

```
In [1]: # Founders: real haplotype data (chlto10.200SNP)
# "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 = 0.1
# 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                                     #  $\alpha \sim N(100,1)$ 
        Va = nQTL*numChr*0.5*mu*mu                  #  $Va = nQTL * 2pq * \text{mean}(\alpha)^2$ 
        qtlEffects= rand(Normal(mu/sqrt(Va), sqrt(1/Va)),numLoci*numChr) # Let  $\alpha \mu = 0.115$ 
        XSim.init(numChr,numLoci,chrLength,geneFreq,mapPos,qtlMarker,qtlEffects,mutRate)
```

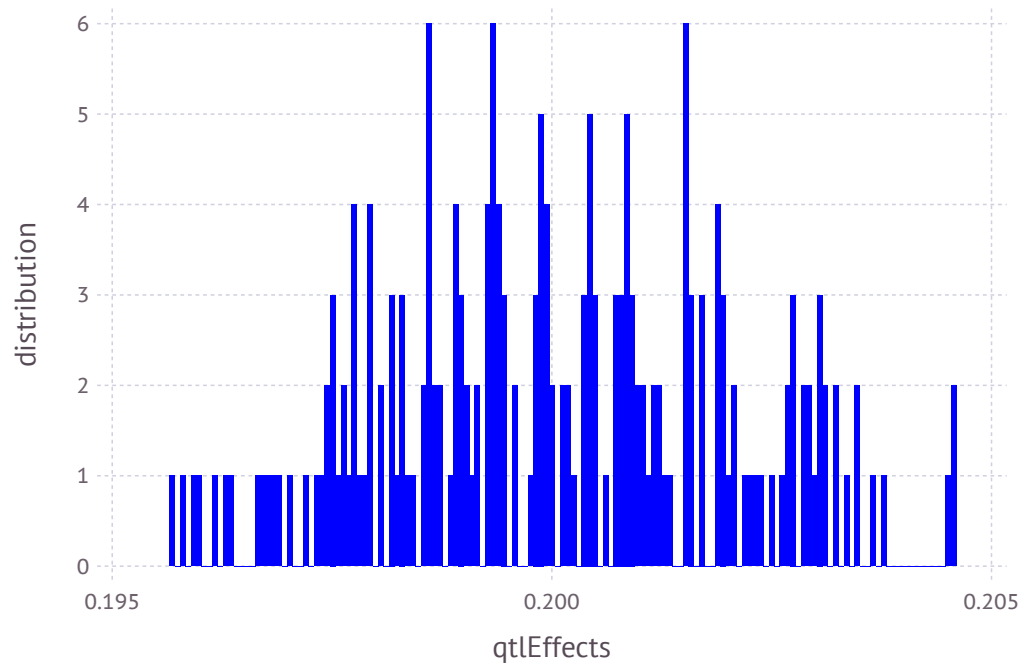
```
In [7]: qtlEffects
```

```
Out[7]: 200-element Array{Float64,1}:  
 0.198995  
 0.203064  
 0.197653  
 0.200809  
 0.197517  
 0.200617  
 0.197013  
 0.201567  
 0.197583  
 0.20287  
 0.200402  
 0.198905  
 0.199163  
  ⋮  
 0.197773  
 0.202043  
 0.204598  
 0.19784  
 0.200707  
 0.199857  
 0.202406  
 0.200917  
 0.198164  
 0.195653  
 0.201216  
 0.204511
```

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

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

Out[9]:



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

Out[10]: 0.20001651236341947

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

Out[11]: 3.888883280845798e-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 animals
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
```

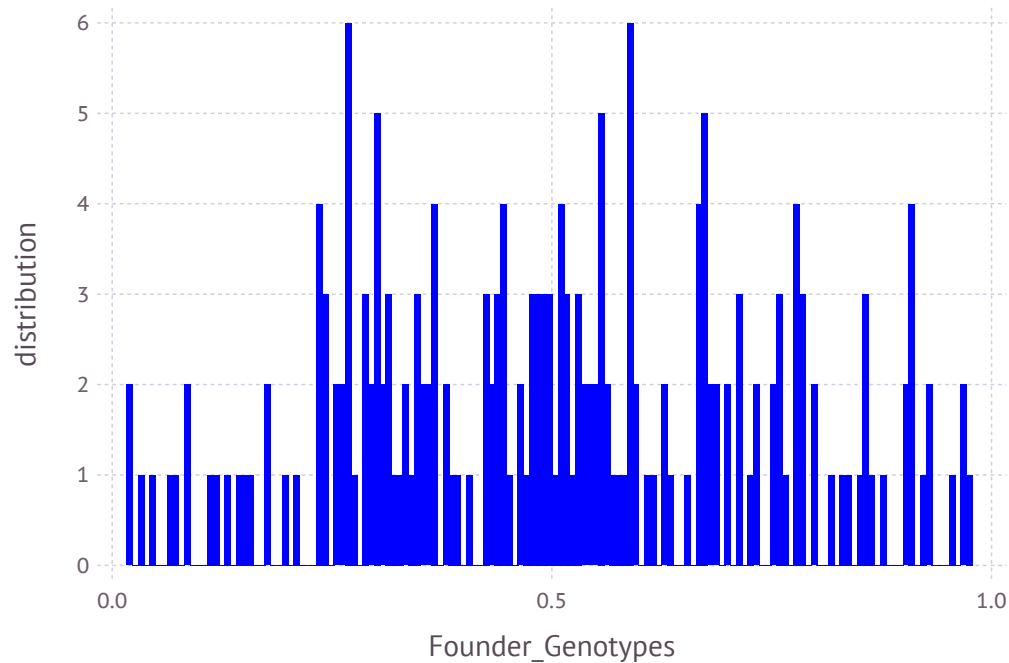
```
In [19]: gSPSire = XSim.getOurGenotypes(popSP[1])
gSPDam = XSim.getOurGenotypes(popSP[2])
gSP = [gSPSire;gSPSire];
```

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

```
Out[20]: 1x200 Array{Float64,2}:
 0.065  0.847625  0.26975  0.955  ...  0.368875  0.3795  0.90675  0.547625
```

```
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);
```



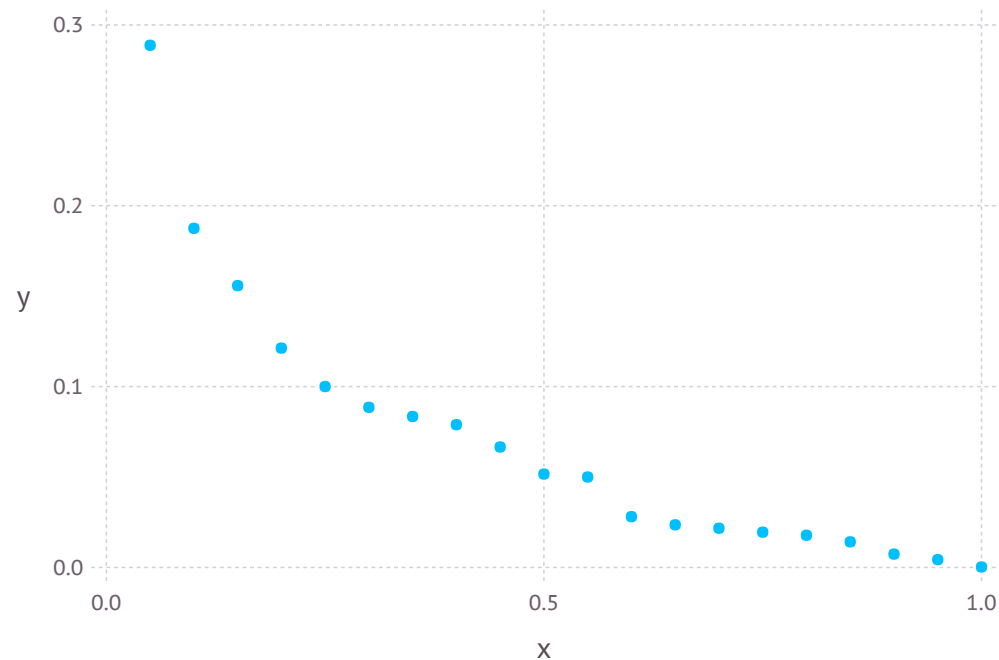
```
In [23]: for i=1:(nRows-20)
        LDMat[i,:] = corMat[i,(i+1):(i+20)].^2
        end
```

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

```
Out[24]: 1x20 Array{Float64,2}:
         0.000266301  0.00432225  0.00740378  ...  0.155865  0.187532  0.288747
```

```
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]: 9.430121577980165
```

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

```
Out[31]: 0.8234239152952353
```

```
In [32]: XSim.common.varRes = 9*varGen    #heritability = 0.1
```

```
Out[32]: 7.4108152376571175
```

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

```
Out[33]: 7.4108152376571175
```

```
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]: 11.096237766507999
```

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

```
Out[36]: 11.094400536135359
```

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

```
Out[37]: 0.7584138297717521
```

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

```
Out[38]: 0.7299143263700312
```

Pedigree: All animals

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

```
Out[39]: 48000x3 Array{Int64,2}:  
  40722  36139  38811  
  40723  32777  37467  
  40724  32925  39538  
  40725  34265  39588  
  40726  33930  40021  
  40727  33993  39593  
  40728  34371  37482  
  40729  33773  40386  
  40730  34366  39002  
  40731  34340  39840  
  40732  35629  37563  
  40733  33948  38987  
  40734  35733  40477  
      ⋮  
  88710  74632  78323  
  88711  76536  79413  
  88712  76172  79802  
  88713  74438  79566  
  88714  76445  79450  
  88715  75696  77460  
  88716  75935  79690  
  88717  76651  78819  
  88718  72743  78695  
  88719  75766  78328  
  88720  74738  80449  
  88721  73143  79695
```

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

```
 0  1  1  2  2  0  1  1  1  1  1  0  0  ...  2  1  0  2  0  2  1  0  0  1  1  2
 0  1  2  2  1  0  0  0  2  0  0  2  1  ...  2  2  2  1  1  1  0  1  1  0  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  0  2  2  0  0  0  2  0  0  2  2  ...  1  1  2  1  1  1  1  1  1  1  2  1
 0  1  1  2  2  1  0  0  2  0  0  2  2  ...  2  1  2  2  1  2  0  0  0  1  1  1
 0  1  1  2  2  0  1  1  1  1  1  0  0  ...  2  1  1  2  2  2  0  0  0  1  1  2
 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  1  1  1  1  1  0  1  ...  2  2  2  0  2  2  0  2  2  0  2  1
 0  2  0  2  2  0  1  1  1  1  1  1  2  ...  2  1  1  1  2  2  1  2  2  0  2  1
 0  1  1  2  2  0  1  1  1  1  1  1  1  ...  1  0  1  2  1  1  1  0  0  0  1  2
 1  2  0  2  2  0  0  0  2  0  0  1  0  ...  2  1  1  1  1  2  0  1  1  0  2  1
 0  2  0  2  2  0  0  0  2  0  0  0  0  ...  2  1  1  2  1  2  0  0  0  0  1  2
 0  2  0  2  2  0  0  0  2  0  0  0  0  ...  2  2  2  0  2  2  0  1  1  1  2  1
  ⋮           ⋮           ⋮           ⋮           ⋮
 0  2  1  2  1  1  1  1  1  1  1  1  ...  2  2  1  1  2  1  1  0  0  2  2  2
 0  1  1  1  2  0  1  1  1  1  1  1  1  ...  1  1  2  1  1  2  0  1  1  0  2  1
 0  2  1  2  1  0  1  1  1  1  1  1  1  ...  2  2  2  1  2  2  0  1  1  1  2  0
 1  2  0  2  2  0  1  1  1  1  1  1  1  ...  2  1  2  1  1  1  1  1  1  1  2  1
 0  2  0  2  2  1  1  1  1  1  1  1  2  ...  1  2  2  1  1  2  0  1  1  1  1  1
 1  2  0  2  2  1  1  1  1  1  1  1  1  ...  1  0  2  1  1  1  1  1  1  1  2  1
 0  2  0  2  2  0  0  0  2  0  0  2  2  ...  2  2  1  1  2  2  0  1  1  0  2  1
 1  2  0  2  2  1  2  2  0  2  2  0  1  ...  2  2  2  0  2  2  0  2  2  0  2  0
 0  2  1  2  1  0  0  0  2  0  0  1  0  ...  2  1  1  1  2  1  0  1  1  0  2  1
 0  2  1  2  1  1  1  1  1  1  1  1  1  ...  2  1  1  1  2  2  0  1  1  1  2  0
 0  2  2  2  0  0  1  1  1  1  1  1  1  ...  1  1  2  1  1  1  1  1  1  1  2  1
 1  2  0  2  2  1  1  1  1  1  1  1  2  ...  1  1  2  1  2  2  1  1  1  0  1  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}:
```

```
 41416
 42165
 43641
 42916
 41715
 43255
 44458
 43279
 41121
 42874
 41121
 40850
 42946
      ⋮
 74632
 76536
 76172
 74438
 76445
 75696
 75935
 76651
 72743
 75766
 74738
 73143
```



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

```
Out[53]: 1000-element Array{Int64,1}:
```

```
41416
42165
43641
42916
41715
43255
44458
43279
41121
42874
40850
42946
41782
⋮
75031
76114
75986
72987
73853
74080
75265
76111
76593
76396
73745
75791
```

```
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}:  
 41416  
 42165  
 43641  
 42916  
 41715  
 43255  
 44458  
 43279  
 41121  
 42874  
 40850  
 42946  
 41782  
      ⋮  
 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  11.126  10.636  
  40723   9.699   9.424  
  40724   8.818   9.426  
  40725   8.844   9.02  
  40726  11.404  10.815  
  40727   9.007   8.418  
  40728   9.582   9.811  
  40729   8.508   8.63  
  40730   7.244   9.016  
  40731   9.419   9.615  
  40732   7.253   8.216  
  40733   8.237   9.819  
  40734   9.414   9.022  
      ⋮  
  88710  13.776  12.228  
  88711  13.009  11.63  
  88712  15.258  12.438  
  88713  11.544  11.231  
  88714  14.058  13.248  
  88715  12.768  10.423  
  88716  11.813  11.437  
  88717   9.764  12.242  
  88718  11.217  11.836  
  88719  15.531  10.831  
  88720  12.795  11.226  
  88721  15.67   12.245
```

```
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 animals 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
10
11
12
20
26
30
31
32
40
46
50
51
⋮
152
160
166
170
171
172
180
186
190
191
192
200
```



```
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
 9
13
14
15
16
17
 ⋮
184
185
187
188
189
193
194
195
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  1  2  1  0  0  1  1  1  1  1  1  1  ...  2  2  2  0  2  2  0  2  2  0  2  0
 0  1  1  2  2  0  1  1  1  1  1  1  1  ...  2  1  1  2  1  2  0  0  0  2  1
 0  1  1  2  2  0  1  1  1  1  1  1  1  ...  1  1  2  1  1  1  1  1  1  2  1
 0  2  0  2  2  0  0  0  2  0  0  2  2  ...  2  1  1  2  1  1  0  0  0  2  2
 0  2  1  2  1  0  0  0  2  0  0  1  1  ...  2  0  1  2  0  1  1  0  0  1  1  2
 0  2  0  2  2  0  1  1  1  1  1  1  0  0  ...  2  1  2  1  2  2  0  1  1  1  1  1
 0  2  0  2  2  1  1  1  1  1  1  1  1  2  ...  1  2  2  1  1  2  0  1  1  1  1  1
 0  2  0  2  2  1  1  1  1  1  1  1  1  2  ...  0  0  2  2  0  0  2  0  0  2  2  2
 0  2  0  2  2  0  0  0  2  0  0  1  2  ...  1  0  2  2  1  2  0  0  0  1  1  2
 1  2  0  2  2  0  1  1  1  1  1  1  1  ...  2  1  0  2  1  2  0  0  0  2  2
 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
 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
 0  2  0  2  2  0  0  0  2  0  0  1  1  ...  2  2  1  1  1  2  1  1  1  1  1  1
  ⋮          ⋮          ⋮          ⋮          ⋮          ⋮
 0  2  1  2  1  1  1  1  1  1  1  1  1  ...  2  2  1  1  2  1  1  0  0  2  2  2
 0  1  1  1  2  0  1  1  1  1  1  1  1  ...  1  1  2  1  1  2  0  1  1  0  2  1
 0  2  1  2  1  0  1  1  1  1  1  1  1  ...  2  2  2  1  2  2  0  1  1  1  2  0
 1  2  0  2  2  0  1  1  1  1  1  1  1  ...  2  1  2  1  1  1  1  1  1  1  2  1
 0  2  0  2  2  1  1  1  1  1  1  1  2  ...  1  2  2  1  1  2  0  1  1  1  1  1
 1  2  0  2  2  1  1  1  1  1  1  1  1  ...  1  0  2  1  1  1  1  1  1  1  2  1
 0  2  0  2  2  0  0  0  2  0  0  2  2  ...  2  2  1  1  2  2  0  1  1  0  2  1
 1  2  0  2  2  1  2  2  0  2  2  0  1  ...  2  2  2  0  2  2  0  2  2  0  2  0
 0  2  1  2  1  0  0  0  2  0  0  1  0  ...  2  1  1  1  2  1  0  1  1  0  2  1
 0  2  1  2  1  1  1  1  1  1  1  1  1  ...  2  1  1  1  2  2  0  1  1  1  2  0
 0  2  2  2  0  0  1  1  1  1  1  1  1  ...  1  1  2  1  1  1  1  1  1  1  2  1
 1  2  0  2  2  1  1  1  1  1  1  1  2  ...  1  1  2  1  2  2  1  1  1  0  1  0
```

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

```
Out[86]: 9000x50 Array{Int64,2}:
```

```
 0  1  1  1  1  0  0  0  2  2  1  0  0  ...  1  1  2  2  0  2  2  2  2  2  0  0
 0  1  1  1  0  0  0  0  0  2  2  2  2  ...  1  0  2  1  1  1  2  1  1  2  1
 0  1  1  1  2  0  0  0  0  1  1  2  1  ...  2  0  2  2  0  2  2  1  1  2  1
 0  0  0  2  2  0  0  0  1  1  1  1  2  ...  1  1  2  2  0  2  2  1  1  2  2
 0  0  0  1  0  1  0  0  0  0  1  2  0  ...  1  1  2  1  1  1  1  0  0  1  2
 0  1  1  0  1  0  0  0  1  1  1  1  0  ...  1  1  1  2  0  2  2  2  1  2
 1  1  1  1  2  1  0  0  0  0  1  2  2  ...  2  0  1  2  1  1  1  2  2  2  1
 1  1  1  1  2  1  0  1  2  1  1  1  1  ...  2  0  1  2  0  2  2  0  0  2  2
 0  0  0  1  0  0  0  0  1  1  2  2  2  ...  0  0  2  2  1  1  2  2  0  2
 0  1  1  1  2  1  0  0  1  1  0  0  0  ...  2  0  0  2  0  2  2  0  1  0
 0  0  0  2  1  1  0  0  1  2  0  2  1  ...  1  1  1  2  1  1  1  2  2  0
 0  0  0  2  2  0  0  0  1  2  2  2  2  ...  1  2  0  2  2  0  0  2  2
 0  0  0  1  2  1  0  1  2  1  1  2  1  ...  1  0  1  2  0  2  1  2  2
  ⋮           ⋮           ⋮           ⋮           ⋮           ⋮
 1  1  1  1  1  2  0  1  2  1  2  1  2  ...  2  0  0  1  1  1  2  1  2  1
 0  1  1  1  2  2  0  2  2  0  2  2  1  ...  2  0  0  2  0  2  2  2  1  2
 0  1  1  1  1  1  0  1  2  1  1  2  0  ...  1  1  1  2  0  2  2  2  2
 0  1  1  1  1  0  0  0  1  1  1  1  0  ...  1  1  2  0  0  2  2  2  1
 1  1  1  1  2  0  0  0  1  2  0  2  1  ...  1  1  1  2  0  2  2  2
 1  1  1  1  2  1  0  1  1  1  2  2  2  ...  1  1  1  1  1  2  2  1
 0  0  0  2  2  0  0  0  2  2  2  2  2  ...  1  0  1  2  2  0  2  1
 1  2  2  0  2  1  0  1  1  0  2  2  1  ...  2  0  2  2  1  1  2  2
 0  0  0  1  1  0  0  0  2  2  0  2  0  ...  1  2  1  2  1  1  1  2
 1  1  1  1  1  0  0  0  1  1  2  1  1  ...  1  1  0  2  0  2  2  1
 0  1  1  1  2  0  0  0  1  2  0  0  0  ...  1  2  0  2  0  2  2  1
 1  1  1  1  1  0  0  0  1  2  2  2  2  ...  1  2  1  2  1  1  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]: 0.1427722749499512
```

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

WARNING: imported binding for cor overwritten in module Main

```
Out[100]: 0.3761729157753614
```

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

```
Out[101]: 48000x50 Array{Int64,2}:
```

```
0 1 1 0 0 0 0 0 1 1 2 2 2 ... 0 2 2 0 0 2 2 1 1 0 2 2
0 0 0 2 1 0 0 0 1 1 0 0 0 ... 0 0 1 2 1 1 2 2 2 2 1 1
0 0 0 1 1 2 0 0 0 0 1 2 2 ... 1 0 1 1 1 1 1 0 0 1 2 2
0 0 0 2 1 0 0 0 1 1 2 1 1 ... 0 2 2 0 2 0 0 1 1 2 1 1
1 0 0 2 1 0 0 0 2 2 1 0 0 ... 1 0 2 1 0 2 2 1 1 2 2 1
0 1 1 0 1 0 0 0 0 2 1 1 1 ... 1 0 1 2 1 1 2 1 1 2 2 2
0 0 0 1 1 1 0 1 2 1 1 0 0 ... 1 1 0 2 0 2 2 0 1 2 1 1
0 1 1 0 1 0 0 0 0 1 0 0 0 ... 0 1 1 1 2 0 1 2 2 2 0 1
0 1 1 1 2 0 0 0 0 2 1 0 0 ... 1 0 2 1 0 2 2 2 1 1 1 1
0 1 1 1 1 1 0 1 2 1 0 1 0 ... 1 1 0 2 0 2 2 1 0 1 2 2
0 0 0 1 0 0 0 0 1 1 1 1 1 ... 2 0 1 2 1 1 1 1 1 1 1 1
0 0 0 0 1 1 0 0 1 2 1 1 1 ... 1 0 0 2 0 2 2 1 1 1 2 2
0 0 0 0 2 0 0 0 0 1 1 2 0 ... 2 0 0 2 0 2 1 2 2 2 0 1
⋮           ⋮           ⋮           ⋮           ⋮           ⋮
1 1 1 1 1 2 0 1 2 1 2 1 2 ... 2 0 0 1 1 1 2 1 2 1 1 2
0 1 1 1 2 2 0 2 2 0 2 2 1 ... 2 0 0 2 0 2 2 2 1 2 1 1
0 1 1 1 1 1 0 1 2 1 1 2 0 ... 1 1 1 2 0 2 2 2 2 2 1 0
0 1 1 1 1 0 0 0 1 1 1 1 0 ... 1 1 2 0 0 2 2 2 1 2 1 1
1 1 1 1 2 0 0 0 1 2 0 2 1 ... 1 1 1 2 0 2 2 2 2 2 1 1
1 1 1 1 2 1 0 1 1 1 2 2 2 ... 1 1 1 1 1 2 2 1 0 2 1 1
0 0 0 2 2 0 0 0 2 2 2 2 2 ... 1 0 1 2 2 0 2 1 2 1 1 1
1 2 2 0 2 1 0 1 1 0 2 2 1 ... 2 0 2 2 1 1 2 2 2 2 0 0
0 0 0 1 1 0 0 0 2 2 0 2 0 ... 1 2 1 2 1 1 1 2 1 1 1 1
1 1 1 1 1 0 0 0 1 1 2 1 1 ... 1 1 0 2 0 2 2 1 1 1 1 0
0 1 1 1 2 0 0 0 1 2 0 0 0 ... 1 2 0 2 0 2 2 1 1 2 1 1
1 1 1 1 1 0 0 0 1 2 2 2 2 ... 1 2 1 2 1 1 2 2 1 2 1 0
```

```
In [102]: QTL=gtlEffects[QTLPos]
```

```
Out[102]: 50-element Array{Float64,1}:
```

```
0.200617  
0.20287  
0.200402  
0.198905  
0.200495  
0.201021  
0.200408  
0.202063  
0.19932  
0.198041  
0.197745  
0.197636  
0.199035  
:  
0.203332  
0.202906  
0.200718  
0.1983  
0.201681  
0.199296  
0.19636  
0.196844  
0.202043  
0.204598  
0.19784  
0.204511
```

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

```
Out[103]: 48000-element Array{Float64,1}:  
 10.587  
  9.39833  
  9.39842  
  9.02114  
 10.7969  
  8.40533  
  9.80313  
  8.61299  
  8.99245  
  9.6022  
  8.19973  
  9.79346  
  8.99946  
  ⋮  
 12.203  
 11.6013  
 12.3985  
 11.1923  
 13.1856  
 10.4071  
 11.3829  
 12.2058  
 11.7943  
 10.7912  
 11.1994  
 12.1792
```

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

```
Out[104]: 9.408605549742964
```

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

```
Out[105]: 10.013501374666918
```

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

```
Out[106]: 10.273654242462808
```



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

```
Out[107]: 10.538407747338226
```

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

```
Out[108]: 10.784577462232138
```

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

```
Out[109]: 11.06219869029687
```

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

```
Out[110]: 9000-element Array{Float64,1}:
```

```
10.205
10.5836
11.0094
10.3977
10.805
10.3992
11.0022
11.8051
10.7788
9.20719
10.9908
10.7948
10.9864
⋮
12.203
11.6013
12.3985
11.1923
13.1856
10.4071
11.3829
12.2058
11.7943
10.7912
11.1994
12.1792
```

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

```
Out[111]: 11.039086332825667
```

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

```
Out[112]: 1.630480783082703
```

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

```
Out[113]: 10.606360944540167
```

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

```
Out[114]: 1.1977553947972037
```

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

```
Out[115]: 10.56045503922889
```

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

```
Out[116]: 1.151849489485926
```

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

```
Out[117]: 10.786804516874382
```

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

```
Out[118]: 1.3781989671314179
```

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

```
Out[119]: 11.003394933153297
```

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

```
Out[120]: 1.5947893834103333
```

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

```
Out[121]: 11.313921931483394
```

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

```
Out[122]: 1.9053163817404304
```

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

```
Out[123]: 11.06219869029687
```

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

```
Out[124]: 1.653593140553907
```