**GOplan Manual**

GOplan is a R package designed for manage animal breeding program, including the breeding program of core population and the whole breeding strategy of crossbreeding system. Moreover, it allows optimization of the crossbreeding system.

The package has three main functions. Parameter “prm\_path” is the full path of input parameter file:

***runCore(prm\_path)*** is the function used for manage core population, you can compare the genetic progress as well as the monetary profit under different key elements in constructing populations such as the longevity of breeding stocks, mating ratio, mating methods, the number of dams, cull rate, the method of breeding value estimation, the number of individual measured or genotyped and so on. It returns five aspects of results, the relative genetic progress, the increase of population inbreeding, the decrease of population genetic variance, the monetary profit and the phenotype.

***runWhole(prm\_path)*** can manage the crossbreeding system, it allows the evaluation of system with sire station or not, and return the predicted value of hybrids’ phenotype and monetary profit.

***runOpt(prm\_path)*** used for find an optimized draft of the whole crossbreeding system. By defining some uncertain parameters and giving reasonable variation ranges, it uses Bayesian Optimization method to draw up a superior solution under fixed iteration times.

**Description of parameter file:**

All needed information is determined by editing the parameter files. It is worth noting that the parameters you do not need should be specified by 0, and do not use any other separators except blank to separate different numbers, or it may cause unknow errors. Do not change the sequence of each parameter.

The parameter file contains six parts, each part would be illustrated below.

**### Analyse Parameters ###**

This part used to determine some basic information when run all three functions.

|  |  |  |
| --- | --- | --- |
| **parameters** | **value** | **explanation** |
| QUICK | FALSE | #FALSE, TRUE |
| Method | blup | #blup, ssgblup, gblup |
| Mate | rand | #Mate method: rand, MC or the path of extra program |
| Ctype | 33 | #type of crossbreeding: 1, 21, 22, 31, 32, 33, 41, 42 |
| Nrep | 50 | #number of repetitions |
| Time | 20 | #time budget of breeding cycle |
| Ncores | 10 | #number of threads |
| nChr | 18 | #number of simulated chromosomes |
| nSnpPerChr | 3000 | #number of SNP per chr |
| nQtlPerChr | 100 | #number of QTL per chr |
| out\_path |  | #path of output file |

**QUICK**: when set to TRUE, the program would simplify the simulation of core population through replacing the process of breeding value estimation by simulation, which means just calculate the first generation’s breeding value and then use its accuracy to simulate the other generations’ breeding value. It can save time and quickly get a result, if you just want to know which factors matters more and do not care about the exact profit, it would be time-saving to use it.

**Method:** now the program support three methods to evaluate the breeding value, and call DMU to run it. blup: best liner unbiased prediction, use only pedigree information; **ssgblup**: single-step blup, use both genomic and pedigree information; gblup: genomic blup, use only genomic information.

**Mate**: rand: random mating; MC: minimal-coancestry mating, using annealing algorithm to find the solution; or you can give the full path of your own mating program, while our package would write out a file called “plans.txt” containing four column, id of dam, id of sire, relationship coefficient and the expected breeding value of offspring.

**Ctype:** crossbreeding type, see detail in Figure1, 1 means single population, 2\* means tow-way crossbreeding, 3\* means three-way crossbreeding and 4\* means four-way crossbreeding.

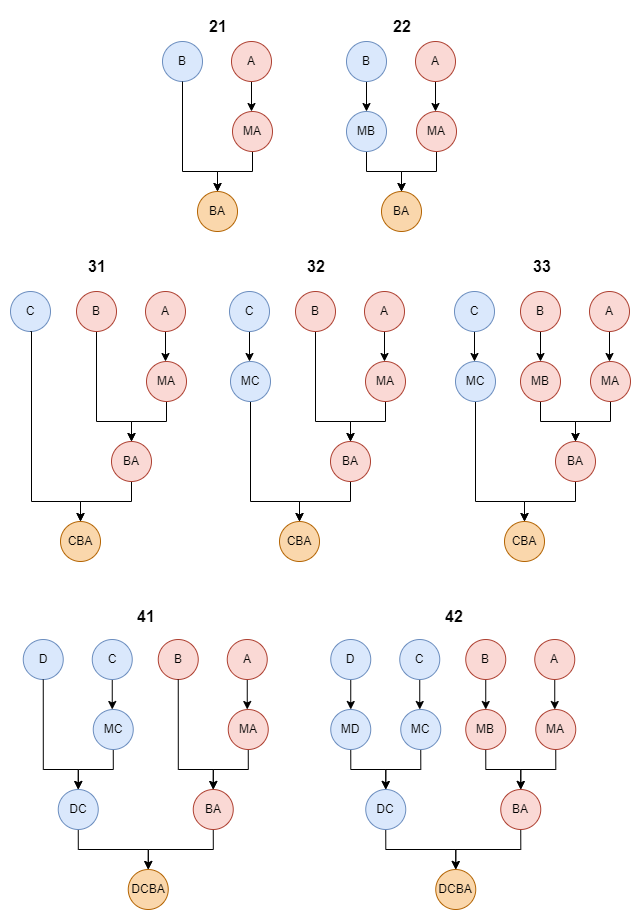


Figure1: the sketch map of different crossbreeding types.

Nrep: the number of repetitions. Final results would be the average of all repetitions.

Time: time budget of breeding cycle.

Ncores: number of threads. Our package supports multithreaded running. But you should make sure that do not set a number exceed the maximum available thread of your system.

nChr: number of simulated chromosomes. Change this parameters to adapt to different breeds.

nSnpPerChr: number of SNP per chr.

nQtlPerChr: number of QTL per chr.

out\_path: path of output file, do not forget add “/” in the end.

**### VARIABLES ###**

This part can either be specified or not, if you do not need to set variables, just skip this part. The number of variables do not have limitation, but more variables means more combinations, which may cause long time running. We recommend less than three variables each time, and one variable one row, with all the levels listed separated by blank.

Variables for ***runCore(prm\_path):*** Ys, Yd, Sor, nfam, nfam\_F, nfam\_M, ngeno, nGeno\_F, nGeno\_M. Using nfam or ngeno means adjust nfam\_F and nfam\_M or nGeno\_F and nGeno\_M simultaneously.

Variables for ***runWhole(prm\_path):*** Asec, Bsec, Csec, Dsec, YABd, YCDs, SorAB, SorCD, SorP

**### Optimize Parameters ###**

This part used to manage the range of each optimize parameter. If you don’t need some parameters, just delete the row of it. When use ***runCore(prm\_path), runWhole(prm\_path)*** you can let this whole part blank.

|  |  |  |  |
| --- | --- | --- | --- |
| **parameters** | **value** | **value** | **explanation** |
| Pars | lower | upper |  |
| Asec | 2 | 6 | #longevity of mutiplier A |
| Bsec | 1 | 4 | #longevity of mutiplier B |
| Csec | 2 | 6 | #longevity of mutiplier C |
| Dsec | 1 | 4 | #longevity of mutiplier D |
| YABd | 2 | 6 | #longevity of hybrid dam AB |
| YCDs | 1 | 4 | #longevity of hybrid dam CD |
| NAd | 600 | 3000 | #female size of nucleus A |
| NBd | 300 | 1000 | #female size of nucleus B |
| NCd | 300 | 1000 | #female size of nucleus C |
| NDd | 300 | 1000 | #female size of nucleus D |

**### Population Structure ###**

This part used to specify the structure of population in the breeding program.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **parameters** | **value** | **value** | **value** | **value** | **explanation** |
| dam0 | 2 |  |  |  | #age of dams when first born |
| sire0 | 2 |  |  |  | #age of sires when first born |
| YABd | 6 |  |  |  | #longevity of hybrid dam AB |
| cull\_AB | 0 | 0 | 0 |  | #cull rates of each age in AB |
| YCDs | 4 |  |  |  | #longevity of hybrid sire CD |
| cull\_CD | 0 | 0 | 0 |  | #cull rates of each age in CD |
| SorAB | 100 |  |  |  | #Mating ratio when AB cross |
| SorCD | 100 |  |  |  | #Mating ratio when CD cross |
| SorP | 100 |  |  |  | #Mating ratio of terminal cross |
| ABSR | 0.85 |  |  |  | #survival rate of AB |
| CDSR | 0.85 |  |  |  | #survival rate of CD |
| PSR | 0.85 |  |  |  | #survival rate of products |
| N\_P | 0 |  |  |  | #expected number of final products |
| nPoints | 0 |  |  |  | #number of selected points each iteration |
| nIter | 0 |  |  |  | #number of iterations |
|  |  |  |  |  |  |
| Breed | A | B | C | D |  |
| n\_female | 500 | 500 | 500 | 0 | #female size |
| Sor | 50 | 50 | 50 | 0 | #mate ratio |
| Ys | 2 | 2 | 2 | 0 | #longevity of sires |
| Yd | 4 | 4 | 4 | 0 | #longevity of dams |
| n\_progeny | 12 | 12 | 12 | 0 | #litter size |
| SR | 0.85 | 0.85 | 0.85 | 0 | #survival rate of offspring |
| nSec | 3000 | 0 | 0 | 0 | #size of mutiplier |
| Ysec | 6 | 2 | 2 | 0 | #longevity of mutiplier |
| nfam\_F | 2 | 2 | 2 | 0 | #number of phenotyping females per litter |
| nfam\_M | 2 | 2 | 2 | 0 | #number of phenotyping males per litter |
| nGeno\_F | 0 | 0 | 0 | 0 | #number of genotyping females per litter |
| nGeno\_M | 0 | 0 | 0 | 0 | #number of genotyping males per litter |

**### Breeds Details ###**

This part used to specify the information of each breed. Just list the information of breed that needed.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **parameters** | **value** | **value** | **value** | **value** | **value** | **explanation** |
| Breedname | A |  |  |  |  |  |
| cull\_s | 0 |  |  |  |  | #cull rates of each age in sires |
| cull\_d | 0 | 0 | 0 |  |  | #cull rates of each age in dams |
| cull\_sec | 0 | 0 | 0 | 0 | 0 | #cull rates of each age in multipliers |
| trait | JZRL |  |  |  |  | #names of considering traits |
| mean | 180 |  |  |  |  | #means of traits |
| var | 100 |  |  |  |  | #phenotype variances of traits |
| heri | 0.3 |  |  |  |  | #heritabilities of traits |
| weigh | -1 |  |  |  |  | #weight coefficients of traits |
| analyse | 1 |  |  |  |  | #analyse method |
| ## phenotype covariance ## | | | | | | |
| JZRL | 1 |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Breedname | B |  |  |  |  |  |
| cull\_s | 0 |  |  |  |  |  |
| cull\_d | 0 | 0 | 0 |  |  |  |
| cull\_sec | 0 | 0 | 0 | 0 | 0 |  |
| trait | JZRL |  |  |  |  |  |
| mean | 170 |  |  |  |  |  |
| var | 100 |  |  |  |  |  |
| heri | 0.3 |  |  |  |  |  |
| weigh | -1 |  |  |  |  |  |
| analyse | 1 |  |  |  |  |  |
| ## phenotype covariance ## | | | | | | |
| JZRL | 1 |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Breedname | C |  |  |  |  |  |
| cull\_s | 0 |  |  |  |  |  |
| cull\_d | 0 | 0 | 0 |  |  |  |
| cull\_sec | 0 | 0 | 0 | 0 | 0 |  |
| trait | JZRL |  |  |  |  |  |
| mean | 160 |  |  |  |  |  |
| var | 100 |  |  |  |  |  |
| heri | 0.3 |  |  |  |  |  |
| weigh | -1 |  |  |  |  |  |
| analyse | 1 |  |  |  |  |  |
| ## phenotype covariance ## | | | | | | |
| JZRL | 1 |  |  |  |  |  |

**analyse**: "1" means single trait, other same numbers mean multiple traits. For example, if there are three traits, T1, T2 and T3, T1 need to be estimated by single trait, while T2 and T3 need to be estimated using multi-trait model. Then we set analyse: 1, 2, 2.

**## phenotype covariance ##**: this part determines the covariance of traits. For example, when there are three traits, T1, T2 and T3. Then it needs to be adjusted as below:

|  |  |  |  |
| --- | --- | --- | --- |
| T1 | 1 | 0 | 0 |
| T2 | 0 | 1 | 0 |
| T3 | 0 | 0 | 1 |

Noticed that we the column names are constant to the row names, though we omit them.

The parameter number of cull rate should always be one less than its corresponding longevity. For example, if the Yd of breed C is 4, then the cull\_d of C must have three values (0, 0, 0). Because when parent exceed their longevity, they would be all culled.

**### Basic Economic Parameters ###**

This part defines the economic information of the breeding program.

|  |  |  |  |
| --- | --- | --- | --- |
| **parameters** | **value** | **value** | **explanation** |
| trait | 0 | JZRL |  |
| jb\_cost | 906 | 1.5 | #basic cost per individual |
| dam\_cost | 1500 | 0 | #dam cost per individual |
| sire\_cost | 1000 | 0 | #sire cost per individual |
| other\_cost | 10 | 0 | #other cost each season |
| pig\_sale | 2040 | 0 | #income of selling a product |
| cull\_sale | 1500 | 0 | #income of culling |
| sire\_sale | 3000 | 0 | # income of selling a sire |
| dam\_sale | 2000 | 0 | # income of selling a dam |
| meas\_cost | 20 |  | #phenotyping cost per individual |
| geno\_cost | 200 |  | #genotyping cost per individual |

**trait:** first column fixed with 0, means the initial value of all economic parameters. Then list the traits concerned, each trait corresponds to a column, means the monetary change when each trait changes one unit. The value can be negative.

**jb\_cost:** the basic cost for raising an individual.

**dam\_cost:** cost for keep a dam in the population each season.

**sire\_cost:** cost for keep a sire in the population each season.

**other\_cost:** total other cost each season.

**pig\_sale:** income of selling a product.

**cull\_sale:** income of culling a breeding stock.

**sire\_sale:** income of selling a sire, a male who had been measured but not be selected can be sold as a sire.

**dam\_sale:** income of selling a dam, a female who had been measured but not be selected can be sold as a dam.

**meas\_cost:** phenotyping cost per individual.

**geno\_cost:** genotyping cost per individual