In [1]:
```
# Founders: real haplotype data (ch1to10.200SNP)
# "Sample animals for sire and dam candidates" & "Dams and Sires": dam size = N of dams
# One Dam -> 1 male & 1 female (1 dam -> 2 offsprings)
# selection: increase
# 5 generation selection: increase
# heritability = 0.5
# Phenotypes : all animals in G0 to G4
# Genotypes  : all progeny in G5 and all sires in each generation
# Change muAlpha = 0.2
# 10 chromosomes; 200 loci per chromosome = > 2000 Loci (500 QTL & 1500 Markers)
```

In [2]:
```
include("/home/nicole/Jupyter/XSimSel.jl")
```

Out[2]: XSim

In [3]:
```
using DataFrames
```

In [4]:
```
using Distributions
```

In [5]:
```
using(Gadfly)
```

# Initialize XSim

In [6]:
```
numChr    = 10
chrLength = 0.1
numLoci   = 200
nQTL      = 50
mutRate   = 0.0
locusInt  = chrLength/numLoci
mapPos    = collect(locusInt/2:locusInt:chrLength)
geneFreq  = fill(0.5,numLoci)
QTL = sample(1:numLoci,nQTL,replace=false)
qtlMarker = fill(false,numLoci)
qtlMarker[QTL] = true
mu = 100                                  #  alpha ~ N(100,1)
Va = nQTL*numChr*0.5*mu*mu                # Va= nQTL*2pq*mean(alpha)^2
qtlEffects= rand(Normal(mu/sqrt(Va), sqrt(1/Va)),numLoci*numChr)  # Let alpha mu = 0.2
XSim.init(numChr,numLoci,chrLength,geneFreq,mapPos,qtlMarker,qtlEffects,mutRate)
```

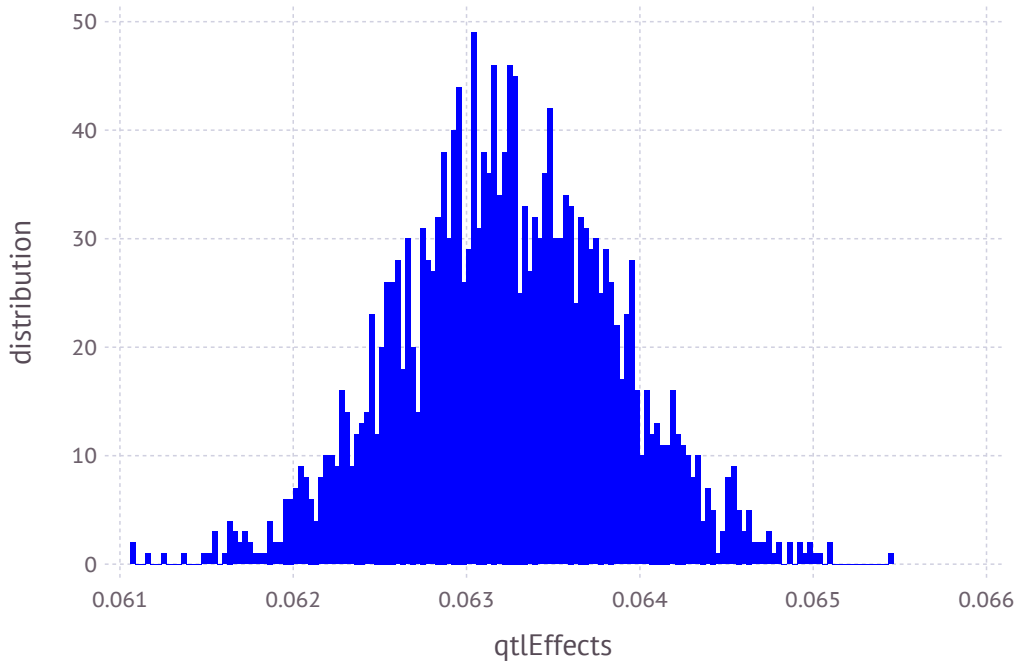
In [7]:
```
qtlEffects
```

Out[7]:
```
2000-element Array{Float64,1}:
 0.0632963
 0.0639308
 0.0631616
 0.0642648
 0.0628607
 0.0641926
 0.064537
 0.062994
 0.063574
 0.0634599
 0.0634855
 0.0640894
 0.0640454
 ⋮
 0.0629622
 0.0633943
 0.0636136
 0.0619846
 0.0626055
 0.062736
 0.0638941
 0.0628504
 0.0637805
 0.0636843
 0.063674
 0.0627396
```

In [8]:
```
writedlm("qtlEffects",qtlEffects)
```

In [9]:
```
plot(x=qtlEffects, Geom.histogram, Guide.XLabel("qtlEffects"), Guide.YLabel("distribution"), Theme(defaul
```

Out[9]:




In [10]:
```
mean(qtlEffects)
```

Out[10]:
```
0.06323345794631947
```

In [11]:
```
var(qtlEffects)
```

Out[11]:
```
3.962516070792277e-7
```

In [12]:
```
# Base Population
gen=0
nGenBase    = 5
popSizeBase = 8000

# Sample 20 sire and 400 dams
popSizeSP = 8000

# Purbred Populations - mating
popSize = 8000
nGener  = 5
nSires  = 200
nDams   = 4000
npop = 1
;
```

In [13]:
```
# Animals with genotypes
posOFF0S = 1
posOFF0E = popSizeSP
posOFF1S = posOFF0E + 1
posOFF1E = posOFF0E + popSize
posOFF2S = posOFF1E + 1
posOFF2E = posOFF1E + popSize
posOFF3S = posOFF2E + 1
posOFF3E = posOFF2E + popSize
posOFF4S = posOFF3E + 1
posOFF4E = posOFF3E + popSize
posOFF5S = posOFF4E + 1
posOFF5E = posOFF4E + popSize
println("posOFF1S: ",posOFF1S,"; posOFF1E: ",posOFF1E)
println("posOFF2S: ",posOFF2S,"; posOFF2E: ",posOFF2E)
println("posOFF3S: ",posOFF3S,"; posOFF3E: ",posOFF3E)
println("posOFF4S: ",posOFF4S,"; posOFF4E: ",posOFF4E)
println("posOFF5S: ",posOFF5S,"; posOFF5E: ",posOFF5E)

# FileName :
PedAll = "PedAll.txt"           # pedigree  file with all animals
PheAll = "PheAll.txt"           # phenotype file with all animsla
GenAll = "GenAll.txt"           # genotype  file with all animals

Gen = "Gen.txt"                 # genotype file with all progeny in G5 and all sires in each generation
Phe = "Phe.txt"                 # phenotype file with all animals in G1 to G4
QTL = "QTL.txt"                 # QTL with with all progeny in G5 and all sires in each generation
Mar = "Mar.txt"                 # Marker with with all progeny in G5 and all sires in each generation
GenNF = "GenNF.txt"             # remove fixed genes from genotype file
QTLNF = "QTLNF.txt"             # remove fixed genes from QTL file
MarNF = "MarNF.txt"             # remove fixed genes from Marker file
;
```

```
posOFF1S: 8001; posOFF1E: 16000
posOFF2S: 16001; posOFF2E: 24000
posOFF3S: 24001; posOFF3E: 32000
posOFF4S: 32001; posOFF4E: 40000
posOFF5S: 40001; posOFF5E: 48000
```

In [14]:
```
fileName = "SS"
pedText = fileName * ".ped"
genText = fileName * ".gen"
pheText = fileName * ".phe"
;
```

In [15]:
```
;rm $pedText $genText $pheText
```

```
rm: cannot remove 'SS.ped': No such file or directory
rm: cannot remove 'SS.gen': No such file or directory
rm: cannot remove 'SS.phe': No such file or directory
```

# Sampling Founders

In [16]:
```
sires = XSim.sampleFounders("/home/nicole/Jupyter/HERdata/data/Clean/newdata/ch1to10.2000SNP.group1")
dams  = XSim.sampleFounders("/home/nicole/Jupyter/HERdata/data/Clean/newdata/ch1to10.2000SNP.group2");
```

```
Sampling 360 animals into base population.
Sampling 361 animals into base population.
```

# Base population

In [17]:
```
baseSires, baseDams, baseGen = XSim.sampleRan(popSizeBase, nGenBase, sires, dams, gen=gen);
```

```
Generation     1: sampling  4000 males and  4000 females
Generation     2: sampling  4000 males and  4000 females
Generation     3: sampling  4000 males and  4000 females
Generation     4: sampling  4000 males and  4000 females
Generation     5: sampling  4000 males and  4000 females
```

# Sample animals for sire and dam candidates

In [18]:
```
popSP = XSim.sampleRan(popSizeSP, 1, baseSires, baseDams, gen=baseGen,fileName=fileName);
```

```
Generation     6: sampling  4000 males and  4000 females
```

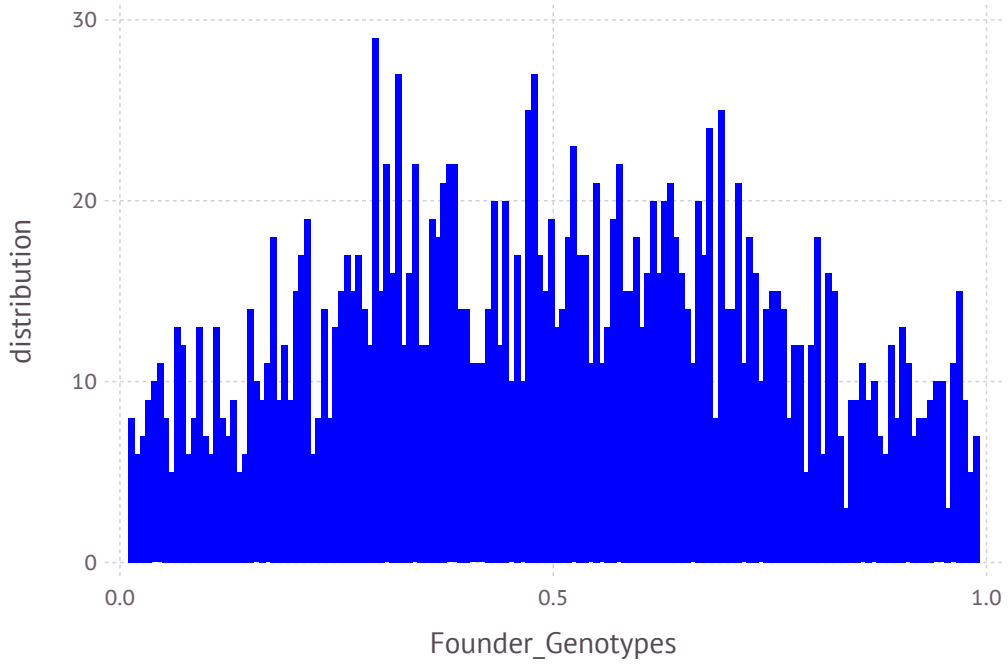
In [19]:
```
gSPSire = XSim.getOurGenotypes(popSP[1])
gSPDam = XSim.getOurGenotypes(popSP[2])
gSP = [gSPSire;gSPSire];
```

In [20]:
```
FCM = mean(gSP/2,1)
```

Out[20]:
```
1x2000 Array{Float64,2}:
 0.065  0.842875  0.274875  0.942125  …  0.271125  0.37675  0.4335  0.2665
```

In [21]:
```
plot(x=FCM, Geom.histogram, Guide.XLabel("Founder_Genotypes"), Guide.YLabel("distribution"), Theme(defaul
```

Out[21]:




In [22]:
```
V=var(gSP,1)
Mark=gSP[:,V.>0]
corMat=cor(Mark)
nRows =size(corMat,1)
LDMat =zeros(nRows-1,20);
```

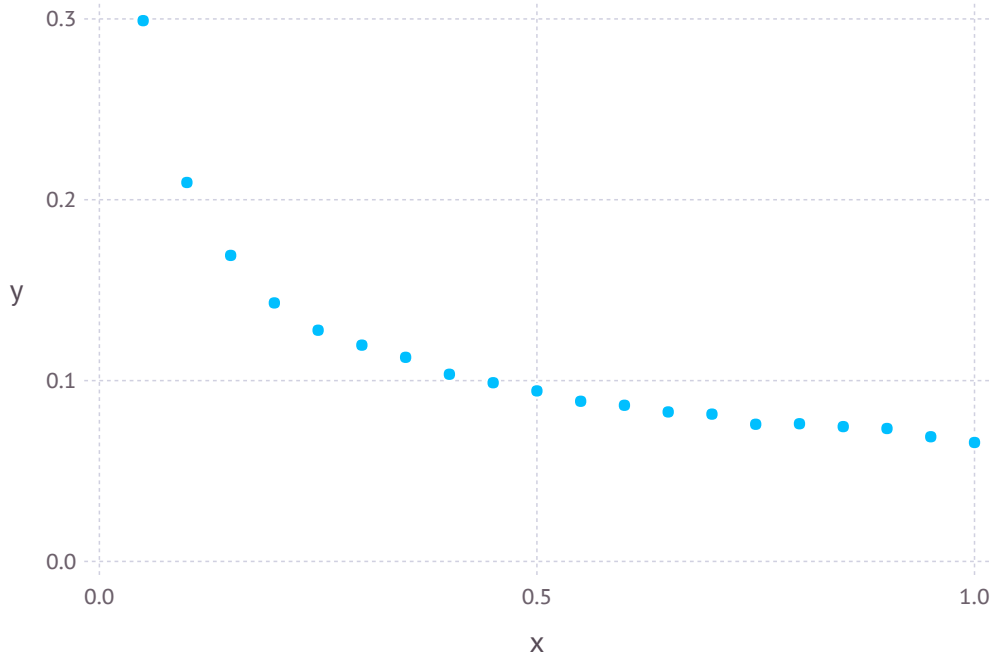
In [23]:
```
for i=1:(nRows-20)
LDMat[i,:] = corMat[i,(i+1):(i+20)].^2
end
```

In [24]:
```
y=mean(LDMat,1)
sort(y,2)
```

Out[24]:
```
1x20 Array{Float64,2}:
 0.0657981  0.0689755  0.0735029  0.074554  …  0.169239  0.209546  0.29905
```

In [25]:
```
plot(x=(1:20)/20*1,y=y)
```

Out[25]:




In [26]:
```
FCMstream = open("SNPCMF.txt", "w")
```

Out[26]:
```
IOStream(<file SNPCMF.txt>)
```

In [27]:
```
for i in 1:size(FCM,2)
    @printf(FCMstream, "%6.4f ", FCM[1,i])
end
```

In [28]:
```
close(FCMstream)
```

# Selection - increase

In [29]:
```
aSPSire = XSim.getOurGenVals(popSP[1])
aSPDam = XSim.getOurGenVals(popSP[2])
aSP = [aSPSire;aSPDam];
```

In [30]:
```
mean(aSP)
```

Out[30]:
```
30.509906324587273
```

In [31]:
```
varGen=var(aSP)
```

Out[31]:
```
0.8521122407768282
```

In [32]:
```
XSim.common.varRes = varGen    #heritability = 0.5
```

Out[32]:
```
0.8521122407768282
```

In [33]:
```
varRes = XSim.common.varRes
```

Out[33]:
```
0.8521122407768282
```

In [34]:
```
popRMP = XSim.sampleSelNoLap(nSires, nDams, nGener,popSP[1],popSP[2],gen=popSP[3],fileName=fileName,direc
```

```
Generation     7: sampling  4000 males and  4000 females
Generation     8: sampling  4000 males and  4000 females
Generation     9: sampling  4000 males and  4000 females
Generation    10: sampling  4000 males and  4000 females
Generation    11: sampling  4000 males and  4000 females
```

In [35]:
```
ymRMP = XSim.getOurGenVals(popRMP[1])    # for males: pop[1]
mean(ymRMP)
```

Out[35]: 33.547039982897395

In [36]:
```
yfRMP = XSim.getOurGenVals(popRMP[2])    # for females: pop[2]
mean(yfRMP)
```

Out[36]: 33.54591655317192

In [37]:
```
amRMP = XSim.getOurGenVals(popRMP[1])
var(amRMP)
```

Out[37]: 0.7774742683331888

In [38]:
```
afRMP = XSim.getOurGenVals(popRMP[2])
var(afRMP)
```

Out[38]: 0.758271611033663

# Pedigree: All animals

In [39]:
```
PED = convert(Array,readtable(pedText,separator=' ',header=false))
```

Out[39]:
```
48000x3 Array{Int64,2}:
 40722  35546  39331
 40723  33361  39194
 40724  34366  38935
 40725  33997  39172
 40726  35732  39197
 40727  36149  37859
 40728  36289  37721
 40729  33967  40698
 40730  34942  38609
 40731  34423  40533
 40732  33254  37872
 40733  33760  37290
 40734  36242  38261
     ⋮
 88710  75978  79399
 88711  75992  80624
 88712  73925  79389
 88713  75866  79270
 88714  74328  78979
 88715  73738  78179
 88716  72869  78890
 88717  76689  80277
 88718  74172  78475
 88719  75367  79040
 88720  75164  78775
 88721  73925  78021
```

In [40]:
```
PEDstream = open(PedAll, "w")
```

Out[40]:
```
IOStream(<file PedAll.txt>)
```

In [41]:
```
for i in 1:size(PED,1)
    @printf(PEDstream, "%19d%19d%19d \n", PED[i,1], PED[i,2], PED[i,3])
end
```

In [42]:
```
close(PEDstream)
```

# Create matrix of `genotype" covariates for all animals

In [43]:
```
nObs = countlines(pedText)
```

Out[43]: 48000

In [44]:
```
nMarker = numChr*numLoci
```

Out[44]: 2000

In [45]:
```
GT = convert(Array,readtable(genText,separator=' ',header=false))
```

Out[45]: 48000x2001 Array{Int64,2}:

| | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 40722 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 0 | 2 | … | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 40723 | 0 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 0 | 0 | 2 | 1 |
| 40724 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | 2 | 1 | 2 | 2 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 40725 | 0 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | | 1 | 2 | 1 | 1 | 1 | 0 | 1 | 2 | 2 | 1 | 1 | 0 |
| 40726 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | 0 | 2 | 0 | 0 | 1 | 1 | 2 | 2 | 0 | 0 | 2 | 1 |
| 40727 | 0 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | … | 1 | 2 | 1 | 1 | 0 | 1 | 2 | 2 | 1 | 0 | 1 | 0 |
| 40728 | 0 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 40729 | 1 | 2 | 1 | 2 | 1 | 0 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 40730 | 0 | 2 | 0 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 0 | 1 | 2 | 2 | 1 | 0 | 2 | 2 |
| 40731 | 0 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | | 1 | 2 | 1 | 1 | 0 | 0 | 2 | 2 | 1 | 0 | 1 | 0 |
| 40732 | 0 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | … | 2 | 1 | 2 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 40733 | 0 | 1 | 1 | 1 | 2 | 0 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 0 | 1 | 2 | 2 | 1 | 0 | 2 | 2 |
| 40734 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| ⋮ | | | | ⋮ | | | | ⋮ | ⋱ | | | ⋮ | | | | | ⋮ | | | | | ⋮ | |
| 88710 | 1 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 1 | 0 | 1 | 1 |
| 88711 | 1 | 2 | 0 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | | 2 | 0 | 2 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 88712 | 1 | 2 | 0 | 2 | 2 | 1 | 0 | 1 | 2 | 0 | … | 2 | 1 | 2 | 2 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 88713 | 0 | 2 | 1 | 2 | 1 | 0 | 0 | 0 | 2 | 0 | | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 88714 | 0 | 1 | 1 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | | 2 | 0 | 2 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 88715 | 0 | 1 | 1 | 1 | 2 | 0 | 1 | 1 | 1 | 1 | | 2 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 88716 | 0 | 1 | 2 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | | 2 | 1 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 88717 | 1 | 2 | 0 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | … | 2 | 1 | 2 | 1 | 0 | 1 | 2 | 2 | 1 | 0 | 1 | 0 |
| 88718 | 0 | 2 | 0 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 0 | 0 | 2 | 2 | 0 | 0 | 2 | 2 |
| 88719 | 0 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | | 0 | 2 | 1 | 1 | 0 | 1 | 2 | 2 | 1 | 0 | 1 | 1 |
| 88720 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | | 2 | 0 | 2 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 88721 | 0 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 1 | 0 | 1 | 1 |

In [46]:
```
allID = GT[:,1];
```

In [47]:
```
GTM = GT[:,2:end];
```

# Create marker file for all animals

In [48]:
```
M = GTM     # maker file for Julia
```

Out[48]:
```
48000x2000 Array{Int64,2}:
 1  1  1  2  2  1  2  2  0  2  2  0  1  …  0  2  0  0  0  0  1  1  0  1  1  0
 0  2  0  2  2  0  0  0  2  0  0  2  2     1  2  1  1  1  1  2  2  0  0  2  1
 0  1  1  1  1  0  1  1  1  1  1  1  2     2  1  2  2  0  0  1  1  1  1  0  0
 0  2  0  2  2  0  0  0  2  0  0  0  0     1  2  1  1  1  0  1  2  2  1  1  0
 0  1  1  1  1  0  1  1  1  1  1  1  2     0  2  0  0  1  1  2  2  0  0  2  1
 0  2  0  2  2  0  0  0  2  0  0  1  1  …  1  2  1  1  0  1  2  2  1  0  1  0
 0  1  1  2  2  1  1  1  1  1  1  0  1     1  1  1  1  1  1  1  1  0  1  1  1
 1  2  1  2  1  0  1  1  1  1  1  1  0     1  1  1  1  0  1  1  1  1  1  1  1
 0  2  0  2  2  1  1  1  1  1  1  1  2     1  1  1  1  0  1  2  2  1  0  2  2
 0  2  0  2  2  0  0  0  2  0  0  0  0     1  2  1  1  0  0  2  2  1  0  1  0
 0  2  0  2  2  0  0  0  2  0  0  1  1  …  2  1  2  1  0  0  1  1  0  1  1  0
 0  1  1  1  2  0  1  1  1  1  1  1  1     1  1  1  1  0  1  2  2  1  0  2  2
 0  1  1  1  1  0  1  1  1  1  1  0  1     1  1  1  1  0  0  1  1  0  1  1  0
 ⋮                 ⋮                 ⋮        ⋱        ⋮                 ⋮
 1  2  0  2  2  0  0  0  2  0  0  1  1     2  2  2  2  0  2  2  2  1  0  1  1
 1  2  0  2  2  0  1  1  1  1  1  1  2     2  0  2  1  0  0  1  1  0  1  1  0
 1  2  0  2  2  1  0  1  2  0  0  0  0  …  2  1  2  2  0  1  1  1  0  1  1  1
 0  2  1  2  1  0  0  0  2  0  0  2  1     1  1  1  1  0  0  1  1  0  1  1  0
 0  1  1  2  2  0  1  1  1  1  1  0  1     2  0  2  1  0  0  1  1  0  1  1  0
 0  1  1  1  2  0  1  1  1  1  1  0  0     2  0  1  1  0  1  1  1  1  1  1  1
 0  1  2  1  0  0  1  1  1  1  1  1  1     2  1  2  2  0  1  1  1  1  1  0  0
 1  2  0  2  2  0  1  1  1  1  1  0  0  …  2  1  2  1  0  1  2  2  1  0  1  0
 0  2  0  2  2  0  1  1  1  1  1  1  2     1  1  1  1  0  0  2  2  0  0  2  2
 0  2  0  2  2  0  0  0  2  0  0  1  1     0  2  1  1  0  1  2  2  1  0  1  1
 1  1  1  1  2  1  1  1  1  1  1  1  0     2  0  2  1  0  0  1  1  0  1  1  0
 0  1  1  2  2  1  1  1  1  1  1  1  2     2  2  2  2  0  2  2  2  1  0  1  1
```

In [49]:
```
Mstream = open(GenAll, "w")
```

Out[49]:
```
IOStream(<file GenAll.txt>)
```

In [50]:
```
for i in 1:size(M,1)
    @printf(Mstream, "%19d", allID[i])
    for j in 1:size(M,2)
        @printf(Mstream, "%3d", M[i,j])
    end
    @printf(Mstream, "\n")
end
```

In [51]:
```
close(Mstream)
```

# Genotypes - all sires and all offsprings in G5

In [52]:
```
AllSire = PED[posOFF1S:posOFF5E,2]
```

Out[52]:
```
40000-element Array{Int64,1}:
 42874
 43040
 43555
 41347
 44302
 43846
 42497
 41879
 43253
 44560
 41365
 43426
 42394
     ⋮
 75978
 75992
 73925
 75866
 74328
 73738
 72869
 76689
 74172
 75367
 75164
 73925
```

In [53]:
```
SireID = unique(AllSire)
```

Out[53]:
```
1000-element Array{Int64,1}:
 42874
 43040
 43555
 41347
 44302
 43846
 42497
 41879
 43253
 44560
 41365
 43426
 42394
     ⋮
 73365
 75017
 76019
 76168
 76438
 75387
 76612
 74839
 76689
 76378
 73325
 74511
```

In [54]:
```
OFF5 = PED[posOFF5S:posOFF5E,1]
```

Out[54]:
```
8000-element Array{Int64,1}:
 80722
 80723
 80724
 80725
 80726
 80727
 80728
 80729
 80730
 80731
 80732
 80733
 80734
     ⋮
 88710
 88711
 88712
 88713
 88714
 88715
 88716
 88717
 88718
 88719
 88720
 88721
```

In [55]:
```
SireOFF5ID = [SireID;OFF5]
```

Out[55]:
```
9000-element Array{Int64,1}:
 42874
 43040
 43555
 41347
 44302
 43846
 42497
 41879
 43253
 44560
 41365
 43426
 42394
     ⋮
 88710
 88711
 88712
 88713
 88714
 88715
 88716
 88717
 88718
 88719
 88720
 88721
```

In [56]:
```
SOFF5ID= DataFrame()
SOFF5ID[:ID] = SireOFF5ID;
```

In [57]:
```
typeof(SOFF5ID)
```

Out[57]:
```
DataFrames.DataFrame
```

In [58]:
```
MT = readtable(GenAll,separator=' ',header=false);
```

In [59]:
```
rename!(MT,:x1,:ID);
```

In [60]:
```
GSOFF5 = join(SOFF5ID, MT, on = :ID, kind = :inner);
```

In [61]:
```
size(GSOFF5)
```

Out[61]: (9000,2001)

In [62]:
```
GSOFF5Row = size(GSOFF5,1)
```

Out[62]: 9000

In [63]:
```
GSOFF5Col = size(GSOFF5,2)
```

Out[63]: 2001

In [64]:
```
GSOFF5stream = open(Gen, "w")
```

Out[64]: IOStream(<file Gen.txt>)

In [65]:
```
for i in 1:size(GSOFF5,1)
    for j in 1
        @printf(GSOFF5stream, "%19d", GSOFF5[i,j])
    end
    for k in 2:size(GSOFF5,2)
        @printf(GSOFF5stream, "%3d", GSOFF5[i,k])
    end
    @printf(GSOFF5stream, "\n")
end
```

In [66]:
```
close(GSOFF5stream)
```

# Phenotypes - All animals

In [67]:
```
PBV = convert(Array,readtable(pheText,separator=' ',header=false))
```

Out[67]:
```
48000x3 Array{Real,2}:
 40722  31.651  30.413
 40723  29.475  29.633
 40724  31.682  31.224
 40725  30.625  30.725
 40726  30.901  30.733
 40727  31.636  30.79
 40728  31.681  31.603
 40729  31.171  31.29
 40730  30.603  31.351
 40731  30.98   30.335
 40732  31.04   31.242
 40733  28.619  30.317
 40734  30.235  31.748
     ⋮
 88710  33.982  35.608
 88711  34.901  34.958
 88712  32.913  34.121
 88713  35.968  34.335
 88714  34.295  34.498
 88715  35.702  34.885
 88716  34.323  35.91
 88717  36.427  35.542
 88718  34.482  34.891
 88719  34.236  34.193
 88720  34.313  34.007
 88721  31.922  33.89
```

In [68]:
```
PBVstream = open(PheAll, "w")
```

Out[68]:
```
IOStream(<file PheAll.txt>)
```

In [69]:
```
for i in 1:size(PBV,1)
    @printf(PBVstream, "%19d %10.3f %10.3f \n", PBV[i,1], PBV[i,2], PBV[i,3])
end
```

In [70]:
```
close(PBVstream )
```

# Phenotypes - all animnls from G0 to G4

In [71]:
```
OFFG0toG4 = PED[posOFF0S:posOFF4E,1]
```

Out[71]:
```
40000-element Array{Int64,1}:
 40722
 40723
 40724
 40725
 40726
 40727
 40728
 40729
 40730
 40731
 40732
 40733
 40734
     ⋮
 80710
 80711
 80712
 80713
 80714
 80715
 80716
 80717
 80718
 80719
 80720
 80721
```

In [72]:
```
OFFG0toG4ID= DataFrame()
OFFG0toG4ID[:ID] = OFFG0toG4;
```

In [73]:
```
typeof(OFFG0toG4ID)
```

Out[73]:
```
DataFrames.DataFrame
```

In [74]:
```
AllPBV= readtable(pheText,separator=' ',header=false);
```

In [75]:
```
rename!(AllPBV,:x1,:ID)
head(AllPBV);
```

In [76]:
```
OFFG0toG4PBV = join(OFFG0toG4ID, AllPBV, on = :ID, kind = :inner);
```

In [77]:
```
Row = size(OFFG0toG4PBV,1)
```

Out[77]:
```
40000
```

In [78]:
```
Col = size(OFFG0toG4PBV,2)
```

Out[78]:
```
3
```

In [79]:
```
Phestream = open(Phe, "w")
```

Out[79]:
```
IOStream(<file Phe.txt>)
```

In [80]:
```
for i in 1:size(OFFG0toG4PBV,1)
    @printf(Phestream, "%19d %10.3f %10.3f \n", OFFG0toG4PBV[i,1], OFFG0toG4PBV[i,2], OFFG0toG4PBV[i,3])
end
```

In [81]:
```
close(Phestream)
```

# Get files with QTL only or Markers only

## QTL file and Markers file

In [82]:
```
QTLPos = XSim.common.G.qtl_index
```

Out[82]:
```
500-element Array{Int64,1}:
    4
   13
   15
   18
   19
   23
   25
   30
   35
   40
   44
   52
   55
    ⋮
 1963
 1968
 1969
 1973
 1982
 1989
 1991
 1993
 1994
 1996
 1998
 1999
```

In [83]:
```
k = size(M,2)
MarkerPos = deleteat!(collect(1:k),sort(QTLPos))
```

Out[83]:
```
1500-element Array{Int64,1}:
    1
    2
    3
    5
    6
    7
    8
    9
   10
   11
   12
   14
   16
    ⋮
 1981
 1983
 1984
 1985
 1986
 1987
 1988
 1990
 1992
 1995
 1997
 2000
```

## Genotype codes: 0, 1, 2

In [84]:
```
IDgen = convert(Array,readtable(Gen,separator=' ',header=false));
```

In [85]:
```
onlyID = IDgen[:,1]
QTLMarker = IDgen[:, 2:end]
```

Out[85]: 9000x2000 Array{Int64,2}:

| | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 2 | 1 | 1 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 2 | … | 1 | 1 | 1 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 2 | 1 |
| 0 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 1 |
| 0 | 0 | 2 | 2 | 2 | 1 | 2 | 2 | 0 | 2 | 2 | 0 | 2 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | 0 | 2 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 2 | 1 |
| 1 | 2 | 0 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | … | 1 | 2 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | | 2 | 2 | 2 | 2 | 0 | 1 | 2 | 2 | 1 | 0 | 1 | 1 |
| 0 | 2 | 0 | 2 | 2 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 2 | | 1 | 1 | 1 | 1 | 0 | 0 | 2 | 2 | 0 | 0 | 2 | 1 |
| 0 | 2 | 0 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| 0 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | 2 | 1 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | … | 0 | 2 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 2 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | | 0 | 2 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 2 | 2 |
| 0 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| ⋮ | | | | ⋮ | | | | | ⋮ | | | | ⋱ | | | ⋮ | | | | | ⋮ | | | | |
| 1 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 1 | 0 | 1 | 1 |
| 1 | 2 | 0 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | | 2 | 0 | 2 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 2 | 0 | 2 | 2 | 1 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | … | 2 | 1 | 2 | 2 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 2 | 1 | 2 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 2 | 1 | | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | 2 | 0 | 2 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | | 2 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 2 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | 2 | 1 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 2 | 0 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | … | 2 | 1 | 2 | 1 | 0 | 1 | 2 | 2 | 1 | 0 | 1 | 0 |
| 0 | 2 | 0 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | | 1 | 1 | 1 | 1 | 0 | 0 | 2 | 2 | 0 | 0 | 2 | 2 |
| 0 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | | 0 | 2 | 1 | 1 | 0 | 1 | 2 | 2 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | 2 | 0 | 2 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 1 | 0 | 1 | 1 |

In [86]:
```
onlyQTL = QTLMarker[:,QTLPos]
```

Out[86]:
```
9000x500 Array{Int64,2}:
 1  2  1  0  0  1  1  2  0  0  2  2  2  …  2  0  0  0  0  1  1  2  2  2  0  2
 2  2  1  1  0  1  1  1  0  0  1  2  1     1  2  2  2  0  2  2  0  2  2  0  0
 2  2  1  0  0  2  0  0  1  0  2  1  2     1  1  1  1  0  1  1  1  1  1  1  1
 2  1  1  0  0  0  0  0  1  0  1  2  0     1  1  1  2  2  0  0  0  0  2  0  2
 2  0  2  0  1  0  0  1  1  0  1  1  1     2  0  0  0  0  1  1  0  0  1  1  1
 1  1  1  1  0  1  1  1  0  0  2  2  1  …  1  1  1  1  0  1  1  0  1  1  1  0
 2  1  1  1  0  1  1  1  0  0  1  2  2     0  1  2  2  0  2  2  0  1  2  0  1
 2  2  0  2  0  0  2  2  0  2  2  2  2     1  2  2  2  0  1  1  0  0  2  0  2
 2  2  1  1  0  0  0  1  0  0  1  2  2     2  0  0  0  0  2  2  0  0  0  2  0
 2  0  2  1  0  0  0  1  1  0  0  1  1     2  1  1  1  0  2  2  0  1  1  1  0
 1  1  2  0  0  0  0  1  1  0  2  2  1  …  2  2  2  2  2  0  0  0  0  2  0  2
 1  2  1  1  0  0  0  1  0  0  1  2  2     2  0  0  0  0  0  0  2  2  2  0  2
 2  0  2  0  0  1  1  1  1  0  2  1  1     2  0  0  0  0  1  1  1  1  1  1  1
 ⋮              ⋮              ⋮        ⋱     ⋮              ⋮
 2  1  2  0  0  0  0  1  1  0  2  1  1     1  2  2  2  0  2  2  0  2  2  0  1
 2  2  1  0  0  1  0  1  0  0  2  2  1     2  1  1  1  0  2  2  0  0  1  1  1
 2  0  2  1  1  2  1  1  0  0  2  2  1  …  2  0  0  1  0  2  2  0  1  1  1  1
 2  1  1  1  0  0  0  0  2  0  1  1  0     2  1  1  1  1  1  1  0  0  1  1  1
 2  1  1  0  0  1  1  1  1  0  2  1  1     2  1  1  1  0  2  2  0  0  1  1  1
 1  0  2  1  0  1  1  1  1  1  2  1  0     2  0  1  0  0  2  1  0  1  1  1  1
 1  1  2  0  0  0  0  1  1  0  1  2  1     1  1  1  1  0  2  2  0  1  1  1  0
 2  0  2  0  1  1  1  2  1  0  2  2  1  …  2  2  2  2  0  2  2  0  1  2  0  1
 2  2  0  1  0  1  0  0  0  0  1  2  1     2  1  1  2  1  1  1  0  0  2  0  2
 2  1  1  1  0  1  1  1  0  0  1  2  2     2  0  0  2  2  0  1  0  1  2  0  1
 1  0  2  2  0  1  0  1  0  1  1  2  1     2  2  2  2  0  2  2  0  0  1  1  1
 2  2  0  1  0  1  0  0  1  0  1  1  2     1  1  1  2  0  2  2  0  2  2  0  1
```

In [87]:
```
onlyMar = QTLMarker[:,MarkerPos];
```

In [88]:
```
QTLstream = open(QTL, "w")
Marstream = open(Mar, "w");
```

In [89]:
```
for i in 1:size(onlyID,1)
    @printf(QTLstream, "%19d", onlyID[i])
    for j in 1:size(onlyQTL,2)
        @printf(QTLstream, "%3d", onlyQTL[i,j])
    end
    @printf(QTLstream, "\n")
end
```

In [90]:
```
for i in 1:size(onlyID,1)
    @printf(Marstream, "%19d", onlyID[i])
    for j in 1:size(onlyMar,2)
        @printf(Marstream, "%3d", onlyMar[i,j])
    end
    @printf(Marstream, "\n")
end
```

In [91]:
```
close(QTLstream)
close(Marstream)
```

# Remove Fixed Gene from the panel

In [92]:
```
VQM = var(QTLMarker,1)
QMnoFixed = QTLMarker[:,VQM .> 0]
VQ = var(onlyQTL,1)
QnoFixed = onlyQTL[:,VQ .> 0]
VM = var(onlyMar,1)
MnoFixed = onlyMar[:,VM .> 0];
```

In [93]:
```
GenNFstream = open(GenNF, "w")
QTLNFstream = open(QTLNF, "w")
MarNFstream = open(MarNF, "w");
```

In [94]:
```
for i in 1:size(onlyID,1)
    @printf(GenNFstream, "%19d", onlyID[i])
    for j in 1:size(QMnoFixed,2)
        @printf(GenNFstream, "%3d", QMnoFixed[i,j])
    end
    @printf(GenNFstream, "\n")
end
```

In [95]:
```
for i in 1:size(onlyID,1)
    @printf(QTLNFstream, "%19d", onlyID[i])
    for j in 1:size(QnoFixed,2)
        @printf(QTLNFstream, "%3d", QnoFixed[i,j])
    end
    @printf(QTLNFstream, "\n")
end
```

In [96]:
```
for i in 1:size(onlyID,1)
    @printf(MarNFstream, "%19d", onlyID[i])
    for j in 1:size(MnoFixed,2)
        @printf(MarNFstream, "%3d", MnoFixed[i,j])
    end
    @printf(MarNFstream, "\n")
end
```

In [97]:
```
close(GenNFstream)
close(QTLNFstream)
close(MarNFstream)
```

# Check heritability

In [98]:
```
P = AllPBV[:,2]
BV = AllPBV[:,3];
```

In [99]:
```
VP = var(P)
VBV = var(BV)
H = VBV/VP
```

Out[99]: 0.676326218354534

In [100]:
```
cor=cor(P,BV)
```

```
WARNING: imported binding for cor overwritten in module Main
```

Out[100]: 0.8218940576898819

In [101]:
```
QTLAll = M[:,QTLPos]
```

Out[101]: 48000x500 Array{Int64,2}:

| | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 2 | 0 | 1 | 1 | 0 | 1 | 2 | 0 | 2 | 1 | 1 | … | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 0 | 2 |
| 1 | 2 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 2 | 2 | | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 0 | 0 | 1 | 1 | 0 |
| 2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 0 | 0 | | 2 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 2 | 1 | 1 |
| 1 | 2 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 | 2 | | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 2 |
| 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | … | 0 | 1 | 1 | 2 | 1 | 1 | 1 | 0 | 1 | 2 | 0 | 1 |
| 2 | 1 | 2 | 0 | 1 | 0 | 0 | 1 | 2 | 0 | 1 | 2 | 1 | | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 0 | 2 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 2 | 2 | | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 2 | 2 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 2 | | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 0 | 1 | 2 | 0 | 2 |
| 2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 0 | 0 | | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 0 | 0 | 2 | 0 | 1 |
| 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 2 | 1 | … | 1 | 2 | 2 | 2 | 0 | 2 | 2 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 2 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 2 | 2 | | 2 | 2 | 2 | 2 | 0 | 1 | 1 | 0 | 1 | 2 | 0 | 2 |
| 1 | 1 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 2 | 2 | | 2 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| ⋮ | | | | ⋮ | | | | | | ⋮ | | | ⋱ | | | ⋮ | | | | ⋮ | | | | |
| 2 | 1 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 1 | 1 | | 1 | 2 | 2 | 2 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 1 |
| 2 | 2 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 2 | 2 | 1 | | 2 | 1 | 1 | 1 | 0 | 2 | 2 | 0 | 0 | 1 | 1 | 1 |
| 2 | 0 | 2 | 1 | 1 | 2 | 1 | 1 | 0 | 0 | 2 | 2 | 1 | … | 2 | 0 | 0 | 1 | 0 | 2 | 2 | 0 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 1 | 0 | | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 2 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 2 | 1 | 1 | | 2 | 1 | 1 | 1 | 0 | 2 | 2 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 2 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 0 | | 2 | 0 | 1 | 0 | 0 | 2 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 2 | 1 | | 1 | 1 | 1 | 1 | 0 | 2 | 2 | 0 | 1 | 1 | 1 | 0 |
| 2 | 0 | 2 | 0 | 1 | 1 | 1 | 2 | 1 | 0 | 2 | 2 | 1 | … | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 0 | 1 | 2 | 0 | 1 |
| 2 | 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | | 2 | 1 | 1 | 2 | 1 | 1 | 1 | 0 | 0 | 2 | 0 | 2 |
| 2 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 2 | 2 | | 2 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 1 | 2 | 0 | 1 |
| 1 | 0 | 2 | 2 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 2 | 1 | | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 0 | 0 | 1 | 1 | 1 |
| 2 | 2 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 2 | | 1 | 1 | 1 | 2 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 1 |

In [102]:
```
QTLo=qtlEffects[QTLPos]
```

Out[102]:
```
500-element Array{Float64,1}:
 0.0642648
 0.0640454
 0.0621757
 0.0634435
 0.0636793
 0.0631503
 0.062927
 0.063941
 0.0638137
 0.0622705
 0.0627602
 0.062701
 0.0631664
 ⋮
 0.0630197
 0.062977
 0.0630884
 0.0629478
 0.0638766
 0.0629622
 0.0636136
 0.0626055
 0.062736
 0.0628504
 0.0636843
 0.063674
```

In [103]:
```
EAlpha=QTLAll*QTLo
```

Out[103]:
```
48000-element Array{Float64,1}:
 30.4273
 29.6549
 31.2549
 30.7449
 30.7427
 30.8118
 31.6199
 31.3047
 31.3761
 30.349
 31.2477
 30.3434
 31.7523
  ⋮
 35.6205
 34.9904
 34.169
 34.3405
 34.5319
 34.9073
 35.9242
 35.5498
 34.9166
 34.215
 34.018
 33.9059
```

In [104]:
```
meanEAlphaG0=mean(EAlpha[1:8000])   # our mu_g
```

Out[104]:
```
30.528678189113815
```

In [105]:
```
meanEAlphaG1=mean(EAlpha[8001:16000])
```

Out[105]:
```
31.195052527189713
```

In [106]:
```
meanEAlphaG2=mean(EAlpha[16001:24000])
```

Out[106]:
```
31.81614667763852
```

In [107]:
```
meanEAlphaG3=mean(EAlpha[24001:32000])
```

Out[107]:
```
32.3899707437827
```

In [108]:
```
meanEAlphaG4=mean(EAlpha[32001:40000])
```

Out[108]:
```
32.974364620737845
```

In [109]:
```
meanEAlphaG5=mean(EAlpha[40001:48000])
```

Out[109]:
```
33.56843768689131
```

In [110]:
```
EAlphaG=onlyQTL*QTLo
```

Out[110]:
```
9000-element Array{Float64,1}:
 31.8153
 31.5846
 32.6374
 31.6186
 31.9566
 31.106
 31.2513
 32.4595
 32.4517
 31.1717
 31.6106
 31.6049
 31.5535
  ⋮
 35.6205
 34.9904
 34.169
 34.3405
 34.5319
 34.9073
 35.9242
 35.5498
 34.9166
 34.215
 34.018
 33.9059
```

In [111]:
```
meanEAlphaG=mean(EAlphaG)
```

Out[111]: 33.503329422059984

In [112]:
```
meanBreedingValueSelected=meanEAlphaG-meanEAlphaG0   # Legarra mu_g
```

Out[112]: 2.9746512329461687

In [113]:
```
meanEAlphaS0=mean(EAlphaG[1:200])
```

Out[113]: 31.82035477549589

In [114]:
```
meanBreedingValueSelectedG0=meanEAlphaS0-meanEAlphaG0
```

Out[114]: 1.2916765863820743

In [115]:
```
meanEAlphaS1=mean(EAlphaG[201:400])
```

Out[115]: 32.44198058070412

In [116]:
```
meanBreedingValueSelectedG1=meanEAlphaS1-meanEAlphaG0
```

Out[116]: 1.9133023915903031

In [117]:
```
meanEAlphaS2=mean(EAlphaG[401:600])
```

Out[117]: 32.9577739638879

In [118]:
```
meanBreedingValueSelectedG2=meanEAlphaS2-meanEAlphaG0
```

Out[118]: 2.429095774774087

In [119]:
```
meanEAlphaS3=mean(EAlphaG[601:800])
```

Out[119]: 33.55463545609624

In [120]:
```
meanBreedingValueSelectedG3=meanEAlphaS3-meanEAlphaG0
```

Out[120]: 3.0259572669824273

In [121]:
```
meanEAlphaS4=mean(EAlphaG[801:1000])
```

Out[121]: 34.137571740862775

In [122]:
```
meanBreedingValueSelectedG4=meanEAlphaS4-meanEAlphaG0
```

Out[122]: 3.6088935517489595

In [123]:
```
meanEAlphaS5=mean(EAlphaG[1001:9000])
```

Out[123]: 33.56843768689131

In [124]:
```
meanBreedingValueSelectedG5=meanEAlphaS5-meanEAlphaG0
```

Out[124]: 3.039759497777492