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 = 0.0
# 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
Va = nQTL*numChr*0.5              # Va= nQTL*2pq*mean(alpha)^2; mean(alpha) = 0
qtlEffects= rand(Normal(0, sqrt(1/Va)),numLoci*numChr)  # Let alpha mu = 0.0
XSim.init(numChr,numLoci,chrLength,geneFreq,mapPos,qtlMarker,qtlEffects,mutRate)
```

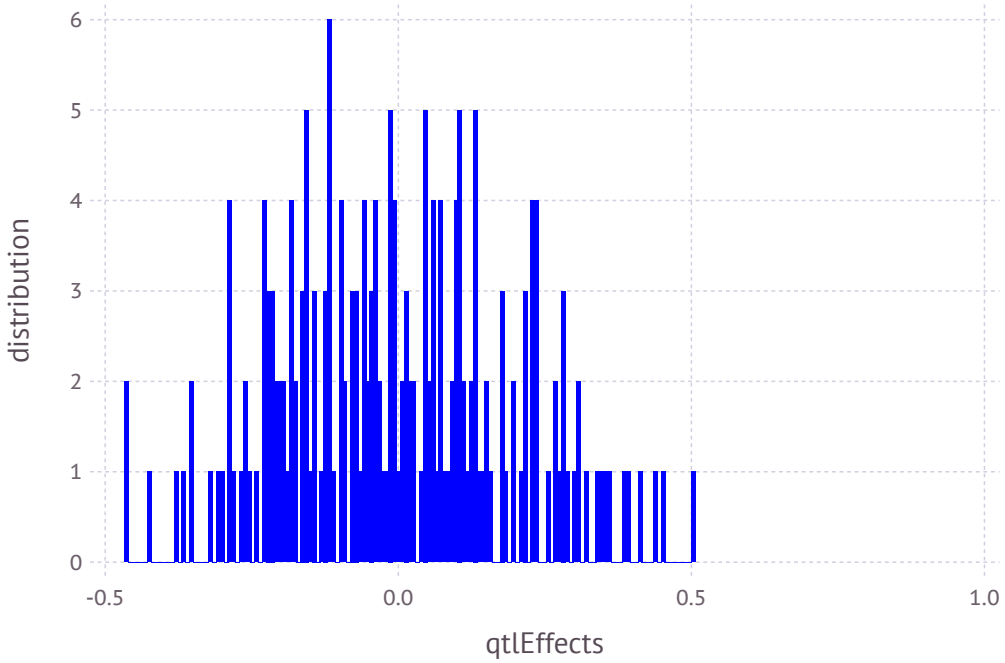
In [7]:
```
qtlEffects
```

Out[7]:
```
200-element Array{Float64,1}:
  0.227503
 -0.196956
 -0.14082
 -0.0449065
  0.231763
 -0.230334
  0.02785
  0.322847
  0.125204
  0.299269
  0.212986
 -0.0462921
 -0.075661
  ⋮
  0.278952
 -0.195483
  0.0502402
 -0.140364
 -0.117381
 -0.00960697
  0.17845
  0.128206
  0.101184
 -0.0596695
  0.266306
  0.133373
```

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

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

Out[9]:




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

Out[10]: -0.003082153472293272

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

Out[11]: 0.038152510746908265

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
```

# 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
```

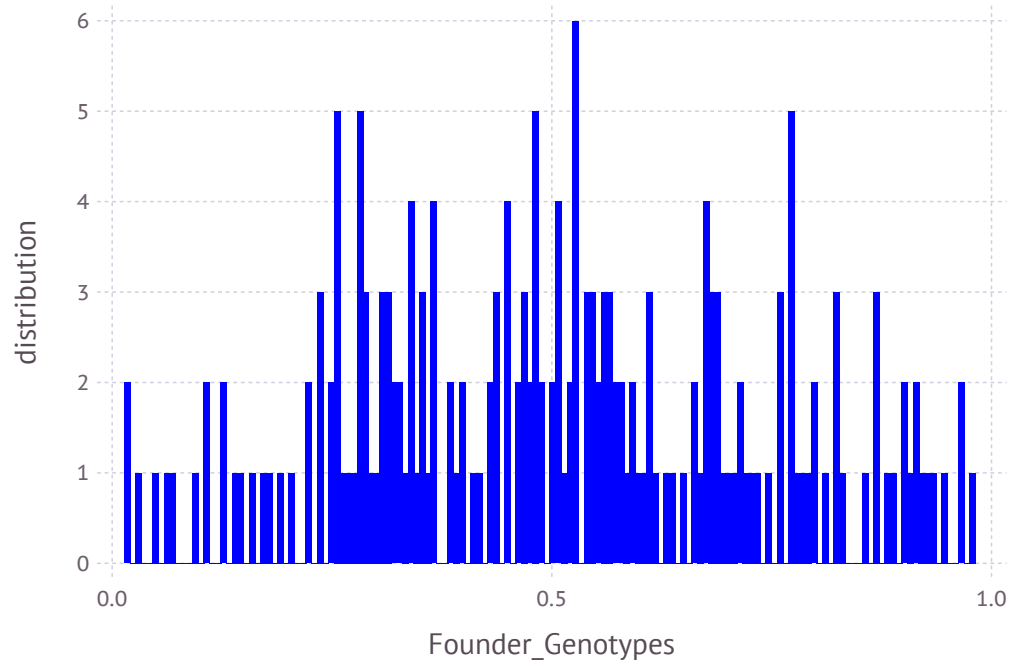
In [19]:
```
gSPSire = XSim.getOurGenotypes(popSP[1])
gSPDam = XSim.getOurGenotypes(popSP[2])
gSP = [gSPSire;gSPSire];
```

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

Out[20]:
```
1x200 Array{Float64,2}:
 0.06925  0.828375  0.28625  0.94375  …  0.367375  0.3985  0.89125  0.54925
```

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

Out[21]:




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

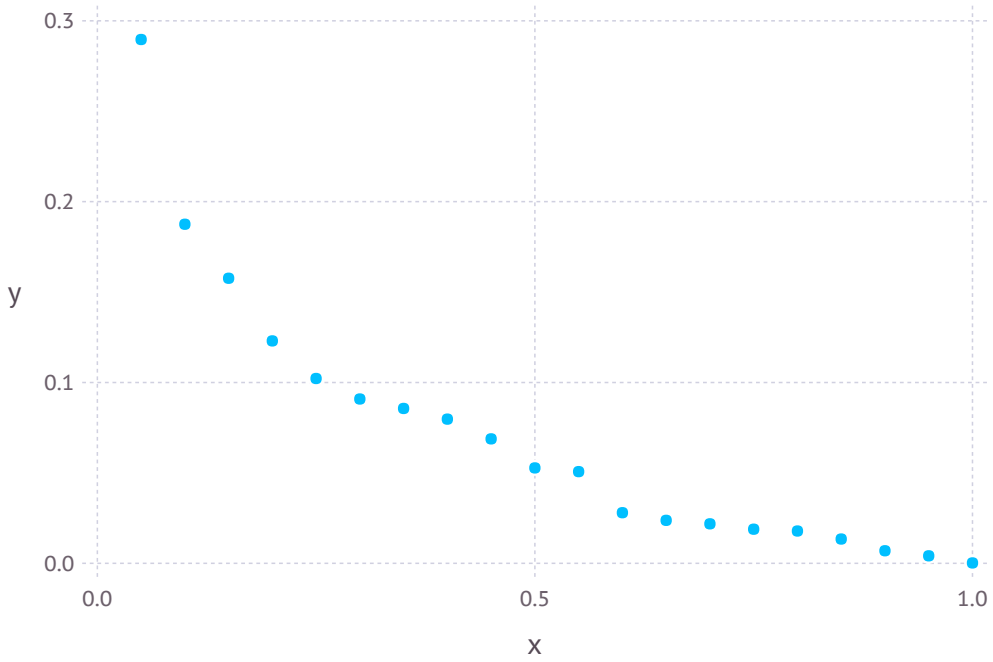
In [23]:
```
for i=1:(nRows-20)
LDMat[i,:] = corMat[i,(i+1):(i+20)].^2
end
```

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

Out[24]:
```
1x20 Array{Float64,2}:
 0.000228834  0.00417655  0.00698409  …  0.157631  0.187492  0.289686
```

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]:
```
11.542597399903466
```

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

Out[31]:
```
1.991062778306584
```

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

Out[32]:
```
1.991062778306584
```

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

Out[33]:
```
1.991062778306584
```

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

```
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]:
```
16.194961753652905
```

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

Out[36]:
```
16.19630994507969
```

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

Out[37]: 1.4315914028417034

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

Out[38]: 1.4423888144232253

# Pedigree: All animals

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

Out[39]:
```
48000x3 Array{Int64,2}:
 40722  36553  36972
 40723  33070  38212
 40724  32975  37847
 40725  34367  38842
 40726  36453  38619
 40727  36109  40147
 40728  36485  37402
 40729  34270  39701
 40730  35042  36893
 40731  35185  37356
 40732  34039  39155
 40733  34412  39612
 40734  36101  36745
     ⋮
 88710  75237  80256
 88711  75237  79148
 88712  76504  79769
 88713  75178  79615
 88714  74512  78963
 88715  76004  77996
 88716  75660  78723
 88717  75767  78624
 88718  75452  80423
 88719  76086  79735
 88720  75423  80131
 88721  73547  78849
```

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

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}:
 0  1  1  1  1  0  1  1  1  1  1  1  2  …  2  0  2  2  2  1  0  0  0  1  1  2
 0  1  2  2  2  0  1  1  1  1  1  0  0     1  2  2  2  1  2  0  0  0  2  1  2
 1  2  0  2  2  1  2  2  0  2  2  0  1     2  1  2  1  2  2  0  1  1  1  1  1
 1  2  0  2  2  0  1  1  1  1  1  0  0     1  0  1  2  0  1  1  0  0  1  2  2
 1  2  0  2  2  0  0  0  2  0  0  2  1     2  2  2  0  2  2  0  1  1  1  2  1
 0  2  0  2  2  0  1  1  2  0  0  0  0  …  1  1  2  1  1  1  1  1  1  1  2  1
 0  1  2  1  0  0  1  1  1  1  1  1  1     2  2  2  1  1  2  0  1  1  0  1  1
 0  2  0  2  2  1  1  1  1  1  1  0  1     0  0  2  2  0  0  2  0  0  2  2  2
 0  2  0  2  2  0  0  0  2  0  0  1  1     2  0  0  2  1  2  1  0  0  2  1  1
 0  1  1  2  2  0  1  1  1  1  1  0  0     2  1  1  1  1  2  0  1  1  0  2  1
 0  2  0  2  2  1  2  2  0  2  2  0  1  …  0  0  2  2  0  0  2  0  0  1  2  1
 0  2  0  2  2  0  0  0  2  0  0  1  1     1  0  1  2  0  1  1  0  0  1  2  2
 0  2  1  2  1  0  1  1  2  0  0  2  1     2  0  0  2  0  2  0  0  0  0  2  2
 ⋮                 ⋮                 ⋮        ⋱        ⋮                 ⋮
 0  1  1  2  2  0  1  1  1  1  1  0  1     1  1  2  2  1  1  1  0  0  1  2  1
 0  1  1  1  1  0  1  1  1  1  1  0  1     1  1  1  2  0  0  2  0  0  2  2  2
 0  1  2  2  1  1  1  1  1  1  1  1  1  …  1  1  2  1  1  1  1  1  1  1  2  1
 1  1  1  2  2  2  2  2  0  2  2  0  2     0  0  2  2  0  0  2  0  0  2  2  2
 0  2  0  2  2  1  1  1  1  1  1  0  1     0  0  2  2  0  0  2  0  0  2  2  2
 0  2  0  2  2  0  0  0  2  0  0  1  1     0  0  2  2  0  0  2  0  0  2  2  2
 1  2  0  2  2  0  1  1  1  1  1  0  0     0  0  2  2  0  0  2  0  0  2  2  2
 0  1  1  1  1  0  1  1  1  1  1  0  1  …  1  1  2  2  1  1  1  0  0  1  2  1
 1  2  0  2  2  1  1  1  0  2  2  0  2     2  1  1  1  1  2  0  1  1  0  2  1
 0  2  1  2  1  0  0  0  2  0  0  1  0     2  1  2  1  1  1  1  1  1  1  2  1
 0  0  2  2  2  2  2  2  0  2  2  0  2     1  1  2  1  1  1  1  1  1  1  2  1
 0  1  1  1  1  0  1  1  1  1  1  0  1     1  0  1  2  0  1  2  0  0  2  1  2
```

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}:
 41217
 41551
 41144
 43630
 44000
 44067
 42777
 42436
 42763
 41010
 41367
 44566
 44393
     ⋮
 75237
 75237
 76504
 75178
 74512
 76004
 75660
 75767
 75452
 76086
 75423
 73547
```

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

Out[53]:
```
1000-element Array{Int64,1}:
 41217
 41551
 41144
 43630
 44000
 44067
 42777
 42436
 42763
 41010
 41367
 44566
 44393
     ⋮
 75922
 76321
 76309
 73399
 74367
 72906
 76407
 75411
 74965
 76316
 76206
 75660
```

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}:
 41217
 41551
 41144
 43630
 44000
 44067
 42777
 42436
 42763
 41010
 41367
 44566
 44393
     ⋮
 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   7.9    10.233
 40723  11.351  11.583
 40724  13.423  13.833
 40725  10.616  11.155
 40726  12.936  11.752
 40727  10.612  12.266
 40728  12.648  11.965
 40729  12.243  11.875
 40730  13.112  13.394
 40731  11.862  12.352
 40732  11.491  11.66
 40733   9.731   9.923
 40734   8.656   9.117
     ⋮
 88710  16.045  16.199
 88711  14.293  17.183
 88712  17.92   16.937
 88713  17.75   18.824
 88714  18.171  17.484
 88715  17.792  16.863
 88716  15.538  16.647
 88717  14.964  16.802
 88718  15.597  17.604
 88719  17.232  17.385
 88720  15.919  17.998
 88721  17.802  17.244
```

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}:
   1
   2
   8
  15
  20
  21
  22
  28
  35
  40
  41
  42
  48
   ⋮
 155
 160
 161
 162
 168
 175
 180
 181
 182
 188
 195
 200
```

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

Out[83]:
```
150-element Array{Int64,1}:
   3
   4
   5
   6
   7
   9
  10
  11
  12
  13
  14
  16
  17
   ⋮
 186
 187
 189
 190
 191
 192
 193
 194
 196
 197
 198
 199
```

## 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 | … | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 |
| 0 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | 1 | 0 | 1 | 2 | 0 | 1 | 1 | 0 | 0 | 2 | 2 | 2 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 |
| 0 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | | 0 | 0 | 2 | 2 | 0 | 0 | 2 | 0 | 0 | 2 | 2 | 2 |
| 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | | 1 | 0 | 1 | 2 | 0 | 1 | 1 | 0 | 0 | 1 | 2 | 2 |
| 0 | 2 | 0 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | … | 1 | 0 | 2 | 2 | 1 | 0 | 1 | 0 | 0 | 1 | 2 | 2 |
| 0 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 0 | 2 | 2 | 0 | 1 | | 2 | 1 | 1 | 1 | 1 | 2 | 0 | 1 | 1 | 0 | 2 | 1 |
| 0 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | | 1 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 0 | 2 | 1 | 2 |
| 1 | 1 | 1 | 2 | 2 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 1 | | 2 | 1 | 1 | 1 | 1 | 2 | 0 | 1 | 1 | 0 | 2 | 1 |
| 1 | 2 | 0 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 0 | 0 | 2 | 2 | 1 |
| 0 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 1 | 1 | 1 | 2 | … | 2 | 2 | 2 | 1 | 2 | 2 | 0 | 1 | 1 | 1 | 2 | 1 |
| 0 | 2 | 0 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | | 2 | 2 | 1 | 1 | 1 | 2 | 0 | 1 | 1 | 0 | 2 | 1 |
| 0 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | 2 | 2 | 2 | 1 | 2 | 2 | 0 | 1 | 1 | 1 | 2 | 0 |
| ⋮ | | | | ⋮ | | | | | ⋮ | | | | ⋱ | | | ⋮ | | | | ⋮ | | | | |
| 0 | 1 | 1 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 0 | 0 | 1 | 2 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | 1 | 1 | 1 | 2 | 0 | 0 | 2 | 0 | 0 | 2 | 2 | 2 |
| 0 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | … | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 |
| 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 0 | 2 | | 0 | 0 | 2 | 2 | 0 | 0 | 2 | 0 | 0 | 2 | 2 | 2 |
| 0 | 2 | 0 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | 0 | 0 | 2 | 2 | 0 | 0 | 2 | 0 | 0 | 2 | 2 | 2 |
| 0 | 2 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | | 0 | 0 | 2 | 2 | 0 | 0 | 2 | 0 | 0 | 2 | 2 | 2 |
| 1 | 2 | 0 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | | 0 | 0 | 2 | 2 | 0 | 0 | 2 | 0 | 0 | 2 | 2 | 2 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | … | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 0 | 0 | 1 | 2 | 1 |
| 1 | 2 | 0 | 2 | 2 | 1 | 1 | 1 | 0 | 2 | 2 | 0 | 2 | | 2 | 1 | 1 | 1 | 1 | 2 | 0 | 1 | 1 | 0 | 2 | 1 |
| 0 | 2 | 1 | 2 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 |
| 0 | 0 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 0 | 2 | | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | 1 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 0 | 2 | 1 | 2 |

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

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

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]: 0.6965072519459544

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

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

Out[100]:
```
0.8316432419387182
```

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

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

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

```
Out[102]: 50-element Array{Float64,1}:
            0.227503
           -0.196956
            0.322847
            0.507281
            0.20907
            0.0724841
            0.100046
            0.435878
            0.0522533
            0.197039
            0.0807879
           -0.0468308
            0.139187
            ⋮
           -0.215442
           -0.0962679
           -0.122904
            0.00828767
           -0.0364038
            0.156938
           -0.0333883
            0.218515
            0.343076
           -0.220151
            0.17845
            0.133373
```

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

ut[103]:
```
48000-element Array{Float64,1}:
  0.745403
  2.41595
  1.65091
  2.5062
 -1.10393
  0.551287
  2.48379
 -0.156749
  0.956788
  0.876376
  2.81273
  1.26613
  2.27262
  ⋮
  1.0053
  2.25161
  1.52832
  2.28851
  1.0954
  0.532626
  2.24487
  1.37229
  1.62688
  1.89639
  2.14379
  2.49974
```

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

ut[104]:
```
1.1398426166587323
```

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

ut[105]:
```
1.254330361529967
```

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

ut[106]:
```
1.2964154624889954
```

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

```
ut[107]: 1.4060320003138391
```

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

```
ut[108]: 1.531002482173568
```

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

```
ut[109]: 1.630478655812346
```

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

```
ut[110]: 9000-element Array{Float64,1}:
           1.16214
           1.64526
           1.5464
           1.22543
           3.38504
           1.98393
           1.61698
           4.17898
           2.01678
           2.64612
          -0.144636
           1.50266
           0.0108551
           ⋮
           1.0053
           2.25161
           1.52832
           2.28851
           1.0954
           0.532626
           2.24487
           1.37229
           1.62688
           1.89639
           2.14379
           2.49974
```

n [111]:
```
meanEAlphaG=mean(EAlphaG)
```
ut[111]: 1.6194364147906135

n [112]:
```
meanBreedingValueSelected=meanEAlphaG-meanEAlphaG0   # Legarra mu_g
```
ut[112]: 0.4795937981318812

n [113]:
```
meanEAlphaS0=mean(EAlphaG[1:200])
```
ut[113]: 1.3744423085082917

n [114]:
```
meanBreedingValueSelectedG0=meanEAlphaS0-meanEAlphaG0
```
ut[114]: 0.23459969184955942

n [115]:
```
meanEAlphaS1=mean(EAlphaG[201:400])
```
ut[115]: 1.3496989185064496

n [116]:
```
meanBreedingValueSelectedG1=meanEAlphaS1-meanEAlphaG0
```
ut[116]: 0.2098563018477173

n [117]:
```
meanEAlphaS2=mean(EAlphaG[401:600])
```
ut[117]: 1.5237187593055506

n [118]:
```
meanBreedingValueSelectedG2=meanEAlphaS2-meanEAlphaG0
```
ut[118]: 0.38387614264681824

n [119]:
```
meanEAlphaS3=mean(EAlphaG[601:800])
```
ut[119]: 1.6646241147129985

n [120]:
```
meanBreedingValueSelectedG3=meanEAlphaS3-meanEAlphaG0
```
ut[120]: 0.5247814980542662

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

```
ut[121]: 1.743008332050455
```

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

```
ut[122]: 0.6031657153917227
```

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

```
ut[123]: 1.630478655812346
```

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

```
ut[124]: 0.49063603915361376
```