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; 20 loci per chromosome = > 200 Loci (50 QTL & 150 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 = 0.01
numLoci   = 20
nQTL      = 5
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)
```

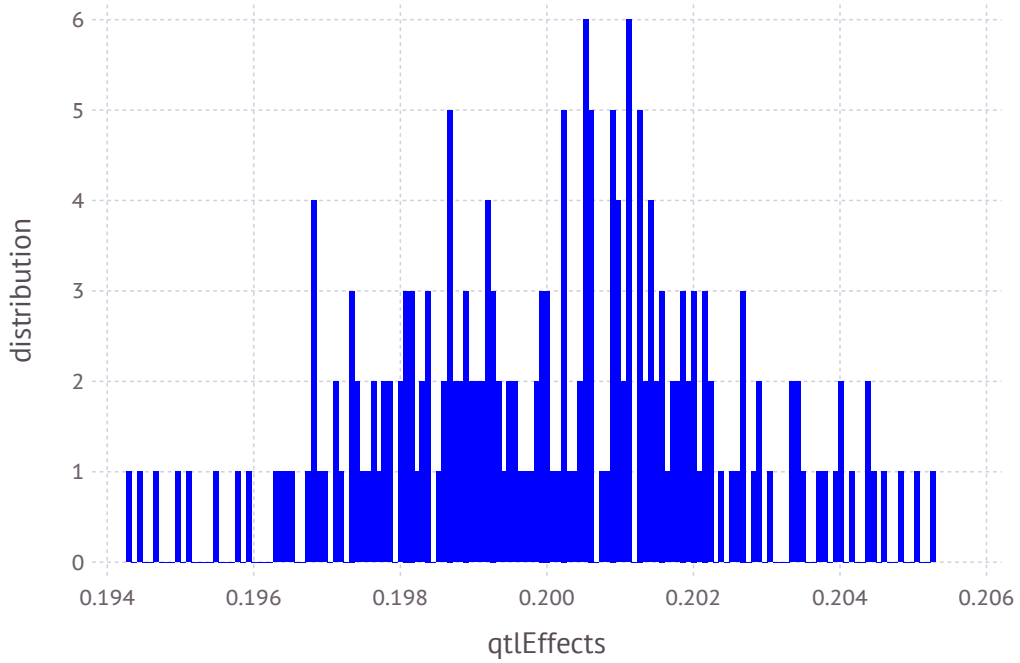
In [7]:
```
qtlEffects
```

Out[7]:
```
200-element Array{Float64,1}:
 0.201152
 0.198672
 0.201138
 0.19776
 0.201271
 0.203908
 0.198159
 0.200579
 0.19767
 0.200142
 0.199163
 0.194937
 0.199465
 ⋮
 0.198502
 0.203057
 0.202204
 0.200972
 0.20023
 0.201836
 0.19875
 0.200463
 0.197442
 0.198136
 0.198684
 0.201671
```

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]: 0.20004530304676574

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

Out[11]: 5.115645668748497e-6

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/ch1to10.200SNP.group1")
dams  = XSim.sampleFounders("/home/nicole/Jupyter/HERdata/data/Clean/newdata/ch1to10.200SNP.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
```

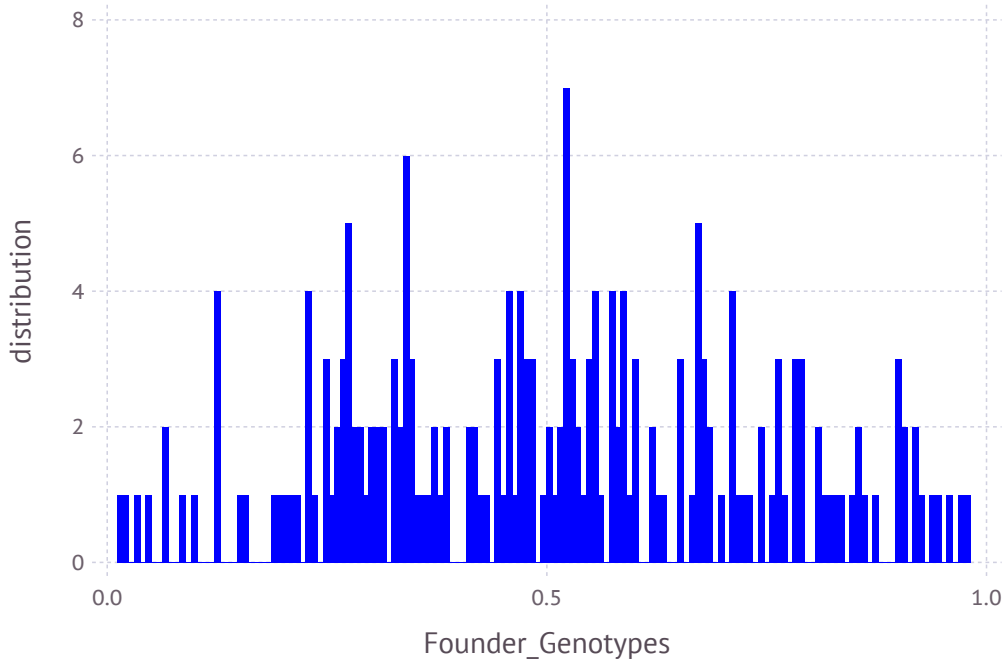
In [19]:
```
gSPSire = XSim.getOurGenotypes(popSP[1])
gSPDam = XSim.getOurGenotypes(popSP[2])
gSP = [gSPSire;gSPSire];
```

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

Out[20]:
```
1x200 Array{Float64,2}:
 0.064125  0.837  0.2885  0.9385  0.8105  …  0.383125  0.89725  0.548875
```

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);
```

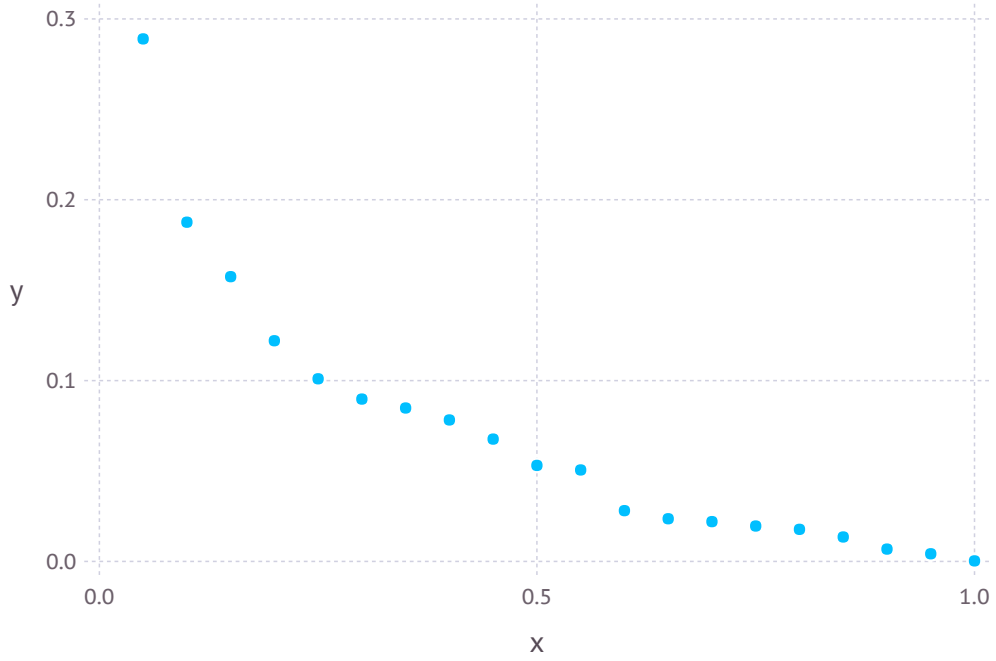
In [23]:
```
for i=1:(nRows-20)
LDMat[i,:] = corMat[i,(i+1):(i+20)].^2
end
```

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

Out[24]:
```
1x20 Array{Float64,2}:
 0.000332359  0.00423222  0.006842  0.0135364  …  0.157466  0.187579  0.28897
```

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]:
```
10.18754734694228
```

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

Out[31]:
```
0.6995423116598276
```

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

Out[32]:
```
0.6995423116598276
```

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

Out[33]:
```
0.6995423116598276
```

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]: 12.804359119363216

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

Out[36]: 12.779371000900447

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

Out[37]: 0.5408355873202002

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

Out[38]: 0.5638221627232005

# Pedigree: All animals

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

Out[39]:
```
48000x3 Array{Int64,2}:
 40722  35613  39323
 40723  33032  37731
 40724  35601  39782
 40725  34965  37722
 40726  33036  38434
 40727  33315  40594
 40728  33916  38577
 40729  34719  40690
 40730  32781  38749
 40731  35277  36871
 40732  33114  40343
 40733  34411  40464
 40734  36690  39754
     ⋮
 88710  76719  78252
 88711  73335  79460
 88712  75461  80312
 88713  75571  80388
 88714  75379  79205
 88715  75508  79468
 88716  75195  80481
 88717  76709  80218
 88718  73413  80120
 88719  76217  78226
 88720  76071  78206
 88721  75094  80347
```

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]: 200

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

Out[45]: 48000x201 Array{Int64,2}:

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

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]:
```
48000x200 Array{Int64,2}:
 1  2  1  2  1  0  1  1  1  1  1  1  0  …  2  2  2  0  2  2  0  2  2  0  2  0
 1  2  0  2  2  1  1  1  1  1  1  1  2     2  2  1  1  1  2  1  1  1  1  2  1
 0  2  0  2  2  1  1  1  1  1  1  1  2     2  0  0  2  0  2  1  0  0  1  1  2
 1  2  0  2  2  0  1  1  1  1  1  0  0     1  2  2  1  1  2  0  2  2  0  2  1
 0  1  1  2  2  0  1  1  1  1  1  0  0     2  1  1  1  2  2  0  1  1  1  2  0
 0  2  1  2  1  0  0  0  2  0  0  1  0  …  2  1  0  2  1  2  0  0  0  0  2  2
 0  2  0  2  2  1  1  1  1  1  1  0  1     2  1  1  1  1  2  0  1  1  0  2  1
 0  0  2  0  0  0  0  0  2  0  0  0  0     0  0  2  2  0  0  2  0  0  2  2  2
 0  2  1  2  1  0  0  0  2  0  0  1  0     0  0  2  2  0  0  2  0  0  2  2  2
 0  0  2  2  2  1  2  2  0  2  2  0  1     2  2  2  0  2  2  0  2  2  0  2  0
 0  2  0  2  2  0  0  0  2  0  0  1  1  …  2  2  1  1  1  2  1  1  1  1  2  1
 0  1  1  1  1  0  1  1  1  1  1  1  2     2  2  1  1  2  2  0  1  1  0  2  1
 0  2  1  2  1  1  1  1  1  1  1  1  1     2  1  1  1  1  2  0  1  1  0  2  1
 ⋮                 ⋮                 ⋮        ⋱        ⋮                 ⋮
 0  1  1  2  2  0  1  1  1  1  1  1  2     2  0  1  2  2  2  0  0  0  2  1  1
 0  2  0  2  2  0  0  0  2  0  0  2  2     2  2  2  2  2  2  0  0  0  2  2  1
 0  2  0  2  2  0  1  1  2  0  0  2  2  …  2  1  2  1  2  2  0  1  1  1  1  1
 0  2  1  2  1  0  0  0  2  0  0  1  0     2  0  1  2  2  2  0  0  0  2  1  1
 0  2  0  2  2  0  0  0  2  0  0  1  1     2  0  2  2  2  2  0  0  0  2  0  2
 0  2  0  2  2  0  0  0  2  0  0  2  2     2  2  1  2  1  2  0  0  0  1  2  2
 0  1  1  1  1  0  1  1  2  0  0  2  2     2  1  1  2  1  2  0  0  0  1  1  2
 0  2  0  2  2  0  0  0  2  0  0  2  1  …  2  0  1  2  2  2  0  0  0  2  1  1
 0  2  0  2  2  0  0  0  2  0  0  2  2     2  1  1  2  2  2  0  0  0  2  2  1
 0  2  0  2  2  0  0  0  2  0  0  2  2     0  0  2  2  0  0  2  0  0  2  2  2
 0  2  0  2  2  0  1  1  2  0  0  2  2     2  1  1  1  1  2  0  1  1  0  2  1
 0  2  1  2  1  0  0  0  2  0  0  2  1     2  2  2  0  2  2  0  2  2  0  2  0
```

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}:
 41834
 44217
 41523
 43583
 44366
 43853
 43891
 42256
 41633
 41984
 43083
 41594
 44168
     ⋮
 76719
 73335
 75461
 75571
 75379
 75508
 75195
 76709
 73413
 76217
 76071
 75094
```

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

Out[53]:
```
1000-element Array{Int64,1}:
 41834
 44217
 41523
 43583
 44366
 43853
 43891
 42256
 41633
 41984
 43083
 41594
 44168
    ⋮
 75706
 75070
 73804
 76658
 74202
 76261
 76488
 75154
 73359
 75161
 76073
 75458
```

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}:
 41834
 44217
 41523
 43583
 44366
 43853
 43891
 42256
 41633
 41984
 43083
 41594
 44168
     ⋮
 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]: (9000,201)

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

Out[62]: 9000

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

Out[63]: 201

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  10.978  10.148
 40723   9.282   9.366
 40724   9.472  11.935
 40725   8.862  10.965
 40726   8.591   9.349
 40727  12.162  11.13
 40728   9.745  10.165
 40729  10.865   9.936
 40730   7.872   9.913
 40731   8.851   9.194
 40732  10.388  11.526
 40733  10.54   10.163
 40734  10.777   9.548
     ⋮
 88710  12.63   13.327
 88711  12.856  13.953
 88712  13.18   12.941
 88713  13.265  12.754
 88714  13.658  13.352
 88715  13.881  14.141
 88716  15.059  13.93
 88717  14.609  14.147
 88718  13.705  12.712
 88719  12.0    11.516
 88720  13.381  13.132
 88721  13.143  13.135
```

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]:
```
50-element Array{Int64,1}:
   6
   9
  12
  13
  18
  26
  29
  32
  33
  38
  46
  49
  52
   ⋮
 153
 158
 166
 169
 172
 173
 178
 186
 189
 192
 193
 198
```

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

Out[83]:
```
150-element Array{Int64,1}:
   1
   2
   3
   4
   5
   7
   8
  10
  11
  14
  15
  16
  17
   ⋮
 184
 185
 187
 188
 190
 191
 194
 195
 196
 197
 199
 200
```

## 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]: 9000x200 Array{Int64,2}:

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

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

Out[86]:
```
9000x50 Array{Int64,2}:
 0  2  1  0  1  1  1  0  2  1  2  1  1  …  2  1  2  0  2  0  0  2  2  1  2  0
 0  2  1  1  1  1  2  1  1  2  2  1  0     1  2  0  1  1  1  0  1  2  1  2  1
 0  2  1  1  1  1  2  1  1  2  1  1  2     2  1  2  1  0  0  2  1  1  1  1  1
 0  2  0  0  0  0  2  0  2  2  2  2  0     2  0  2  0  1  0  1  1  2  1  2  1
 0  2  1  1  1  0  2  1  1  1  2  1  1     2  1  0  1  2  0  1  1  2  2  2  2
 0  2  1  1  1  0  2  0  2  1  0  1  2  …  2  0  2  1  2  0  0  1  1  2  1  2
 0  2  1  0  1  0  2  0  2  2  2  2  0     1  1  1  0  1  0  1  0  1  2  0  1
 0  2  2  1  2  1  1  2  0  2  1  1  1     2  0  1  1  2  1  2  1  2  1  2  0
 1  1  1  2  0  0  2  1  1  1  0  0  2     2  0  0  2  1  0  1  1  2  1  2  1
 0  2  2  1  2  0  2  1  1  1  1  1  2     2  1  1  0  1  0  2  0  2  2  1  1
 1  1  0  1  0  2  2  2  0  2  2  1  1  …  1  2  1  1  1  0  1  0  1  2  1  2
 0  1  0  0  0  0  1  1  1  1  2  2  0     2  1  2  1  2  0  2  2  2  0  2  0
 0  1  1  1  0  0  2  2  0  0  1  2  1     2  0  2  0  1  0  1  1  2  1  1  0
 ⋮                 ⋮                 ⋮        ⋱        ⋮                 ⋮
 0  1  1  2  1  0  0  2  1  2  2  2  0     1  1  0  2  2  0  2  1  2  2  2  2
 0  2  2  2  2  1  2  1  1  2  1  1  2     2  1  1  1  1  0  2  2  2  2  2  2
 0  2  2  2  2  0  2  0  2  2  1  1  2  …  1  1  2  0  2  0  1  2  2  1  2  1
 0  2  1  0  1  0  1  1  1  2  2  2  1     2  2  1  1  2  0  2  1  2  2  2  2
 0  2  1  1  1  0  2  0  2  2  2  2  1     2  2  1  1  2  0  2  2  2  2  2  2
 0  2  2  2  2  1  2  2  0  1  2  2  1     2  2  1  1  1  0  1  2  2  2  1  1
 0  2  2  2  2  0  2  0  2  2  2  2  1     2  2  0  2  2  0  2  2  2  2  1  1
 0  2  2  1  2  0  1  1  1  2  2  2  0  …  1  2  1  1  2  0  2  1  2  2  2  2
 0  2  2  2  2  0  1  1  1  2  2  2  1     2  0  1  0  2  0  1  1  2  2  2  2
 0  2  2  2  1  0  2  1  1  1  1  1  1     2  0  1  1  2  1  2  0  0  2  0  2
 0  2  2  2  2  1  2  2  0  1  1  2  1     1  2  0  2  1  1  2  1  2  1  1  0
 0  2  2  1  2  1  2  1  1  2  1  2  1     1  2  0  2  2  1  2  2  2  0  2  0
```

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]: 0.6489095232621248

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

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

Out[100]:
```
0.8064011876698486
```

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

Out[101]:
```
48000x50 Array{Int64,2}:
 0  1  1  0  0  1  1  2  0  2  2  1  1  …  1  1  1  0  2  0  1  2  2  0  2  0
 1  1  1  2  1  1  1  0  2  1  0  1  0     1  1  1  1  1  1  1  2  2  1  1  1
 1  1  1  2  1  0  2  0  2  2  2  2  1     2  1  1  1  2  0  1  0  2  2  0  1
 0  1  0  0  0  0  1  1  1  2  2  2  0     2  1  2  0  2  0  1  1  1  1  1  0
 0  1  0  0  0  0  0  2  0  2  1  2  0     1  1  0  0  2  0  0  1  2  1  2  1
 0  2  1  0  1  0  1  1  1  2  2  2  1  …  2  0  2  0  1  0  2  0  2  2  1  0
 1  1  0  1  0  0  2  0  2  2  1  0  2     1  1  2  0  2  0  2  1  2  1  1  0
 0  2  0  0  0  0  1  2  0  1  2  2  0     1  1  2  1  1  0  0  0  0  2  0  2
 0  2  1  0  1  0  0  2  0  2  0  2  2     1  2  0  2  2  0  0  0  0  2  0  2
 1  0  0  1  0  0  2  1  1  1  1  1  0     2  2  2  0  1  0  2  2  2  0  2  0
 0  2  1  1  1  0  2  2  0  0  1  1  1  …  0  2  1  1  2  0  1  2  2  1  1  1
 0  1  1  2  1  1  2  0  1  1  1  1  2     1  2  1  0  0  0  1  1  2  1  2  0
 1  1  1  1  0  0  2  1  1  0  1  1  2     2  2  0  1  2  0  2  1  2  1  1  0
 ⋮              ⋮              ⋮           ⋱        ⋮              ⋮
 0  1  1  2  1  0  0  2  1  2  2  2  0     1  1  0  2  2  0  2  1  2  2  2  2
 0  2  2  2  2  1  2  1  1  2  1  1  2     2  1  1  1  1  0  2  2  2  2  2  2
 0  2  2  2  2  0  2  0  2  2  1  1  2  …  1  1  2  0  2  0  1  2  2  1  2  1
 0  2  1  0  1  0  1  1  1  2  2  2  1     2  2  1  1  2  0  2  1  2  2  2  2
 0  2  1  1  1  0  2  0  2  2  2  2  1     2  2  1  1  2  0  2  2  2  2  2  2
 0  2  2  2  2  1  2  2  0  1  2  2  1     2  2  1  1  1  0  1  2  2  2  1  1
 0  2  2  2  2  0  2  0  2  2  2  2  1     2  2  0  2  2  0  2  2  2  2  1  1
 0  2  2  1  2  0  1  1  1  2  2  2  0  …  1  2  1  1  2  0  2  1  2  2  2  2
 0  2  2  2  2  0  1  1  1  2  2  2  1     2  0  1  0  2  0  1  1  2  2  2  2
 0  2  2  2  1  0  2  1  1  1  1  1  1     2  0  1  1  2  1  2  0  0  2  0  2
 0  2  2  2  2  1  2  2  0  1  1  2  1     1  2  0  2  1  1  2  1  2  1  1  0
 0  2  2  1  2  1  2  1  1  2  1  2  1     1  2  0  2  2  1  2  2  2  0  2  0
```

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

Out[102]:
```
50-element Array{Float64,1}:
 0.203908
 0.19767
 0.194937
 0.199465
 0.201153
 0.197746
 0.200865
 0.197332
 0.197994
 0.202864
 0.201258
 0.198955
 0.204007
 ⋮
 0.202702
 0.195917
 0.200537
 0.201787
 0.201951
 0.204145
 0.200508
 0.203416
 0.198502
 0.200972
 0.20023
 0.198136
```

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

Out[103]:
```
48000-element Array{Float64,1}:
 10.2083
  9.39967
 12.0201
 11.0199
  9.41081
 11.2058
 10.2263
 10.0066
 10.0055
  9.22237
 11.6034
 10.1892
  9.61237
  ⋮
 13.4094
 14.019
 13.0194
 12.8099
 13.4127
 14.2098
 14.0095
 14.2014
 12.809
 11.6206
 13.2107
 13.2121
```

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

Out[104]:
```
10.253259241672637
```

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

Out[105]:
```
10.801497623189713
```

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

Out[106]:
```
11.348398237487046
```

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

Out[107]: 11.860926245351532

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

Out[108]: 12.374534092766527

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

Out[109]: 12.864323594441526

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

Out[110]:
```
9000-element Array{Float64,1}:
 10.812
 10.9995
 11.0094
 11.2142
 12.0115
 11.2158
 10.61
 11.4259
 10.6235
 11.9983
 11.8097
 11.6187
 10.2148
  ⋮
 13.4094
 14.019
 13.0194
 12.8099
 13.4127
 14.2098
 14.0095
 14.2014
 12.809
 11.6206
 13.2107
 13.2121
```

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

Out[111]: 12.809244183211781

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

Out[112]: 2.555984941539144

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

Out[113]: 11.347340973260025

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

Out[114]: 1.0940817315873872

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

Out[115]: 11.899976514991163

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

Out[116]: 1.6467172733185258

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

Out[117]: 12.356932631248242

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

Out[118]: 2.1036733895756043

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

Out[119]: 12.873249347826949

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

Out[120]: 2.6199901061543116

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

Out[121]: 13.365544999542767

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

Out[122]: 3.1122857578701293

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

Out[123]: 12.864323594441526

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

Out[124]: 2.6110643527688886