pSBVB: Polyploid Sequence Based Virtual Breeding.

A flexible, efficient gene dropping algorithm to simulate sequence based population data and complex traits.

Miguel Pérez-Enciso

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

Purpose

Polyploid sequence based virtual breeding (**pSBVB**) is a modification of **SBVB** software (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. Most of computing time is actually spent in reading the `vcf` file. Future developments will optimize this step by reading and writing binary mapped files. The `vcf` file may not contain missing genotypes and is assumed to be phased.

Main features

*Any number of traits.
*Tool adapted to work with both, auto and allo-polyploid organisms.
*Any number of QTNs, trait specific.
*Any number of additive and dominant effects.
*Can generate a correlation matrix to modelate meiosis in polyploid especies.
*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.
*It allow to compute Genomic relationship matrix in several ways.
*Any number of chromosomes, allows for sex chromosomes and varying local recombination rates, that can be sex specific.

Installation

The source code, manual and examples can be obtained from https://github.com/mperezenciso/sbvb0

To compile:

```
gfortran -O3 kind.f90 ALliball.f90 aux_sub11.f90 psbvb.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.

Usage

To run (assuming `vcf` i file is compressed):

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

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

allele1_snp1_ind1 allele2_snp1_ind1 allele3_snp1_ind1 ... allelep_snp1_ind1
allele1_snp1_ind2 allele2_snp1_ind2 allele3_snp1_ind2 ... allelep_snp1_indp

allele1_snp2_ind1 allele2_snp2_ind1 allele3_snp2_ind1 ... allelep_snp2_ind1
allele1_snp2_ind2 allele2_snp2_ind2 allele3_snp2_ind2 ... allelep_snp2_indp

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

```
… | psbvb –isbvb.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 1. In the following, we list the main ones.

PLOIDY

hCompared to SBVB (designed for diploid organisms), **pSBVB** allows simulating meiosis in autopolyploid or allopolyploid species. For that, **pSBVB** requires a matrix of dimension $h \times h$, must be consecutive integers h is the ploidy level specifying the pairing factors described above. To specify this matrix you must insert in file parameter:

tells the program that the organisms used have ploidy h.

Compared to SBVB (designed for diploid organisms), **pSBVB** allows simulating meiosis in autopolyploid or allopolyploid species. For that, **pSBVB** requires a matrix of dimension $h \times h$, must be consecutive integers h is the ploidy level specifying the pairing factors described above. To specify this matrix you must insert in file parameter:

RHOMATRIX

Verrrrrrr como se ponía esta matriz de recombinación

Specifying genetic architecture

If more than one trait is generated, then use

NTRAIT

ntraits

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

i_chrom	i_pos

or

i_chrom	i_pos	add_eff_Trait_1	add_eff_Trait_2	...	add_eff_Trait_n

or to additive and dominant effects:

i_chrom	i_pos	add_eff_Trait_1	add_eff_Trait_2		add_eff_Trait_n	dom_eff_Trait_1	dom_eff_Trait_2	**...**	dom_eff_Trait_n

NOTE: The bellow file, must separated by spaces, and where *ichr* is chromosome and *ipos* is position in base pair, *add_eff* is additive effect, i.e the effect of homozygous alleles and *dom_eff* is the heterozygous effect.

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 distribution $U \sim (lower_{bound}, upper_{bound})$, 'n' from a normal distribution $N \sim (mu, var)$ and 'g' from a gamma $\gamma \sim (s, b)$. For a gamma distribution, you can specify the probability p that a derived allele decreases the phenotype with:

PSIGNQTN

p

The default value is 50%. By default, effects are sampled independently of frequency, i.e., half effects are + and the rest are -, but it is possible to generate a correlation (rho) using the next parameter:

RHOQA

rho

This option can be useful to simulate past selection.

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.

Phenotyping simulations

As **pSBVB** takes ploidy into account to generate the phenotypes and incorporates several options to generate the molecular relationship matrix that are pertinent to polyploids. In a diploid organism, the phenotype for i-th individual can be simulated from:

$$y_i = \mu + \sum_{j=1}^{Q} \gamma_{ij}\alpha_j + \sum_{j=1}^{Q} \delta_{ij}d_j + \epsilon_i$$

Where μ is the mean general, α 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. For polyploids, the phenotype of individual i (y_i) (equivalent equation) is simulated from:

$$y_i = \mu + \sum_{j=1}^{Q} \eta_{ij}\alpha_j + \sum_{j=1}^{Q} \phi_{ij}d_j + \epsilon_i$$

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 α_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 $\phi_{ij} = 1$, that is, any genotype having at least one allele ‘1’ and ‘0’ has the expected phenotypic value d. You can coded ϕ_{ij} as any integer between 1 and $h - 1$.

Finally, the residual ϵ_i is sampled from a $N \sim (0, ve)$, where ve is adjusted given either **H2** or **H2G** using the genotypes from the base popula2tion.

For multiple traits, the fields **H2** or **H2G**, **RHOQA**, and **QTLDISTA** and **QTLDISTD** must be repeated, eg, for two traits:

H2

0.5

0.23

RHOQA

0

-0.4

QTNDISTA

u -0.2 0.2

g 1 0.5

which means that the firts trait have a heredability of 0.5, a **RHOQA** parameter of 0 and **QTNDISTA** have an uniform distribution $(0.2, 0.2)$ and the second trait have a heredability of 0.23, **RHOQA** parameter is -0.4 and **QTNDISTA** have a gamma distribution with parameters $(1, 0.5)$

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

1	0	0
2	0	0
...	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

MAPFILE

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_cm2mbMales	local_cm2mb_females

The maximum number of chromosomes allowed by default is 23; should you require more, then section **MAXNCHR** must be included as:

MAXNCHR

nchrom

pSBVB 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**.

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

SNP files

pSBVB can compute the genomic relationship matrix for all sequence data (in two specific ways, see bellow), 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, as idicated:

SNPFILE

i_chrom	i_pos

if you add the command **MIMICDIPLOID** to parameter file, then Genomic relationship matrix is computed assuming than only presence or absence of the alternative allele could be known for the remaining, i.e., although the organism was polyploid, Genomic matrix is computed mimic diploid.

Output

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

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

id	y	$add_i, i = 1, .., ntraits$	$(add + dom)_i, i = 1, .., ntraits$

where add is the first sum in equation of **pSBVB** software, shows above and dom is the second term. For several traits, first are printed all add effects for every trait, next add+dom.

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

$ichr$	pos	$freq_{base}$	(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, add **NOSEQUENCE** command 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 par file.

NOTE: To compress marker output, include **GZIP** option 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. The program 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** options in `.par` file.

1.With **RESTART**, haplotypes, phenotypes and **QTN** effects are preserved. This is useful to implement selection.

2.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 **pSBVB** 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, **pSBVB** can be run with different pedigree files and the **RESTART** option. **pSBVB** 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, **pSBVB** generates haplotypes and phenotypes for the new offspring.

IMPORTANT:
The `.hap` file is used only if **RESTART** is included in parfile. If no `.hap` file 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.

WARNING:
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. **pSBVB** 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

1	0	0
3	0	0
...	0	0
20	0	0
21	1	12
...	...	...

Then individuals 5-20 are generated by randomly crossing 1-4 ids, from id 21 onwards, normal pedigree gene dropping is implemented. The option in parfile 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 two, a toy example consisting in three **SNPs** from a tetraploid individuals and one example consisting of 150 SNPs from the X chromosome of 100 lines from octoploid strawberry lines. The description of the files is:

Base genotypes
test.vcf: original vcf file
test.gen: results from cat test.vcf | perl vcf2tped2.pl -hap | cut -d ' ' -f 1,4-

One trait (cat test.gen | sbvb -i test.par)
test.par: par fisigmale
test.qtn: list of causal SNPs, additive effects are sampled from a gamma
test.epi: epi file with two interactions (commented out in test.par by default)
test.chip: a list of SNPs from a given array
test.outy: phenotype and breeding values
test.outq: QTN effects

test.outm* : genotypes data
test.outg: GRMs

Two traits (cat test.gen | sbvb -i test.par)
test2.par
test2.qtnsigma
test2.epi
test2.outq
...

Citation

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

Appendix