

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

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

```
Out[2]: XSim
```

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

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

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

## Initialize XSim

```
In [6]: numChr      = 10
chrLength = 0.1
numLoci    = 200
nQTL       = 50
mutRate    = 0.0
locusInt   = chrLength/numLoci
mapPos     = collect(locusInt/2:locusInt:chrLength)
geneFreq   = fill(0.5,numLoci)
QTL = sample(1:numLoci,nQTL,replace=false)
qtlMarker  = fill(false,numLoci)
qtlMarker[QTL] = true
mu = 100                                     #  $\alpha \sim N(100,1)$ 
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  $\alpha \mu = 0.2$ 
XSim.init(numChr,numLoci,chrLength,geneFreq,mapPos,qtlMarker,qtlEffects,mutRate)
```

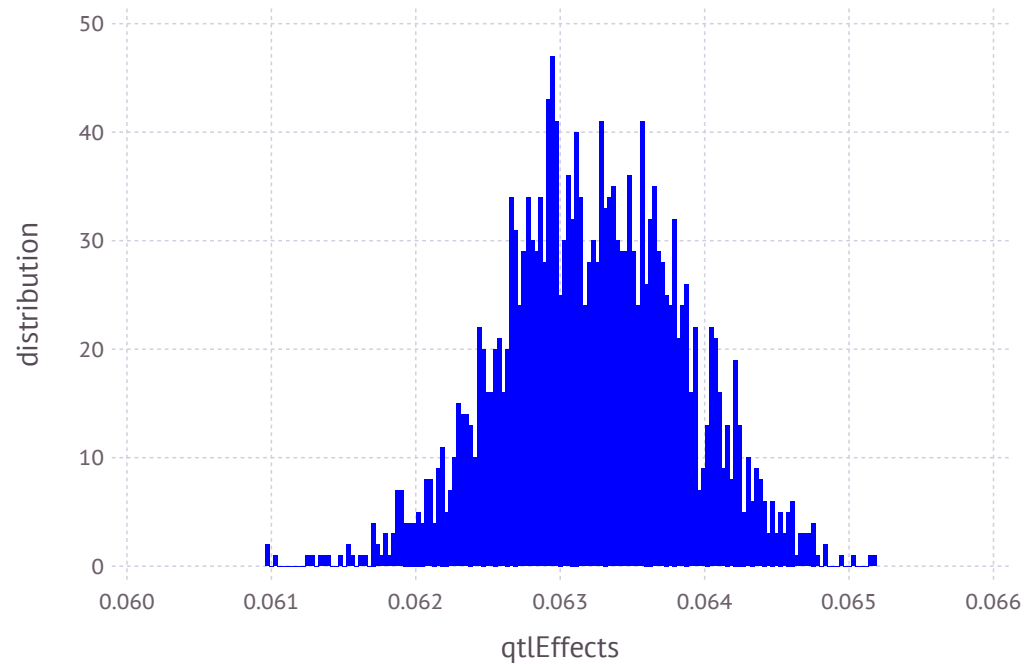
```
In [7]: qtlEffects
```

```
Out[7]: 2000-element Array{Float64,1}:  
  0.0626443  
  0.0628468  
  0.0614654  
  0.0634296  
  0.0639981  
  0.0632229  
  0.0624375  
  0.0624748  
  0.0636636  
  0.0639303  
  0.0637998  
  0.0637967  
  0.0629287  
  ⋮  
  0.0630898  
  0.0630963  
  0.0624527  
  0.0630838  
  0.063296  
  0.0619012  
  0.0638837  
  0.0635829  
  0.0623598  
  0.0623546  
  0.0630483  
  0.0630623
```

```
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]: 0.0632232694139118
```

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

```
Out[11]: 3.9603653399844524e-7
```

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

# Sample 20 sire and 400 dams
popSizeSP = 8000

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

```

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

# FileName :
PedAll = "PedAll.txt"           # pedigree file with all animals
PheAll = "PheAll.txt"          # phenotype file with all 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/ch1to10.2000SNP.group1")
         dams  = XSim.sampleFounders("/home/nicole/Jupyter/HERdata/data/Clean/newdata/ch1to10.2000SNP.group2");
```

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

## Base population

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

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

## Sample animals for sire and dam candidates

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

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

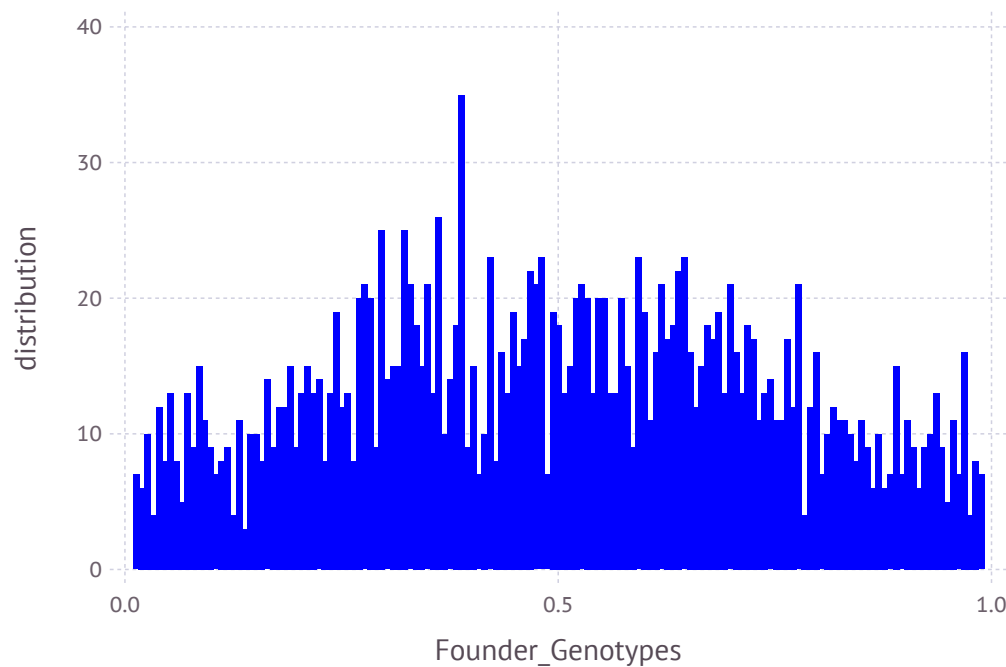
```
In [19]: gSPSire = XSim.getOurGenotypes(popSP[1])
gSPDam = XSim.getOurGenotypes(popSP[2])
gSP = [gSPSire;gSPSire];
```

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

```
Out[20]: 1x2000 Array{Float64,2}:
 0.05575  0.84125  0.27725  0.9535 ... 0.28375  0.352375  0.437  0.262
```

```
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);
```



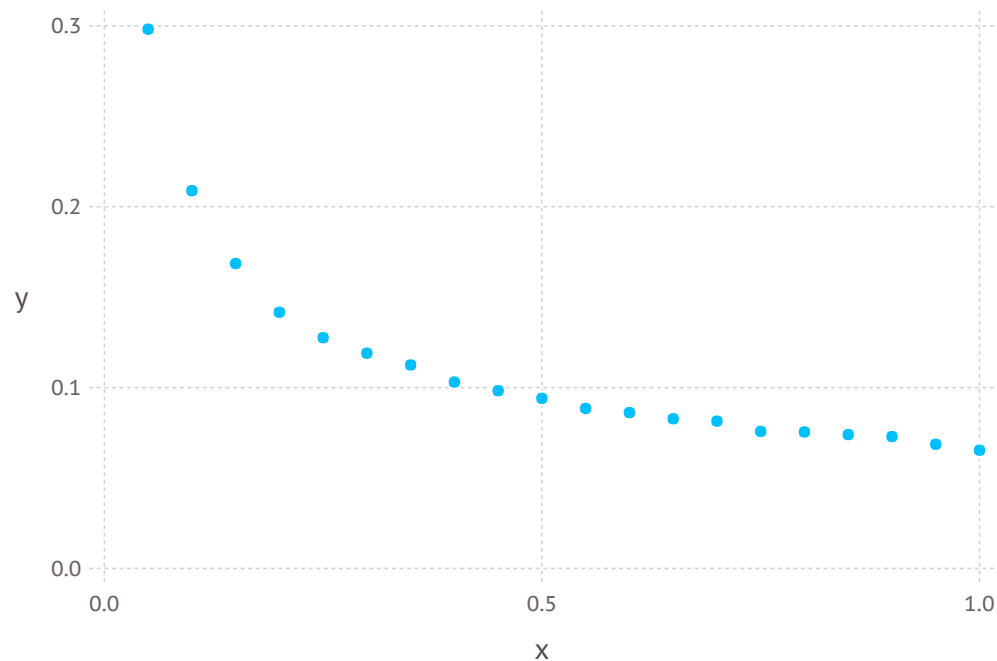
```
In [23]: for i=1:(nRows-20)
LDMat[i,:] = corMat[i,(i+1):(i+20)].^2
end
```

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

```
Out[24]: 1x20 Array{Float64,2}:
 0.0653668  0.06865  0.0729459  0.0740086  ...  0.168611  0.208907  0.298217
```

```
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]: 31.545704937035524
```

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

```
Out[31]: 0.8652552483790809
```

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

```
Out[32]: 0.8652552483790809
```

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

```
Out[33]: 0.8652552483790809
```

```
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]: 34.67292602263478
```

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

```
Out[36]: 34.68631577822002
```

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

```
Out[37]: 0.7067980200537511
```

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

```
Out[38]: 0.7179004327149401
```

## Pedigree: All animals

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

```
Out[39]: 48000x3 Array{Int64,2}:  
  40722  36707  39928  
  40723  35207  39660  
  40724  35670  37260  
  40725  34389  37587  
  40726  36151  39352  
  40727  34155  39759  
  40728  36012  39958  
  40729  34178  39612  
  40730  32871  38361  
  40731  34988  40260  
  40732  35395  39853  
  40733  32912  37485  
  40734  35874  39190  
      ⋮  
  88710  74678  80664  
  88711  74479  79674  
  88712  75248  77681  
  88713  75945  80299  
  88714  75686  80350  
  88715  76022  80567  
  88716  75122  77297  
  88717  74924  80143  
  88718  73322  78424  
  88719  75685  77075  
  88720  75122  77719  
  88721  75498  80390
```

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

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

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

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

# Create matrix of "genotype" covariates for all animals

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

```
Out[43]: 48000
```

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

```
Out[44]: 2000
```

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

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

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

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

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

## Create marker file for all animals

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

```
Out[48]: 48000x2000 Array{Int64,2}:
```

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

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

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

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

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

## Genotypes - all sires and all offsprings in G5

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

```
Out[52]: 40000-element Array{Int64,1}:
```

```
42923
41703
43989
44112
44026
44407
40819
43085
41502
41497
43554
41502
43449
⋮
74678
74479
75248
75945
75686
76022
75122
74924
73322
75685
75122
75498
```



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

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

```
42923
```

```
41703
```

```
43989
```

```
44112
```

```
44026
```

```
44407
```

```
40819
```

```
43085
```

```
41502
```

```
41497
```

```
43554
```

```
43449
```

```
40857
```

```
:
```

```
74973
```

```
75472
```

```
75821
```

```
75218
```

```
75289
```

```
76236
```

```
75122
```

```
75945
```

```
76657
```

```
75996
```

```
72887
```

```
73952
```

```
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}:  
 42923  
 41703  
 43989  
 44112  
 44026  
 44407  
 40819  
 43085  
 41502  
 41497  
 43554  
 43449  
 40857  
      ⋮  
 88710  
 88711  
 88712  
 88713  
 88714  
 88715  
 88716  
 88717  
 88718  
 88719  
 88720  
 88721
```

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

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

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

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

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

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

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

```
Out[61]: (9000,2001)
```

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

```
Out[62]: 9000
```

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

```
Out[63]: 2001
```

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

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

```
In [65]: for i in 1:size(GSOFF5,1)
          for j in 1
              @printf(GSOFF5stream, "%19d", GSOF5[i,j])
          end
          for k in 2:size(GSOFF5,2)
              @printf(GSOFF5stream, "%3d", GSOF5[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  33.012  31.82  
  40723  30.421  31.116  
  40724  31.696  30.109  
  40725  30.591  30.87  
  40726  30.978  31.127  
  40727  30.874  30.867  
  40728  31.85   32.44  
  40729  32.395  31.936  
  40730  33.32   32.01  
  40731  31.9    32.138  
  40732  30.235  30.868  
  40733  27.963  29.6  
  40734  30.315  31.43  
      ⋮  
  88710  35.88   36.45  
  88711  35.907  35.861  
  88712  32.962  33.971  
  88713  36.277  34.531  
  88714  35.418  35.347  
  88715  36.582  36.375  
  88716  32.953  35.601  
  88717  38.396  36.504  
  88718  35.352  35.235  
  88719  35.556  35.419  
  88720  36.164  35.417  
  88721  35.002  35.23
```

```
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]: 500-element Array{Int64,1}:
```

```
 1
 5
 7
 8
23
24
36
38
43
46
48
51
52
 ⋮
1969
1971
1972
1976
1978
1980
1984
1987
1989
1990
1999
2000
```



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

```
Out[83]: 1500-element Array{Int64,1}:
```

```
 2
 3
 4
 6
 9
10
11
12
13
14
15
16
17
 :
1983
1985
1986
1988
1991
1992
1993
1994
1995
1996
1997
1998
```

## Genotype codes: 0, 1, 2

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

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

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

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

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

```
Out[86]: 9000x500 Array{Int64,2}:
```

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

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

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

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

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

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

## Remove Fixed Gene from the panel

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

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

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

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

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

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

## Check heritability

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

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

```
Out[99]: 0.6798688691717885
```

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

WARNING: imported binding for cor overwritten in module Main

```
Out[100]: 0.8238124898423019
```

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

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

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

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

```
Out[102]: 500-element Array{Float64,1}:
```

```
0.0626443  
0.0639981  
0.0624375  
0.0624748  
0.0628699  
0.0627749  
0.063248  
0.063345  
0.0633857  
0.0635127  
0.0637932  
0.0628666  
0.0628237  
⋮  
0.0640807  
0.0636503  
0.062467  
0.0634644  
0.0629775  
0.0637998  
0.0640579  
0.063087  
0.0630898  
0.0630963  
0.0630483  
0.0630623
```

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

```
Out[103]: 48000-element Array{Float64,1}:  
 31.8149  
 31.0831  
 30.0838  
 30.8523  
 31.1133  
 30.8488  
 32.4177  
 31.9332  
 31.9878  
 32.1228  
 30.8408  
 29.5873  
 31.4121  
  ⋮  
 36.4224  
 35.8395  
 33.9563  
 34.5033  
 35.3498  
 36.3547  
 35.5928  
 36.4712  
 35.2174  
 35.4105  
 35.4001  
 35.2116
```

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

```
Out[104]: 31.530776930666516
```

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

```
Out[105]: 32.20852800492752
```

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

```
Out[106]: 32.8361152672931
```



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

```
Out[107]: 33.434858742580055
```

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

```
Out[108]: 34.058606626453
```

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

```
Out[109]: 34.66366864709146
```

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

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

```
32.9296
33.9425
32.8752
32.0599
32.9424
31.7349
31.8696
33.3697
33.6887
32.2508
33.68
32.9466
32.6062
⋮
36.4224
35.8395
33.9563
34.5033
35.3498
36.3547
35.5928
36.4712
35.2174
35.4105
35.4001
35.2116
```

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

```
Out[111]: 34.597294755675115
```

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

```
Out[112]: 3.0665178250085994
```

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

```
Out[113]: 32.881272640716986
```

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

```
Out[114]: 1.3504957100504704
```

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

```
Out[115]: 33.46496890831444
```

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

```
Out[116]: 1.9341919776479237
```

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

```
Out[117]: 34.04668752382552
```

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

```
Out[118]: 2.5159105931590027
```

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

```
Out[119]: 34.672635218718774
```

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

```
Out[120]: 3.1418582880522585
```

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

```
Out[121]: 35.26595383014613
```

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

```
Out[122]: 3.7351768994796153
```

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

```
Out[123]: 34.66366864709146
```

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

```
Out[124]: 3.132891716424947
```