Maize Selection Tests

Investigating genome size variation in maize and teosinte. Ultimately, we want to identify whether our nuclear phenotypes (genome size, repeat abundance) are changing more rapidly than we would expect given kinship matrices. We have data from approximately 90 Zea landraces with point estimates for phenotype+genotype, and <10 populations each of parv and mex, with population averages for each. First, I plot the data to visualize the variation across altitudinal clines. We want to get a sense of the altitudinal trends.

Read in the maize data with this chunk.

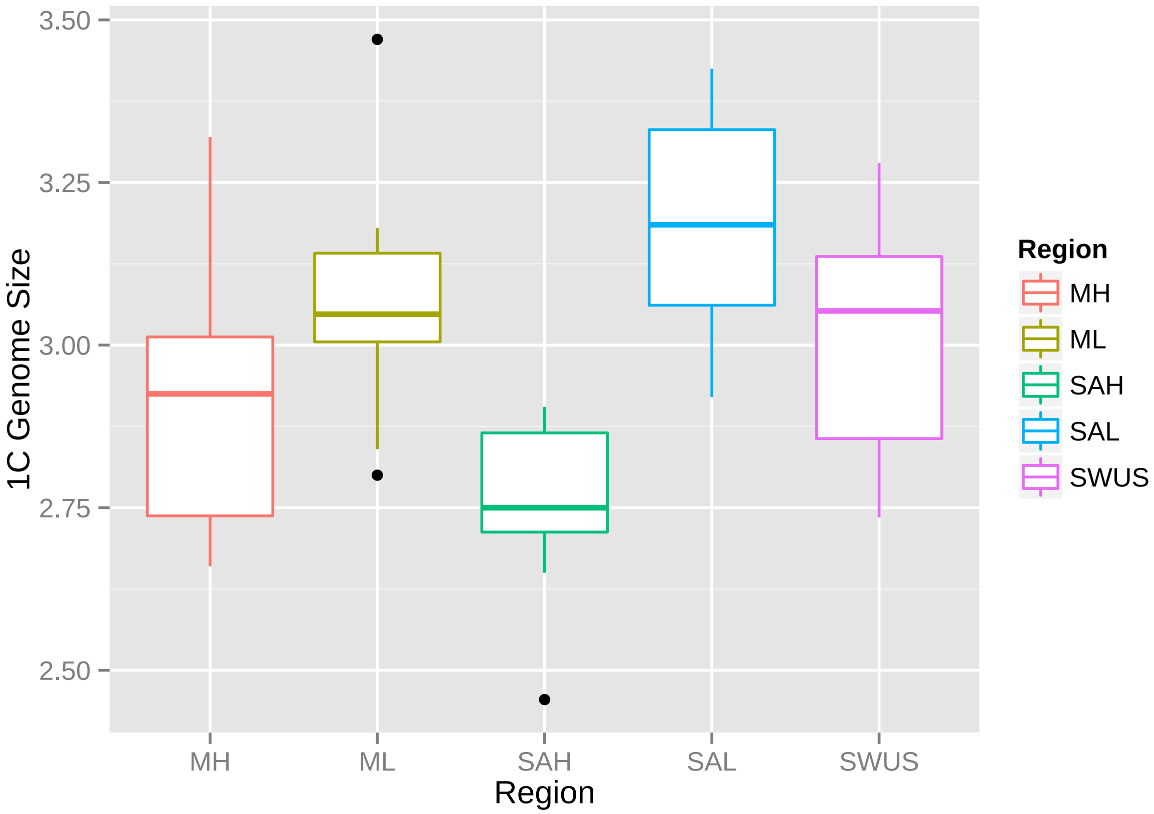
```r
setwd("~/Documents/Projects/Genome_Size_Analysis")
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.0.2
```

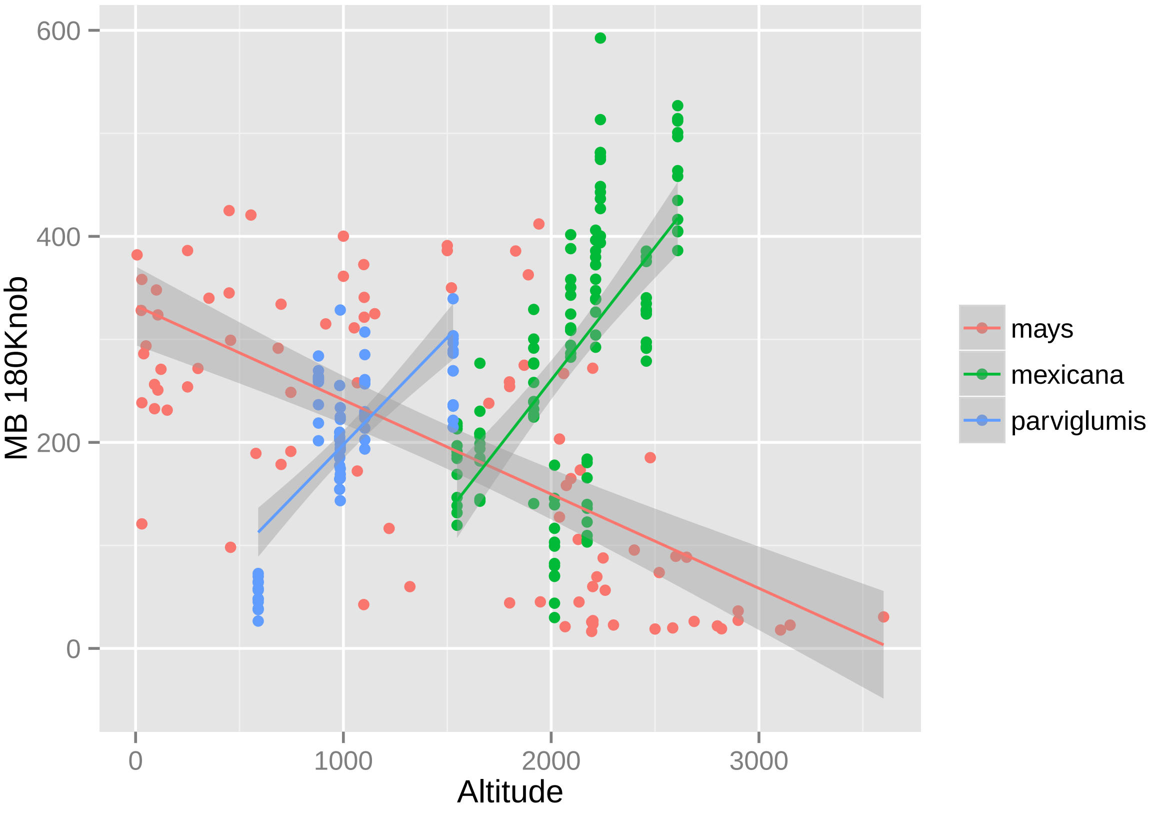
```r
dataall <- read.csv("Master_Data_noNA.csv")
dataal <- subset(dataall, dataall$X1C_GS!="NA")
data <- subset(dataal, dataal$X1C_GS<3.6)
dmays <- subset(data, data$Species=="mays")
```

Plot the variation. Here, we note that several repeats have opposing clines. Furthermore, we also find that the teosintes have a different trend in terms of within species genome size cline. Though they do show that the highland mex is of average smaller size than the lowland parv, both mex and parv increase in genome size across altitude.

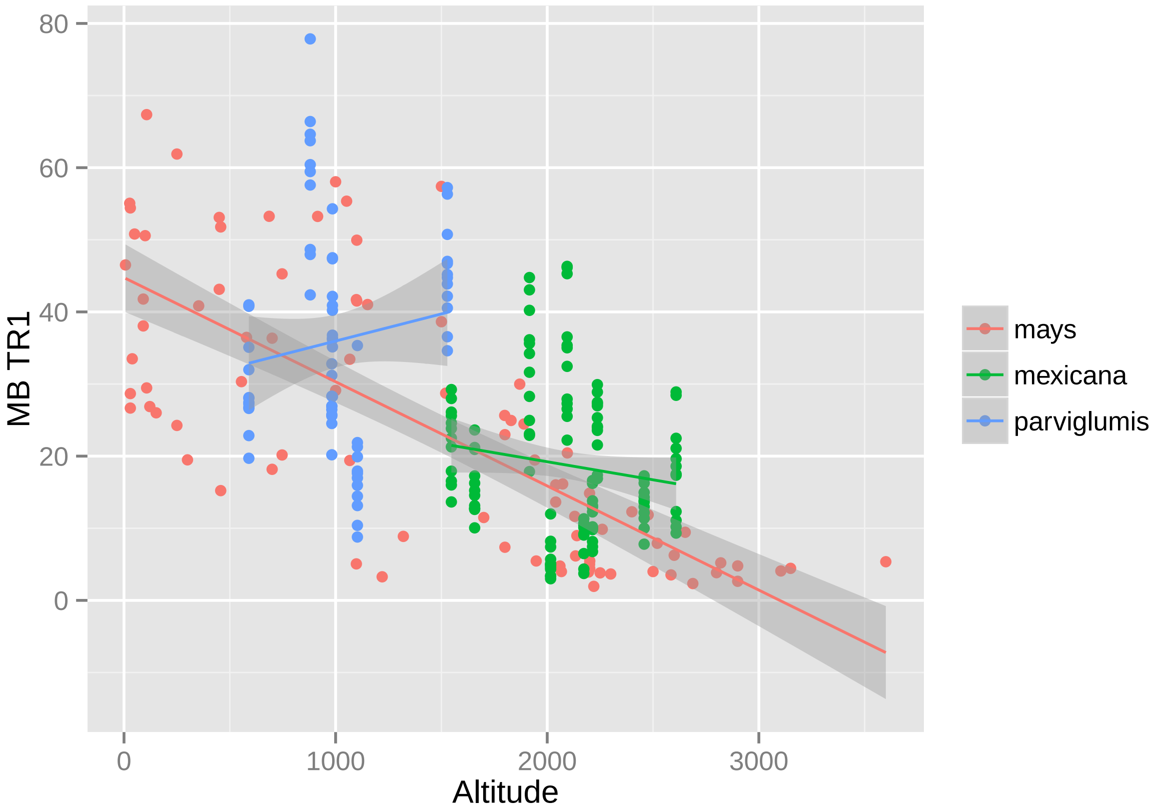
```r
p1 <- ggplot(dmays, aes(Region, X1C_GS, color=Region)) + geom_boxplot()+ ylab("1C Genome Size")
p1
```



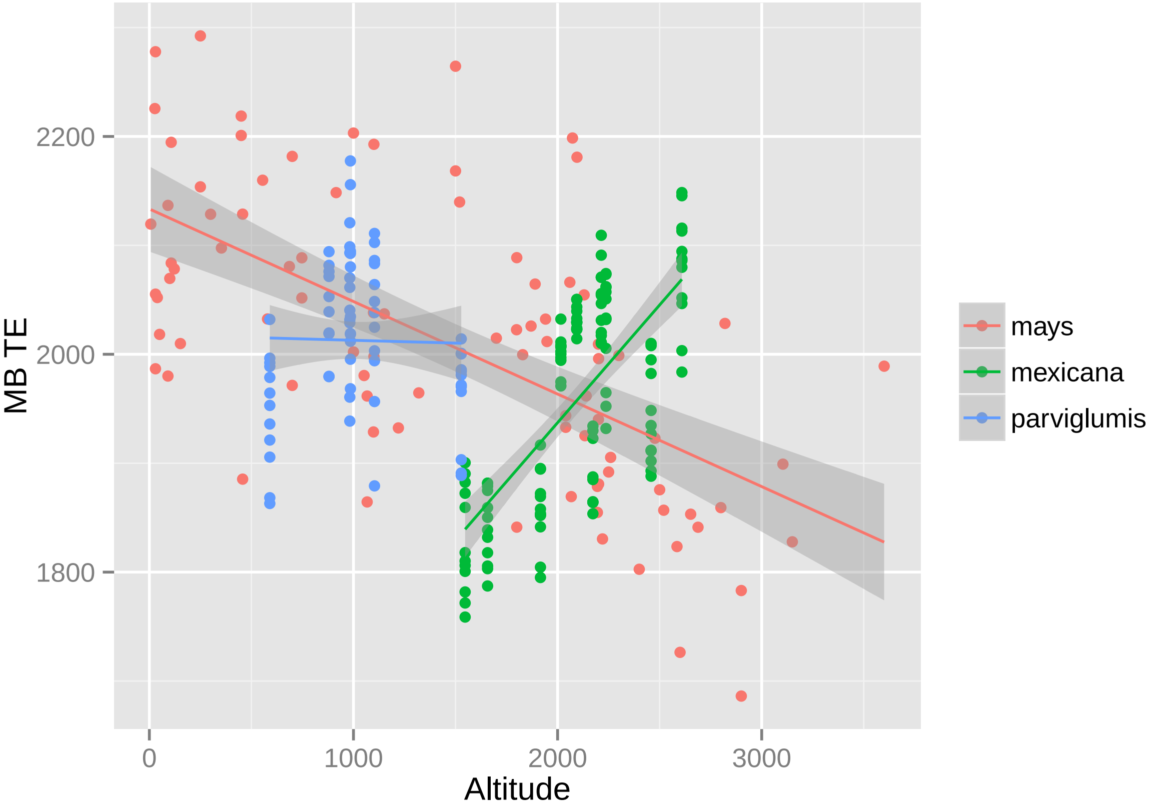
```
p2 <- ggplot(data, aes(Altitude, X180knobMB, color=Species)) + geom_point()+ ylab("MB 180Knob") + theme
p2
```



```
p <- ggplot(data, aes(Altitude, TR1MB, color=Species)) + geom_point()+ ylab("MB TR1") + theme(legend.ti
p
```



```
p <- ggplot(data, aes(Altitude, TotallTeMB, color=Species)) + geom_point()+ ylab("MB TE") + theme(legenc
p
```



Feel free to alter pieces of the plotting code to see trends in different repeats. Given the observed trends, we can now advance to studying selection on genomic elements. Read in the genotype data with this code, and generate the genetic matrix with the following code.

Of importance is that we have point estimates for landraces and population averages for the teosintes. RRBLUP will take in allele frequencies and generate kinship matrices, so we feed it the point estimates in maize, and then the population averages in teosinte. NOTE: GENERATE POP AVERAGES FOR TEOSINTE OVER XMAS.

go back to teosinte data, convert to frequencies of allele * 2 so that we get the # of alleles in each individual

```
setwd("~/Documents/Projects/Genome_Size_Analysis/Github_ParallelGS/SelectionTests/")
library("rrBLUP")
```

```
## Warning: package 'rrBLUP' was built under R version 3.0.2
```

```
geno <- read.csv("~/Documents/Projects/Genome_Size_Analysis/Github_ParallelGS/SNP_data/Landrace_noSWUS_
dt <-t(geno)
A <- A.mat(dt)
```

Read in the phenotype data, and make sure the order of the samples are the same as order in the genetic matrix. To do so, we use the tmp variables that store the proper order sample IDs in the maize samples. Note, this is just for the maize samples, the teosintes are done later.

```
pheno <-read.csv("~/Documents/Projects/Genome_Size_Analysis/Github_ParallelGS/PhenotypeData/Landraces_no
#to make sure order is the same
tmp1 <- as.data.frame(colnames(geno))
names(tmp1)[1] <- "names"
tmp2 <- as.data.frame(pheno$FullID)
tmp <- setdiff(tmp1,tmp2)

phenoorder <- merge(tmp1,pheno, by.x="names", by.y="FullID",sort=FALSE)
```

This next chunk was our first iteration of Jeremy's code, we can ignore it for now and step to the current version. Kept here for archaelogy.

```
#library ( mvtnorm )
#EnvVarTest <- function ( phenos , kinship.mat , test.vector, verbose=F ) {

  # 'phenos' is a vector containing the phenotype
  # (i.e. number of repeats) for each individual; dimensions are N x 1
  # 'kinship.mat' is the kinship matrix; dimensions are N x N;
  # rows and columns need to be in the same order as the phenotypes in the vector
  # test.vector is the environmental factor of interest (in this case altitude)

#   eigs <- eigen ( kinship.mat )
#   # get eigendecomposition of kinship matrix
#   rt.inv <- eigs$vec %*% diag ( 1/sqrt(eigs$val) ) %*% t ( eigs$vec ) # standardizing distances across
#   # calculate inverse of the square root matrix
#   rotated.phenos <- t ( rt.inv ) %*% phenos
#   # rotate phenotypes from population space into principal component space
#   test.vector <- test.vector / (sqrt ( 2 * sum ( test.vector^2 ) ) )
#   # scale to be unit length after rotation
#   #recover()
#   rotated.vector <- rt.inv %*% test.vector
#   # rotate environmental variable from population space into principal component space
#   model <- lm ( rotated.phenos ~ 1+rotated.vector)
#   # fit regression model
#   r.sq <- cor.test ( rotated.phenos , rotated.vector )$estimate^2
#   # get r^2
#   ANOVA <- anova ( model )
#   # get p value
#   if(verbose){ print(ANOVA) }
#   return ( c ( model$coef[2] , r.sq , ANOVA[5][[1]][1] )) # return
# #}
```

EnvVarTest, the code written by Jeremy Berg to run an Fst-Qst like test on our data. Comments are included within the code to explain what it is doing at each step.

```
library ( mvtnorm )
```

```
## Warning: package 'mvtnorm' was built under R version 3.0.2
```

```
EnvVarTest <- function ( phenos , kinship.mat , test.vector, verbose=F ) {

    # 'phenos' is a vector containing the phenotype (i.e. number of repeats) for each individual; dimens
    # 'kinship.mat' is the kinship matrix; dimensions are N x N; rows and columns need to be in the sam
    # test.vector is the environmental factor of interest (in this case altitude)

    #recover()
    eigs <- eigen ( kinship.mat ) ## get eigendecomposition of kinship matrix
    rt.inv <- eigs$vec %*% diag ( 1/sqrt(eigs$val) ) %*% t ( eigs$vec )# calculate inverse of the squar

  cent.test.vector <- test.vector - mean ( test.vector )
  cent.phenos <- phenos - mean ( phenos )

  rotated.phenos <- rt.inv %*% cent.phenos # rotate phenotypes from population space into principal com
    unit.test.vector <- cent.test.vector / (sqrt ( 2 * sum ( cent.test.vector^2 ) ) ) # scale to be uni
    #recover()




  rotated.vector <- rt.inv %*% cent.test.vector # rotate environmental variable from population space i
    model <- lm ( rotated.phenos ~ rotated.vector) # fit regression model
    r.sq <- cor.test ( rotated.phenos , rotated.vector )$estimate^2 # get r^2
    ANOVA <- anova ( model ) # get p value

    # expected fraction of variance associated with environmental variable under neutrality
    F.env <- ( t ( unit.test.vector ) %*% kinship.mat %*% unit.test.vector ) / sum ( diag ( kinship.mat
    sums.sq <- cumsum ( ANOVA[2] )

    # fraction of variance associated with environmental variable for trait
    Q.env <- sums.sq[1,]/ ( sums.sq[2,] )
if(verbose==T){print(ANOVA)}
    return ( c ( model$coef[2] , r.sq , ANOVA[5][[1]][1] , ANOVA[3][2,] , F.env , Q.env) ) # return reg

}
```

Regression coefficient is the B

Now run the equation on each of the nuclear phenotypes. Altitude is our environmental variable, and we will expand on doing this with other bioclim variables. The MB signifies megabases of the repeat, and the . is a replacement for the % symbol.

```
EnvVarTest(phenoorder$X180knobMB,A,phenoorder$Altitude)
```

```
## rotated.vector            cor
##     -4.950e-02      1.176e-01      1.503e-03      5.432e+03      2.780e-02
##
##      1.176e-01
```

```
EnvVarTest(phenoorder$X180knob.,A,phenoorder$Altitude)
```

```
## rotated.vector            cor
##      -0.001420       0.106999       0.002541       4.972795       0.027795
##
##       0.106999
```

```
EnvVarTest(phenoorder$TotallTeMB,A,phenoorder$Altitude)
```

```
## rotated.vector            cor
##     -6.148e-02      1.135e-01      1.845e-03      8.728e+03      2.780e-02
##
##      1.135e-01
```

```
EnvVarTest(phenoorder$TotallTe.,A,phenoorder$Altitude)
```

```
## rotated.vector            cor
##        0.00043        0.02360        0.16555        2.26142        0.02780
##
##        0.02360
```

```
EnvVarTest(phenoorder$TR1MB,A,phenoorder$Altitude)
```

```
## rotated.vector            cor
##     -8.281e-03      1.408e-01      4.735e-04      1.237e+02      2.780e-02
##
##      1.408e-01
```

```
EnvVarTest(phenoorder$TR1.,A,phenoorder$Altitude)
```

```
## rotated.vector            cor
##     -0.0002423      0.1349698      0.0006342      0.1112094      0.0277952
##
##      0.1349698
```

```
EnvVarTest(phenoorder$CentCMB,A,phenoorder$Altitude)
```

```
## rotated.vector            cor
##      8.894e-05      4.507e-04      8.489e-01      5.187e+00      2.780e-02
##
##      4.507e-04
```

```
EnvVarTest(phenoorder$CentC.,A,phenoorder$Altitude)
```

```
## rotated.vector            cor
##      1.547e-05      1.229e-02      3.184e-01      5.682e-03      2.780e-02
##
##      1.229e-02
```

```
EnvVarTest(phenoorder$CRMallMB,A,phenoorder$Altitude)
```

```
## rotated.vector            cor
##     -0.0004517      0.0909714      0.0055882      0.6025926      0.0277952
##
##      0.0909714
```

```
EnvVarTest(phenoorder$CRMall.,A,phenoorder$Altitude)
```

```
## rotated.vector             cor
##     -2.730e-06      3.618e-03      5.891e-01      6.068e-04      2.780e-02
##
##      3.618e-03
```

```
EnvVarTest(phenoorder$X1C_GS,A,phenoorder$Altitude)
```

```
## rotated.vector             cor
##     -0.0001129      0.1495820      0.0003042      0.0214347      0.0277952
##
##      0.1495820
```

```
#plot(phenoorder$TotallTe. ~ phenoorder$Altitude)
#lm(phenoorder$TotallTe. ~ phenoorder$Altitude)
```

Jeremy’s simulation code, need to acknowledge NIH grant, reminder for paper.

```
# load
#
# my.betas <- c ( seq ( 0 , 1 , by = 0.1 ), seq ( 1.2 , 2, 0.2 ) )
# env <-rnorm ( n = 83 , 0 , sd = 1 ) # different individuals
# tmp <- list ()
# power <- numeric ( length ( my.betas ) )
# for ( i in 1 : length ( my.betas ) ) {
#
#   beta <- my.betas [ i ]
#   mu <- rep ( 0 , 152 )
#   my.mean <- mu + beta * env
#   test.data <- rmvnorm ( n = 1000 , mean = my.mean , sigma = this.cov.mat )
#
#   tmp [[ i ]] <- matrix ( NA , nrow = 1000 , ncol = 6 )
#   for ( j in 1 : 1000 ) {
#       tmp [[ i ]] [ j , ] <- EnvVarTest ( test.data[ j , ] , this.cov.mat , env )
#   }
#   #hist ( tmp [[ i ]] [ , 3 ] , breaks = 50 )
#
#   power [ i ] <- sum ( tmp [[ i ]] [ , 3  ] < 0.05 ) / 1000
#
# }
#
# Q.stat <- do.call ( cbind , lapply ( tmp , function ( x ) x [ , 6 ] ) )
#
# par ( mfrow = c ( 2, 1 ) )
# plot ( my.betas , power , xlab = expression ( beta ) , ylab = "Power"  , type = "l" , lwd = 2 )
#
# boxplot ( t ( Q.stat ) ~ my.betas , pch = 20 , xlab = expression ( beta ), ylab = expression ( Q[ENV]
# abline ( h = tmp [[ 1 ]] [ 1, 5 ] )
# Q.stat <- do.call ( cbind , lapply ( tmp , function ( x ) x [ , 6 ] ) )
#
```

```
# par ( mfrow = c ( 2, 1 ) )
# plot ( my.betas , power , xlab = expression ( beta ) , ylab = "Power"  , type = "l" , lwd = 2 )
#
# boxplot ( t ( Q.stat ) ~ my.betas , pch = 20 , xlab = expression ( beta ), ylab = expression ( Q[ENV]
# abline ( h = tmp [[ 1 ]] [ 1, 5 ] )
```

Chunk for all of the teosintes, though currently not working as I need to make the matrices with population averages of genotypes in RRblup. Initial population covariance matrices from Bayenv were made by Tim Beissenger, and the pop averages from my phenotypes stuff. Bayenv uses different assumptions than rrBLUP, and therefore we want to stick to just rrBLUP.

```
# phenoteo <-read.csv("~/Documents/Projects/Genome_Size_Analysis/Github_ParallelGS/PhenotypeData/Teosin
#
# genoteo <- read.csv("~/Documents/Projects/Genome_Size_Analysis/Github_ParallelGS/SNP_data/TeosinteAll
#
# EnvVarTest(phenoteo$X180knobMB,genoteo,phenoteo$Altitude)
# EnvVarTest(phenoteo$X180knob.,genoteo,phenoteo$Altitude)
# EnvVarTest(phenoteo$TR1MB,genoteo,phenoteo$Altitude)
# EnvVarTest(phenoteo$TR1.,genoteo,phenoteo$Altitude)
# EnvVarTest(phenoteo$CentCMB,genoteo,phenoteo$Altitude)
# EnvVarTest(phenoteo$CentC.,genoteo,phenoteo$Altitude)
```

Chunk for just parv.

```
# phenoparv <-read.csv("~/Documents/Projects/Genome_Size_Analysis/Github_ParallelGS/PhenotypeData/Teosi
#
# genoparv <- read.csv("~/Documents/Projects/Genome_Size_Analysis/Github_ParallelGS/SNP_data/TeosintePa
#
# EnvVarTest(phenoparv$X180knobMB,genoparv,phenoparv$Altitude)
# EnvVarTest(phenoparv$X180knob.,genoparv,phenoparv$Altitude)
# EnvVarTest(phenoparv$TR1MB,genoparv,phenoparv$Altitude)
# EnvVarTest(phenoparv$TR1.,genoparv,phenoparv$Altitude)
# EnvVarTest(phenoparv$CentCMB,genoparv,phenoparv$Altitude)
# EnvVarTest(phenoparv$CentC.,genoparv,phenoparv$Altitude)
```

Chunk for selection on mex.

```
# phenomex <-read.csv("~/Documents/Projects/Genome_Size_Analysis/Github_ParallelGS/PhenotypeData/Teosin
#
# genomex <- read.csv("~/Documents/Projects/Genome_Size_Analysis/Github_ParallelGS/SNP_data/TeosinteMex
#
# EnvVarTest(phenomex$X180knobMB,genomex,phenomex$Altitude)
# EnvVarTest(phenomex$X180knob.,genomex,phenomex$Altitude)
# EnvVarTest(phenomex$TR1MB,genomex,phenomex$Altitude)
# EnvVarTest(phenomex$TR1.,genomex,phenomex$Altitude)
# EnvVarTest(phenomex$CentCMB,genomex,phenomex$Altitude)
# EnvVarTest(phenomex$CentC.,genomex,phenomex$Altitude)
```

Power sims chunk. The rough draft but working version of it, cleaning it up with a funciton in the next go around.

```r
#install.packages("MASS")
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.0.2
```

```r
# p = mvn(my + Selection * Altitude, Va * Covariance)
# p will be an output of our phenotypes
mu <- rep(0, 83) #I think since we want 1K sims, we want to make the length of this 1k
Altitude <- phenoorder$Altitude
B <- 10
pizza <- mu + B * Altitude
#covariance matrix is given in A
VA=1 #additive genetics variance
simpheno <- mvrnorm(n=1000,pizza, VA*A) #n=number of samples we have, for maize 83, for allteo 16 pop
#pieces i am missing: where to stick in altitude?  I think it is at the mu

pvals=sapply(1:1000,function(X) EnvVarTest(simpheno[X,],A,Altitude)[3])
#EnvVarTest(simpheno[X,],A,Altitude,verbose=T)
#hist(pvals,breaks=100)
#plot(density(pvals))
```

Function for sims. For the variables, here is an in text explanation: Samples is how many reps we want to run slope is our beta term, how strong selection is add.gen.var we assume to be 1, but is Va gen.covar.mat is our covariance matrix based on SNP genetypes, built earlier as matrix A altitudes is a vector of altitudes

```r
sim <- function ( samples , slope , add.gen.var , gen.covar.mat , altitudes , verbose=F) {
  mu <- rep(0, 83)
  Altitude <- altitudes
  GCM <- gen.covar.mat
  B <- slope
  env.vector <- mu + B * Altitude
  VA = add.gen.var
  simpheno <- mvrnorm(n=samples,env.vector, VA*GCM)
  pvals=sapply(1:samples,function(X) EnvVarTest(simpheno[X,],gen.covar.mat,Altitude)[3])
  allVa=sapply(1:samples,function(X) EnvVarTest(simpheno[X,],gen.covar.mat,Altitude)[4])
  hist(pvals,breaks=100, main="Distribution of P Values")
  signif <- length(subset(pvals, pvals < 0.05))
  avgVa <- mean(allVa)
  hist(allVa,breaks=100, main="Histogram of Va Values for 1000 Simulations")
  return ( c ( signif , avgVa ) )
}
```

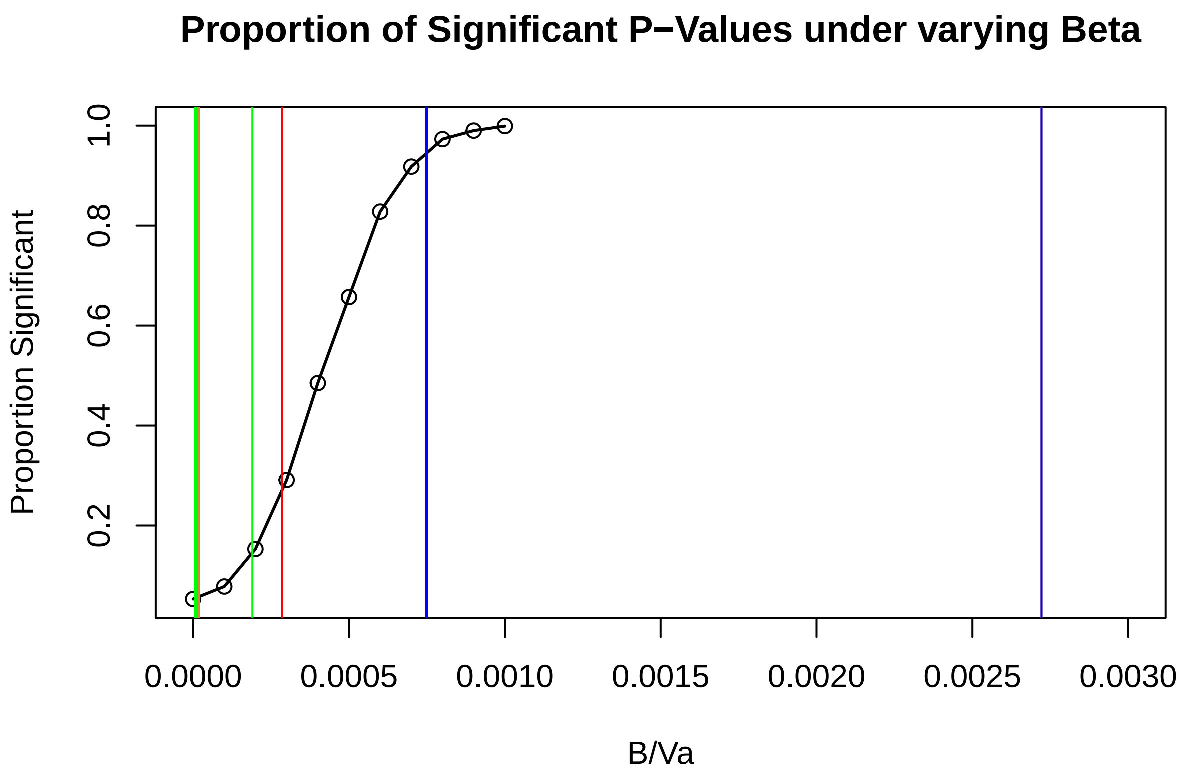
Now that the simulations are working, I want to make a plot of beta/Va, which is the comparable value of s that we observe in our biologically real results. To make the plot, I will run a few of the following sims with changing values for beta. Store those values in vectors, and make a plot! Code for simulations is commented out.

```r
#sim(1000,0,1,A,phenoorder$Altitude) # 53/1000
#sim(1000,0.0001,1,A,phenoorder$Altitude) # 78/1000
#sim(1000,0.0002,1,A,phenoorder$Altitude) # 153/1000
#sim(1000,0.0003,1,A,phenoorder$Altitude) # 291/1000
#sim(1000,0.0004,1,A,phenoorder$Altitude) # 485/1000
```

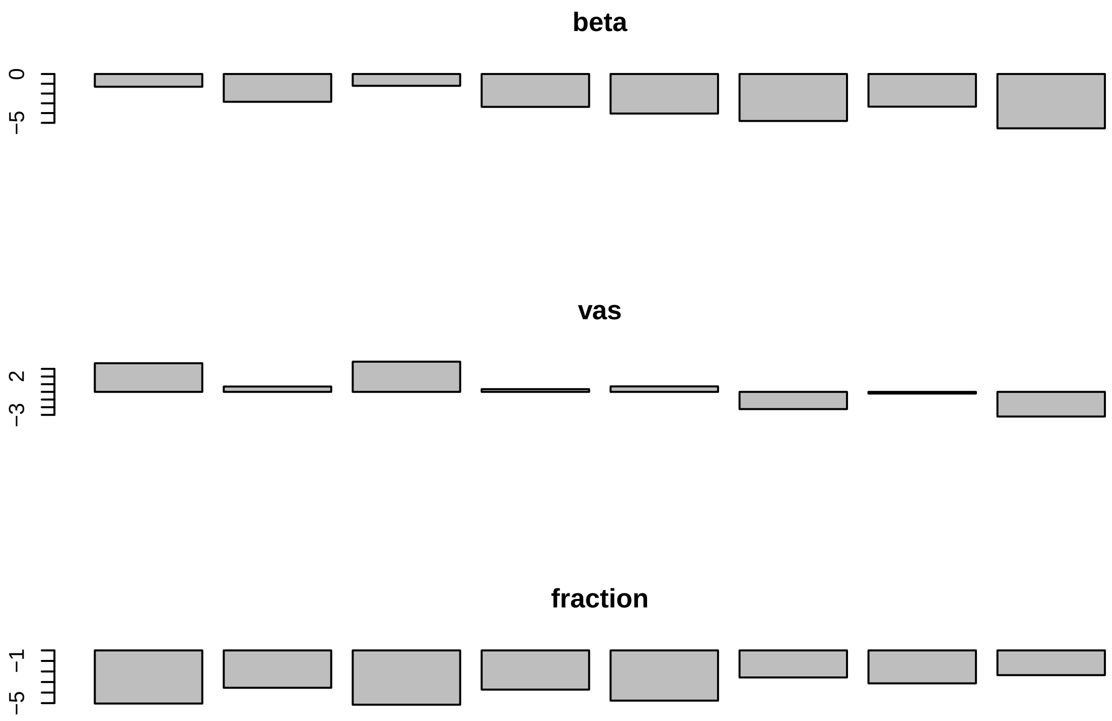
```
#sim(1000,0.0005,1,A,phenoorder$Altitude) # 657/1000
#sim(1000,0.0006,1,A,phenoorder$Altitude) # 828/1000
#sim(1000,0.0007,1,A,phenoorder$Altitude) # 918/1000
#sim(1000,0.0008,1,A,phenoorder$Altitude) # 973/1000
#sim(1000,0.0009,1,A,phenoorder$Altitude) # 990/1000
#sim(1000,0.001,1,A,phenoorder$Altitude) # 999/1000
```

Head to the grindstone on this, thinking about relationship between significance and Beta/Va.

```
accepted <- c(.053 , .078, .153, .291, .485, .657, .828, .918, .973, .990, .999)
beta <- c(0)
beta <- as.numeric(union(beta, seq(0.0001,0.001,0.0001)))
plot(accepted ~ beta, main="Proportion of Significant P-Values under varying Beta", xlab="B/Va", ylab="
lines(beta, accepted, lwd=1.5)
knobmb <- abs(-0.049495790/5431.778287163)
abline(v=knobmb, col="red",lwd=1.5)
knob. <- abs(-0.001419767/4.972794944)
abline(v=knob., col="red")
temb <- abs(-0.061480283/8727.985264731)
abline(v=temb, col="green",lwd=1.5)
te. <- abs(0.0004300344/2.2614178222)
abline(v=te., col="green")
centcmb <- abs(0.00008894284/5.18658746447)
abline(v=centcmb, col="peru",lwd=1.5)
centc. <- abs(0.00001546521/0.00568202942)
abline(v=centc., col="peru")
crmmb <- abs(0.0004516771/0.6025925993)
abline(v=crmmb, col="blue",lwd=1.5)
crm. <- abs(0.000002730368/0.000606835736)
abline(v=centc., col="blue")
```

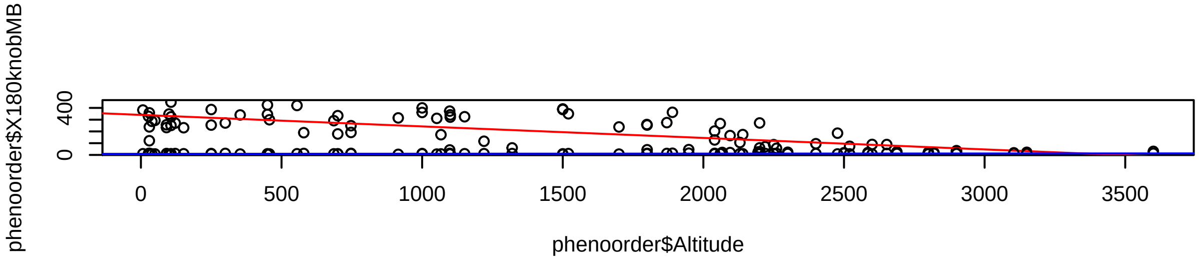


```
betas<- c(0.049495790,0.001419767,0.061480283,0.0004300344,0.00008894284,0.00001546521,0.0004516771,0.0
vas <- c(5431.778287163,4.972794944,8727.985264731,2.2614178222,5.18658746447,0.00568202942,0.602592599
par(mfrow=c(3,1))
barplot(log10(betas),main="beta")
barplot(log10(vas),main="vas")
fraction <- betas/vas
barplot(log10(fraction),main="fraction")
```



```r
plot(phenoorder$Altitude,phenoorder$X180knobMB)
abline(lm(phenoorder$X180knobMB~phenoorder$Altitude),col="red")
points(phenoorder$Altitude,phenoorder$CentCMB)
abline(lm(phenoorder$CentCMB~phenoorder$Altitude),col="blue")
lm(phenoorder$CentCMB~phenoorder$Altitude)
```

```
##
## Call:
## lm(formula = phenoorder$CentCMB ~ phenoorder$Altitude)
##
## Coefficients:
##         (Intercept)  phenoorder$Altitude
##            9.399284             0.000966
```



With these sims, we see that we can pick up beta/Va of pretty low, which will be a solid sentence in the paper. I assume we have to redo these things for the teosinte matrices once I build them.