploids, the equivalent equation can be expressed as:

$$y_i = \mu + \sum_{j=1}^{Q} \eta_{ij} a_j + \sum_{j=1}^{Q} \varphi_{ij} d_j + \varepsilon_i, \qquad (2)$$

where η_{ij} is the number of copies of the alternative allele (coded say as 1) minus half the ploidy ($h/2$) for *j*-th locus and *i*-th individual, and a_j is therefore the expected change in phenotype per copy of allele '1' in the *j*-th locus. In polyploids, as many dominance coefficients as ploidy level (h) minus two can technically be defined. However, this results in an over-parameterized model that is of no practical use. Here instead we define the φ_{ij} parameter as the minimum number of copies of allele 1 such that the expected phenotype is d. By default, pSBVB uses $\varphi_{ij} = 1$, that is, any genotype having at least one allele '1' and '0' has the expected phenotypic value *d*. Finally, ε_i the residual is sampled from a $N(0, v_\varepsilon)$ where v_ε is adjusted given either H2 or H2G using the genotypes from the base population.

For multiple traits, values in sections `H2` or `H2G`, `RHOQA`, and `QTLDISTA` and `QTLDISTD` must be repeated, e.g., for two traits:

```
H2
0.5
0.23

RHOQA
```

```
0
-0.4

QTNDISTA
u  -0.2 0.2
g    1  0.5
```

Polyploid vcf file (VCFFILE)

In the polyploid vcf file, alleles are listed in order according to the homeolog group where the allele is segregating. An octoploid genotype 0|0|1|0|0|0|0|0 means that all alleles positions are coded as '0' except that in the first haplotype of second homeolog pair.

Recombination in polyploids

The ploidy level must be specified with section

```
PLOIDY
h
```

in the parameter file. By default, pSBVB assumes autopolyploidy and permits recombination between each homeolog chromosome pair with equal probability. Strict alloploidy is specified with

```
ALLOPLOIDY
```

Intermediate rates of recombination between homeolog pairs can be specified with

```
RHOPLOIDY
rho_elements
```

where rho_elements is a matrix of $h \times h$ elements specifying the probability of recombination between *i* and *j* homeologs. The diagonal (p of recombining with itself) is set to 0 by the program.

Pedigree file (PEDFILE)

The format is

```
id   id_father   Id_mother   [sex]
```

where all ids must be consecutive integers, 0 if father or mother unknown, sex is optional (1 for males, 2 for females) and only needed if sex chr is specified. The number of individuals in the vcf file must be specified with section

```
NBASE
nbase
```

in parfile. The pedigree file must contain the first rows as

1 0 0

2 0 0
...
nbase 0 0

that is, those in vcf file are assumed to be unrelated.

Recombination map files

By default, pSBVB assumes a cM to Mb ratio of 1. This ratio can be changed genomewide with `CM2MB` section in the par file. In addition, local recombination rates can be specified with the `MAPFILE` section. The mapfile takes format

```
ichr  last_bp  local_cm2mb
```

where `local_cm2mb` is the recombination rate between last_bp and previous bound (1 bp if first segment), or

```
ichr  last_bp  local_cm2mb_males  Local_cm2mb_females
```

The maximum number of chromosomes allowed by default is 23; should you require more, then section `MAXNCHR` must be included. SBVB permits sex chromosomes, the sex chromosome must be declared with `SEXCHR` section. Then, sex 1 is assumed to be the heterogametic sex, and a sex column should be present in the PEDFILE. Note that this does not apply to polyploids and to plants in general.

WARNING: SEXCHR section with polyploids has not been tested.

WARNING: chromosome ids must be integer consecutive numbers, even for the sex chr if present.

SNP files

SBVB can compute the genomic relationship matrix for all sequence data, and/or specific SNP subsets to mimic different genotyping arrays. Several SNP lists can be analyzed in the same run repeating the SNPFILE section in the par file. Each SNP file has the same format as the QTN file, i.e., chromosome and base pair position.

Molecular relationship files (GRM)

pSBVB may compute GRM. By default, GRM is computed from

$$\mathbf{G} = \frac{\mathbf{M} - h\mathbf{p} \quad \mathbf{M} - h\mathbf{p}^{T}}{h\mathbf{p}(1-\mathbf{p})^{T}}$$

where $\mathbf{M}$ is a n x m matrix with elements m_{ij} containing the number of copies of the alternative allele for i-th individual (i=1..n) and j-th marker (j=1..m), and $\mathbf{p}$ is a m-dimension vector with marker allele frequencies, and h is the ploidy level. Equation above assumes the exact copy number of alleles is known. Since this is unrealistic in many settings, pSBVB may also compute GRMs with two approximations:

1. MIMIC_DIPLOID: The maximum genotype value is set to 2, i.e., only three genotype values can be distinguished, h is set to 2.

2. MIMIC_HAPLOID: Assuming that only one allele can be distinguished for the others, i.e., that a given marker allele behaves as fully dominant. To accommodate this, pSBVB allows computing a modified **G*** where element m_{ij} is coded as 0 if all alleles are 0 and 1 otherwise. The rest of the algorithm proceeds as before, with $h = 1$.

Output

The program writes some general info on the screen, and the following files

- **OUTYFILE** format (contains phenotypes and breeding values):

```
id   Y   (add[i],i=1…ntrait)  ((add+dom)[i],i=1…ntrait)
```

  where add is the first sum in eq. [1] above, dom the second term and epi is the third term. For several traits, first are printed all add effects for every trait, next add+dom, and ending with add+dom+epi.

- **OUTQFILE** format (contains QTN info):

```
ichr    pos  freq_base   freq    (add[i],dom[i];i=1..ntraits)
```

  where ichr is chromosome, pos is QTN bp position, freq_base is frequency in vcf file, freq is frequency along the pedigree, plus additive, dominant effects and add variance ($2pq\alpha^2$) contribution for each locus by trait.

- **OUTGFILE** format (contains GRM, one per SNPFILE plus sequence). A matrix of $n \times n$, where n is the number of individuals in the pedigree. As many outgfiles as snpfiles are written with subscripts .1, .2 etc. .0 corresponds to sequence. To avoid using sequence, NOSEQUENCE in parfile.

- **OUTMFILE** format (contains genotypes for evey SNP file and sequence, in plink format optionally using OUTPLINK in parfile). As many outmfiles as snpfiles are written with subscripts .1, .2 etc. .0 corresponds to sequence. To avoid using sequence, NOSEQUENCE in parfile

Outqfile, outqtn, GRM and marker files are written only if the respective sections OUTQFILE, OUTGFILE and OUTMFILE appear in the par file. Note in particular that OUTMFILE with sequence can be **huge!** To avoid printing sequence info, use

```
NOSEQUENCE
```

in the parfile. To compress marker output, include GZIP in parfile.

Restart the program keeping the same haplotypes

Sometimes one can be interested in running the same experiment but with different genetic architectures or different SNP arrays. SBVB offers two convenient ways to do this as it may keep track of haplotypes so exactly the same genetic structure is preserved, `RESTART` and `RESTARTQTL`.

- With `RESTART`, haplotypes, phenotypes and QTN effects are preserved. This is useful to implement selection.
- With `RESTARTQTN`, haplotypes are preserved but phenotypes and QTN effects are sampled again. `RESTARQTN` can be used to run different genetic architectures in the same haplotypes so results can be exactly comparable across models.

The program then writes a *hap file that contains all haplotype structure the first time is run. When SBVB is called again with say another `SNPFILE`, then individuals have the same haplotypes as in previous runs and a new GRM can be generated with the new SNP file. An important application is to run **selection**. In fact, SBVB can be run with different pedigree files and the `RESTART` option. SBVB generates only new haplotypes for those individuals not in current hap file. In a selection scheme, the user should add a new generation pedigree to current pedfile with the offspring of selected individuals. In the new run, SBVB generates haplotypes and phenotypes for the new offspring.

IMPORTANT:

- **The hapfile is used only if RESTART is included in parfile. If no hapfile is present, a new one is generated the first time. You can check that RESTART is in use checking, e.g, that all phenotypes are the same in different runs of SBVB.**
- **RESTARTQTN is logically not suitable for selection, since effects are sampled anew in each run.**

Expanding the base population

Very often, complete sequence is available only for very few individuals. SBVB implements an automatic option to generate additional individuals by randomly crossing the available ones and random breeding for a pre specified number of generations. To use this feature, the pedigree file must contain larger number of individuals with unknown parents than in the vcf file. For instance, assume your vcf file contains only four individuals and the pedfile is

```
 1    0    0
 2    0    0
 3    0    0
 ...
20    0    0
21    1   13
22    3   11
 ....
```

Then individuals 5-20 are generated by randomly crossing 1-4 ids, from id 21 onwards, normal pedigree gene dropping is implemented. The option is

```
EXPAND_BASEPOP
```

```
ntgen   nfam
```

which means that the new individuals are generated by crossing *nfam* individuals of the *vcf* file for *ntgen* generations.

Examples

Folder Examples contains

1. A toy example consisting of three SNPs from a tetraploid organism
2. An example consisting of 150 SNPs from 100 lines of octoploid strawberry lines.

Toy example

```
# Run
zcat GenosB.vcf.gz | perl vcf2tped2.pl -hap | cut -d ' ' -f 1,4- | /
sbvb -i toy.par
```

Run Toy strawberry example

```
# Generating QTL file with 40 causal loci
NQTL=40
shuf Data_st.chip | head -n $NQTL | sort -gk1,1 -gk2,2 > Y_st.qtn

# Run the example
cat toy_st.gen | sbvb -i file.par
```

Citations

L. Zingaretti, A. Monfort, M. Pérez-Enciso. 2018. Modeling and simulation tools for genomic selection in polyploid species: a case study in octoploid strawberry. Submitted.

M. Pérez-Enciso, N. Forneris, G. de los Campos, A. Legarra. 2017. An evaluation of sequence-based genomic prediction in pigs using an efficient new simulator. Genetics 205:939-953.

APPENDIX: Full list of commands in parameter file

```
# psbvb (Sequence Based Virtual Breeding)
# comments can be included as this
! or as this
# USAGE:
#     zcat file.vcf.gz | perl vcf2tped2.pl -hap | cut -d ' ' -f 1,4- |
./sbvb -i sbvb.par
#
# WARNING: chromosomes ids in vcf file must be consecutive integer
numbers. No alphanumeric characters are allowed

NTRAIT
ntrait

PLOIDY

h

MAXNCHR !—> max no. of chromosomes [23] maxnchr

SEXCHR !--> chr id (number) of sex chromosome

QTLFILE !--> file with qtl posns (chr& bp) add &dom effects can be
defined in cols 3 & 4 qtlfile

EPIFILE

epifile

PEDFILE

pedfile

SNPFILE !--> file with genotyped snps: chr, bp, can be repeated
snpfile

MAPFILE !-->recomb map file: chr, basepos, cm2Mb [cm2Mb_sex2] mapfile

HAPFILE !--> hap structure so program can be restarted with RESTART
hapfile

OUTPLINK !--> prints mkr in plink tpedformat

OUTGFILE !--> GRM outfile outgfile

OUTQFILE !--> output qtl file out_q_file

OUTYFILE !--> y outfile outyfile

OUTMFILE !--> output file with mkr data
```

outmfile

GZIP !--> compress output files

NBASE !-->nind which genotypes are read from STDIN

nbase

H2 !--> heritability h2 ! repeated if multiple traits

H2G !--> broad heritability h2g ! repeated if multiple traits

RHOQA !--> desired correlation between allele effect and frequency rhoqa ! repeated if multiple traits

SIGNQTN !--> P of derived allele being deleterious (only with gamma) [0.5] p_sign_qtl

QTLDISTA !--> QTL add effects are sampled from a distribution: u(niform), g(amma), n(ormal) [u, l_bound, u_bound] | [n, mu, var] | [g, s, b] ! repeated if multiple traits

QTLDISTD !--> QTL dom effects are sampled from a distribution [u, l_bound, u_bound] | [n, mu, var] | [g, s, b] ! repeated if multiple traits

CM2MB !--> cM to Mb rate, default cm2mb [1.0] cm2mb

MXOVER !--> Max no xovers, default 3 mxover

RESTART !--> prepares files for new run of sbvb

RESTARTQTL !--> restart qtl effects but keeps haplotype structure

NOPRINTHAP !--> does not print hap file, eg, if no new haplotypes have been generated

NOSEQUENCE !--> does not use sequence for GRM,

EXPAND_BASEPOP !--> breeds new base individuals involving random mating for ntgen generations ! from nfam families

ntgen nfam

MIMIC_DIPLOID

ALLOPOLYPLOID !--> if PLOIDY is higher than 2 and you want to simulate an allopolyploid organism.

MIMIC_HAPLOID !--> Assuming that only one allele can be distinguished for the others, i.e., that a given marker allele behaves as fully dominant.