

# 计算机模拟演示结合杂种信息的基因组选择

张哲

## 目录

|                                      |          |
|--------------------------------------|----------|
| <b>1 引言</b>                          | <b>1</b> |
| <b>2 具体操作</b>                        | <b>1</b> |
| 2.1 加载所需软件包 . . . . .                | 1        |
| 2.2 第一步：创建历史群体 . . . . .             | 3        |
| 2.3 第二步：起始 (founder) 群体构建 . . . . .  | 3        |
| 2.4 第三步：用于杂交的群体构建 . . . . .          | 4        |
| 2.5 第四步：杂交 . . . . .                 | 5        |
| <b>3 基于 GEBVC 的选择与基于 GEBVP 选择的对比</b> | <b>7</b> |
| 3.1 选择策略更换 . . . . .                 | 7        |
| 3.2 结果对比 . . . . .                   | 8        |

## 1 引言

本代码主要演示了如何利用 xbreed 软件包来模拟针对杂种的表现来对纯种进行基因组选择的过程，并与一般的针对纯种自身的表现来对纯种进行选择的效果进行对比，对比的标准是杂种的平均表型是否随着时间上升。

## 2 具体操作

### 2.1 加载所需软件包

```
library(xbreed)
```

所模拟群体的整体框架如图 1 所示：

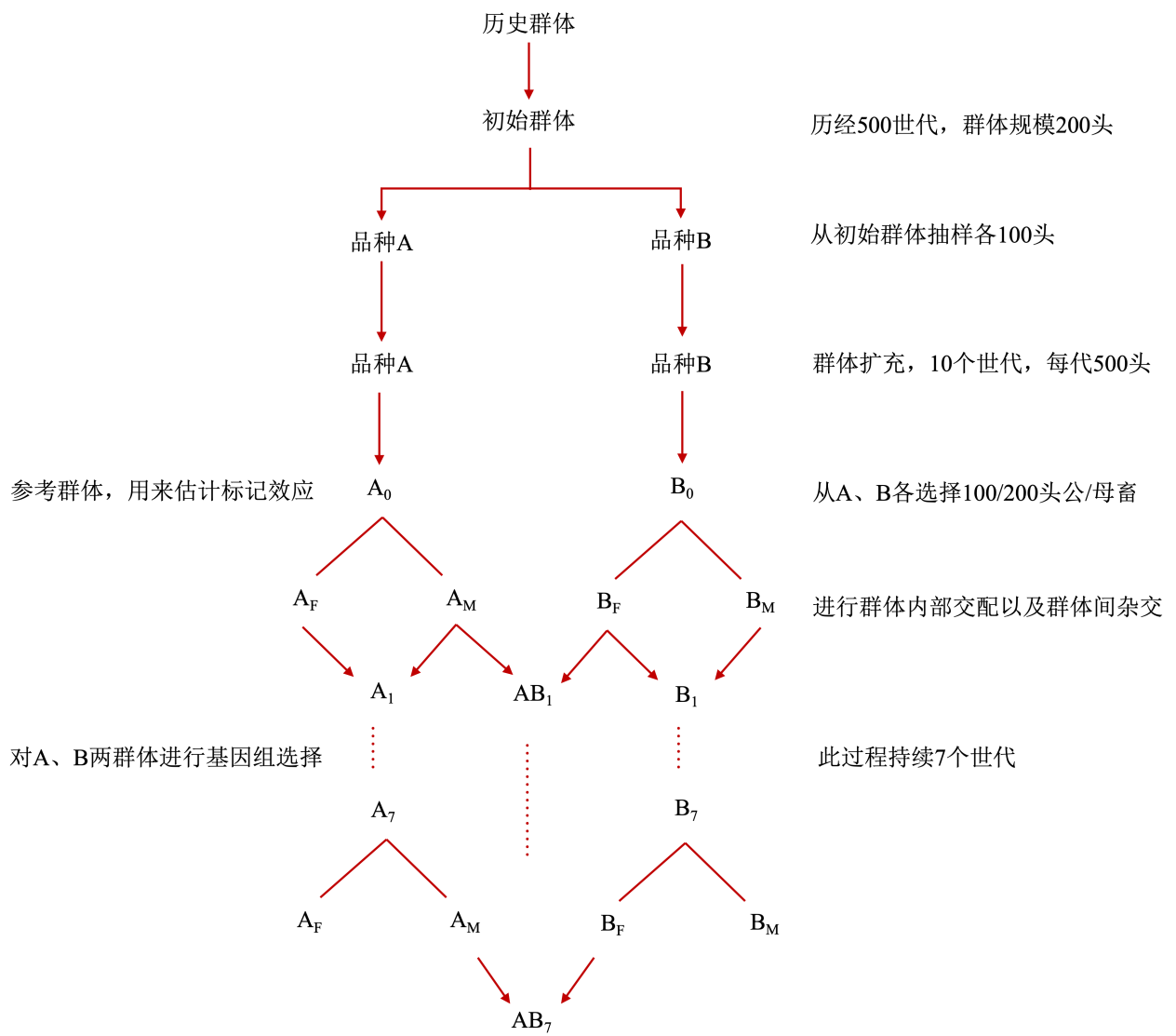


图 1: 模拟整体的框架

## 2.2 第一步：创建历史群体

### 2.2.1 基因组

基因组由三条染色体构成，其上随机分布着 QTL 与标记，QTL 共有 135 个，标记共有 1750 个

```
genome<-data.frame(matrix(NA, nrow=3, ncol=6))
names(genome)<-c("chr", "len", "nmrk", "mpos", "nqtl", "qpos")
genome$chr<-c(1:3)
genome$len<-c(80,60,50)
genome$nmrk<-c(500,1000,250)
genome$mpos<-c('rnd', 'rnd', 'rnd')
genome$nqtl<-c(40,50,45)
genome$qpos<-c('rnd', 'rnd', 'rnd')
```

### 2.2.2 历史 (hisorical) 群体创建

共历经 500 个世代，最终形成 200 个个体所组成的历史群体

```
hp<-make_hp(hpsize=200,
            ng=500,h2=0.3,d2=0.2,phen_var=1,
            genome=genome,mutr=5*10**-4,sel_seq_qtl=0.1,sel_seq_mrk=0.05,laf=0.5)
```

## 2.3 第二步：起始 (founder) 群体构建

从历史群体中进行随机抽样来构建 A、B 两个品种的起始群体，并通过设置母畜的产仔数（5 头）来对两品种进行扩充

### 2.3.1 品种 A

从历史群体中随机抽样公母各 50 头

```
Breed_A_Male_fndrs<-data.frame(number=50,select='rnd')
Breed_A_Female_fndrs<-data.frame(number=50,select='rnd')
```

从历史群体抽样之后通过连续随机交配来扩大群体，每次都是随机选择 50 头公畜与 100 头母畜来进行随机交配，另外，用来繁衍下一代个体的亲本通过表型从低到高进行选择

```
Selection<-data.frame(matrix(NA, nrow=2, ncol=3))
names(Selection)<-c('Number', 'type', 'Value')
Selection$Number[1:2]<-c(50,100)
Selection$type[1:2]<-c('phen', 'phen')
Selection$Value[1:2]<-c('l', 'l')
```

litter\_size 设为 5，从而达到扩充群体的目的，通过调用 sample\_hp 可以完成其实群体 A 的构建

```
Breed_A<-sample_hp(hp_out=hp, Male_founders=Breed_A_Male_fndrs,
                   Female_founders=Breed_A_Female_fndrs,
                   ng=10, Selection=Selection,
                   litter_size=5, Display=FALSE)
```

### 2.3.2 品种 B

从历史群体中随机抽样公母各 50 头

```
Breed_B_Male_fndrs<-data.frame(number=50, select='rnd')
Breed_B_Female_fndrs<-data.frame(number=50, select='rnd')
```

为了让两个品种的遗传结构的差异进一步加大，品种 B 按照表型进行从高到低的选择

```
Selection<-data.frame(matrix(NA, nrow=2, ncol=3))
names(Selection)<-c('Number', 'type', 'Value')
Selection$Number[1:2]<-c(50, 100)
Selection$type[1:2]<-c('phen', 'phen')
Selection$Value[1:2]<-c('h', 'h') #这里表示优先挑选表型值较大的个体
```

品种 B 也是经过 10 个世代，每个世代每头母猪产生 5 头仔畜

```
Breed_B<-sample_hp(hp_out=hp, Male_founders=
                   Breed_B_Male_fndrs, Female_founders=Breed_B_Female_fndrs,
                   ng=10, Selection=Selection,
                   litter_size=5)
```

## 2.4 第三步：用于杂交的群体构建

从品种 A 中随机选择 100 头公畜、200 头母畜，每头母畜产生 5 头仔畜，这样就构建了群体规模为 1000 头个体的 A0

```
founder_pop1<-data.frame(matrix(NA, nrow=2, ncol=3))
names(founder_pop1)<-c('size', 'generation', 'select')
founder_pop1[1,]<-c(100, 10, 'rnd')
founder_pop1[2,]<-c(200, 10, 'rnd')
```

从品种 B 中随机选择 100 头公畜、200 头，每头母畜产生 5 头仔畜，这样就构建了群体规模为 1000 头个体的 B0

```
<-data.frame(matrix(NA, nrow=2, ncol=3))
names(founder_pop2)<-c('size','generation','select')
founder_pop2[1,<-c(100,10,'rnd')
founder_pop2[2,<-c(200,10,'rnd')
```

从品种 A 随机选择公畜 100 头，从品种 B 随机选择母畜 200 头，用来杂交产生 AB0

```
founder_cross<-data.frame(matrix(NA, nrow=2, ncol=3))
names(founder_cross)<-c('pop','size','select')
founder_cross[1,<-c('pop1',100,'rnd')
founder_cross[2,<-c('pop2',200,'rnd')
```

## 2.5 第四步：杂交

针对 A 群体设置的选择模式，方法为 GEBVC

```
Selection_pop1<-data.frame(matrix(NA, nrow=2, ncol=3))
names(Selection_pop1)<-c('Number','type','Value')
Selection_pop1$Number[1:2]<-c(100,200)
Selection_pop1$type[1:2]<-c('gebvc','gebvc')
Selection_pop1$Value[1:2]<-c('h','h')
```

同样地，针对 B 群体设置的选择方法也是 GEBVC

```
Selection_pop2<-data.frame(matrix(NA, nrow=2, ncol=3))
names(Selection_pop2)<-c('Number','type','Value')
Selection_pop2$Number[1:2]<-c(100,200)
Selection_pop2$type[1:2]<-c('gebvc','gebvc')
Selection_pop2$Value[1:2]<-c('h','h')
```

设置如何从 A、B 两群体中选择杂交亲本的方式，方法依然为 GEBVC

```
Cross_design<-data.frame(matrix(NA, nrow=2, ncol=4))
names(Cross_design)<-c('pop','size','select','value')
Cross_design[1,<-c('pop1',100,'gebvc','h')
Cross_design[2,<-c('pop2',200,'gebvc','h')
Cross_design
```

设置 A 群体的参考群体，大小为 500 头，随机选择（这里实际上相当于保留了全部个体 500 头），标记效应的估计方法为 BayesB

```

train_A<-data.frame(matrix(NA, nrow=1, ncol=5))
names(train_A)<-c('size','sel','method','nIter','show')
train_A$size<-500
train_A$sel<- 'rnd'
train_A$method<- 'BayesB'
train_A$nIter<-1000 #抽样次数
train_A$show<-FALSE

```

针对 B 群体设置 GEBV 的估计方法，与 A 群体的参数相同

```

train_B<-data.frame(matrix(NA, nrow=1, ncol=5))
names(train_B)<-c('size','sel','method','nIter','show')
train_B$size<-500
train_B$sel<- 'rnd'
train_B$method<- 'BayesB'
train_B$nIter<-1000
train_B$show<-FALSE

```

保存 AB1 - AB7 这七个世代的杂种群体的数据

```

output_cross<-data.frame(matrix(NA, nrow=7, ncol=5))
names(output_cross)<-c('data','qtl','marker','freq_qtl','freq_mrk')
output_cross[,1]<-1:7
output_cross[,2]<-1:7
output_cross[,3]<-1:7
output_cross[,4]<-1:7
output_cross[,5]<-1:7

```

调取杂交函数 `xbreed()` 进行杂交

```

set.seed(123) #设置随机种子，从而保证此过程可重复
cross_AB<-xbreed(pop1=Breed_A,pop2=Breed_B,founder_pop1=
  founder_pop1,founder_pop2=founder_pop2,
  founder_cross=founder_cross,
  Selection_pop1=Selection_pop1,Selection_pop2=Selection_pop2,
  Cross_design=Cross_design,train_type='purebred',
  train_pop1=train_A,train_pop2=train_B,ng=7,litter_size=5,
  saveAt='cross_pop_gebvc',output_cross=output_cross,Display=FALSE)

```

### 3 基于 GEBVC 的选择与基于 GEBVP 选择的对比

上一小节利用的是 Esfandyari et al. (2015) 的方法，针对杂种的表现来选择纯种，所依赖的选择标准是 GEBVC，那么，如果我们按照常规的针对纯种自身的表现来进行选择，即基于 GEBVP 来进行选择时，效果又如何呢？

#### 3.1 选择策略更换

针对 A 群体设置的选择模式，方法为 GEBVP，即只针对纯种群体自身的 GEBV 进行排序

```
Selection_pop1<-data.frame(matrix(NA, nrow=2, ncol=3))
names(Selection_pop1)<-c('Number','type','Value')
Selection_pop1$Number[1:2]<-c(100,200)
Selection_pop1$type[1:2]<-c('gebv','gebv')
Selection_pop1$Value[1:2]<-c('h','h')
```

同样地，针对 B 群体设置的选择方法也是 GEBVP

```
Selection_pop2<-data.frame(matrix(NA, nrow=2, ncol=3))
names(Selection_pop2)<-c('Number','type','Value')
Selection_pop2$Number[1:2]<-c(100,200)
Selection_pop2$type[1:2]<-c('gebv','gebv')
Selection_pop2$Value[1:2]<-c('h','h')
```

设置如何从 A、B 两群体中选择杂交亲本的方式，方法依然为 GEBVP

```
Cross_design<-data.frame(matrix(NA, nrow=2, ncol=4))
names(Cross_design)<-c('pop','size','select','value')
Cross_design[1,]<-c('pop1',100,'gebv','h')
Cross_design[2,]<-c('pop2',200,'gebv','h')
```

最后利用 `xbreed()` 函数来进行杂交，同样保存 7 个世代杂种的数据用于比较

```
set.seed(123) #设置与 GEBVC 选择时相同的随机种子，保证两次模拟的各个初始参数相同
cross_AB<-xbreed(pop1=Breed_A,pop2=Breed_B,founder_pop1=
  founder_pop1,founder_pop2=founder_pop2,
  founder_cross=founder_cross,
  Selection_pop1=Selection_pop1,Selection_pop2=Selection_pop2,
  Cross_design=Cross_design,train_type='purebred',
  train_pop1=train_A,train_pop2=train_B,ng=7,litter_size=5,
  saveAt='cross_pop_gebvp',output_cross=output_cross,Display=FALSE)
```

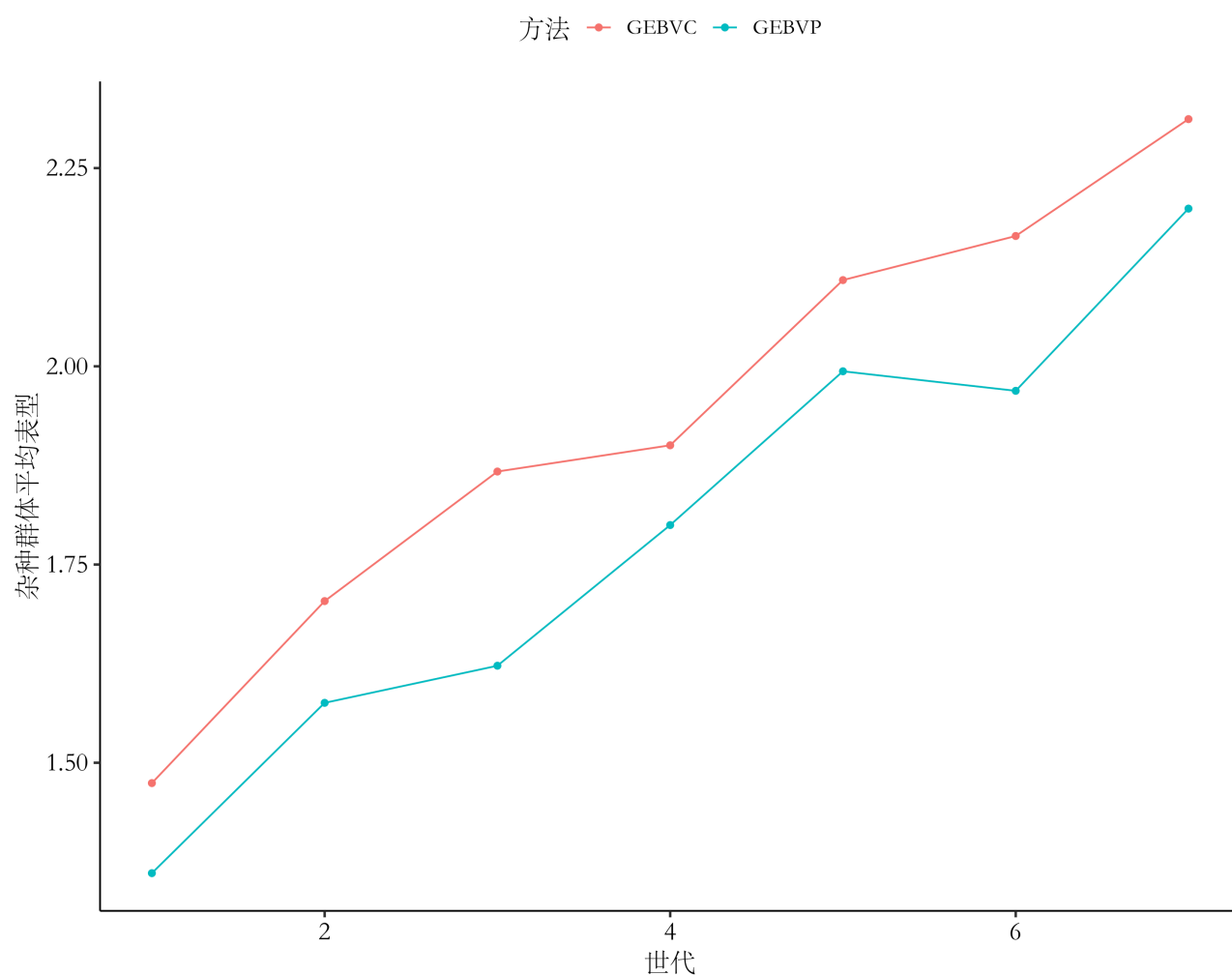


图 2: 不同选择策略下的杂种群体表型均值

### 3.2 结果对比

图 2 展示了基于 GEBVP 与基于 GEBVC 进行选择时七个世代杂种群体 AB1 - AB7 的表型均值的变化, 从图可以看出, 在我们这种模拟的条件下, 基于 GEBVC 选择用于杂交的亲本群体的效果要优于 GEBVP, 其表型均值在各个世代都要高于基于 GEBVP 进行选择的杂种群体均值。