Adaptive genetic variation

Justification: Natural selection acts on phenotypic variation that is genetically determined. As such, it can be difficult to get a complete picture about adaptation from scanning genomes using molecular markers. The reason is that genetic outliers, even if true positives, have little to no information present about what phenotype they affect and how this phenotype results in fitness differences. Moreover, it is debatable to scan genomes for the presence of outliers if you have yet to demonstrate that the populations being sampled are locally adapted.

Learning objectives: This lab was constructed to give you experience in working with basic quantitative and population genetic analyses useful to testing hypotheses about local adaptation. Phenotypic measurement is undergoing a revolution, so that familiarity with basic methods in quantitative genetics will serve you well in the future. By the end of the laboratory, you should be able to do the following:

1. Construct, fit, and assess linear mixed models (LMMs) to estimate genetic values for a phenotypic trait measured for families existing in a common garden.
2. Use LMMs to estimate heritability of a trait, its differentiation among populations, and its correlation with environment.
3. Test whether or not phenotypic trait differentiation is statistically different than genetic differentiation at random molecular markers.
4. Perform and assess output from basic association analyses linking genetic variation with environmental variation.

Background: The data with which you are working come from a study of western white pine (*Pinus monticola* Dougl. ex D. Don) sampled around the Lake Tahoe Basin of California and Nevada. These data consist of 157 trees sampled from 10 populations (n = 9 to 22 trees/population). Within each population, trees were sampled within three plots. For each plot, GPS coordinates were collected (i.e. each plot in each population has its own GPS coordinates) and used to generate a set of 7 environmental variables. From these trees, needle tissue was collected from which total DNA was extracted and genotyped for 164 single nucleotide polymorphisms (SNPs). Seeds were also collected and planted in a common garden. The seedlings (n = 5 to 35/tree) were measured for several phenotypic traits. The phenotypic trait we will be working with today is known as the carbon isotope ratio ($\delta^{13}C$). It is the ratio of two isotopes of carbon (^{13}C and ^{12}C) relative to an experimental control, and it is strongly correlated with intrinsic water-use efficiency in plants. Plants need water to live, so it is not a stretch of the imagination to believe that this phenotypic trait has a link with plant fitness.

We will thus have three types of data:

1. SNP genotypes for all trees sampled in the field.
2. Environmental data collected from each plot within each population.
3. Phenotypic measurements for 5 seedlings per tree made in a common garden.

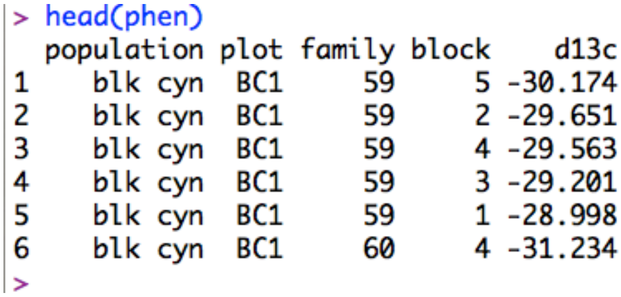
Part 1: Construct, fit, and assess LMMs to estimate genetic values for a phenotypic trait measured for families existing in a common garden

Motivation: A lot of genetics can be carried out without use of any molecular markers. Practitioners of empirical population genetics forget this quite often. A common garden allows us to create a standardized environment in which we minimize the influence of environment on the expression of a particular phenotypic trait. Since we know, after over a century of showing it to be true, that phenotypic variation results from genetic variation, environmental variation, and the interaction of genetic and environmental variation, then if we standardize the environment, phenotypic variation we see in the common garden is due to genetic variation (or if multiple gardens are used, genetics and the interaction of genetics and the environment).

Goals & Background: The goal for this part of the laboratory is to construct, fit, and assess LMMs for δ^{13}C. We will be using the data in the file named `"WWP_phenotypic_data.txt"`. These data are organized in a tab-delimited text file with seedlings grouped by maternal tree (i.e. its mother tree), plot, and population. Also included is an experimental treatment known as "block". In a common garden, seedlings from the same maternal tree are randomized among blocks to avoid the influence of micro-environmental variation on expression of phenotypic traits.

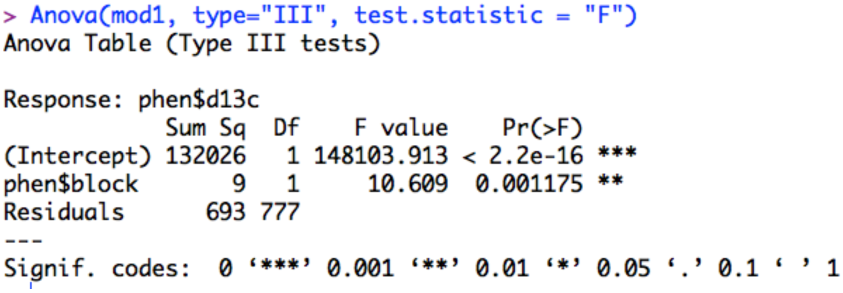
Mechanics: The steps below give you the workflow to complete this part of the laboratory. If a function is unfamiliar to you, to can use either a single or double question mark in front of the function name to call up the help page (e.g. `?library()`).

1. Let's load the required libraries we need using the `library()` function. If you do not have these libraries, you can use the `install.packages()` function to do so.
   a. `library(lme4)`
   b. `library(car)`
2. Let's load the data into R using the `read.delim()` function.
   a. `phen <- read.delim("WWP_phenotypic_data.txt", sep = "\t", header = T)`
   b. Optional: set your working directory to the directory containing the files for analysis using the `setwd()` function.
3. You should always check to see that the loaded object looks sensible prior to doing much analysis. You can do this using the `head()` function, so `head(phen)` should produce a print out to your screen of the first few lines of the `phen` object.



```
> head(phen)
  population plot family block    d13c
1    blk cyn  BC1     59     5 -30.174
2    blk cyn  BC1     59     2 -29.651
3    blk cyn  BC1     59     4 -29.563
4    blk cyn  BC1     59     3 -29.201
5    blk cyn  BC1     59     1 -28.998
6    blk cyn  BC1     60     4 -31.234
>
```

4. Now, we are ready to fit a series of linear models. We will fit three models in total for this laboratory: (a) a fixed effect model with only an intercept, (b) a LMM with an intercept (fixed) and a random effect due to family, and (c) a LMM with an intercept, a random effect due to family nested within population, and a random effect of population. We will thus be ignoring the plot identifiers. All models will also have a fixed effect of block.
    a. Model 1: `mod1 <- lm(phen$d13c ~ 1 + phen$block)`
    b. Model 2: `mod2 <- lmer(d13c ~ 1 + (1|family) + block, data = phen, REML = F)`
    c. Model 3: `mod3 <- lmer(d13c ~ 1 + (1|population) + (1|family) + block, data = phen, REML = F)`
        i. Note you could also have used: `lmer(d13c ~ 1 + (1|population/family) + block, data = phen, REML = F)`
5. We are now ready to explore each of these models and to look at which model best fits our data. First, you can use the `Anova()` function in the `car` library to test the statistical significance of the fixed terms in each mod.
    a. Using the following command you should see the following output for `mod1`. This is an ANOVA table using type III sums of squares (see your stats book).



```
> Anova(mod1, type="III", test.statistic = "F")
Anova Table (Type III tests)

Response: phen$d13c
            Sum Sq  Df    F value    Pr(>F)    
(Intercept) 132026   1 148103.913 < 2.2e-16 ***
phen$block       9   1     10.609  0.001175 ** 
Residuals      693 777                         
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

    b. Do the same for `mod2` and `mod3`. What do you conclude about the effect of block in each of the models?
        i. Remember this is an experimental treatment, so your conclusion directly addresses whether micro-environmental variation exists in your common garden.
6. Now, let’s explore which model best fits our data. To do this we will use the Akaike Information Criterion (AIC). This statistic is like a penalized likelihood score, where the penalty increases as the number of parameters in a model increases. When using AIC, the model with the lowest score is the preferred model. To get AIC values for each model, do something like the following using the `AIC()` function.
    a. `aic_vals <- c(AIC(mod1), AIC(mod2), AIC(mod3)); names(aic_vals) <- c("mod1","mod2","mod3")`
    b. Which model has the lowest AIC score?
    c. After figuring out the answer to (b), we can express the relative support of each model using Akaike weights. These can be thought of as the conditional probabilities of each model. To do this, we will need to write a few lines of R commands. Luckily for you, I have provided a function

named `aic_weights` that will calculate these for you. This function is located in the `"supplemental_R_functions.R"` file. Use `source("supplemental_R_functions.R")` to get these into R.

   i. `aic_out <- aic_weights(aic_vals)`

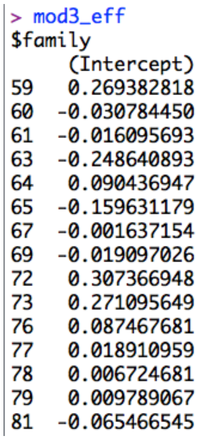
   ii. Inspect the values in `aic_out`. They add to one and can be thought of as conditional probabilities of each model, with the conditioning being on only these three models being examined. Which model has the highest probability? How much larger is it than the other probabilities? What does this tell you about your optimal model?

7. Now that we have the best model, let's use it to calculate the genetic values for each maternal tree for δ^{13}C. To do this, we will work directly with the `mod3` output from before. What we are after is the value of δ^{13}C for each tree from which we measured δ^{13}C from five of her offspring in the common garden. This is the genetic value and represents the value of δ^{13}C that would result if you knew all the genes and effect sizes of variation within those genes determining variation for this trait (see genetics without molecular markers!).

   a. To get the effects due to family and population, we can use the `ranef()` function. This function produces estimates for each family and population in a list with named elements ($family & $population).

      i. First, let's get the list we need: `mod3_eff <- ranef(mod3)`

      ii. Look at the values by printing `mod3_eff`. Shown are the first few lines:



```
> mod3_eff
$family
     (Intercept)
59   0.269382818
60  -0.030784450
61  -0.016095693
63  -0.248640893
64   0.090436947
65  -0.159631179
67  -0.001637154
69  -0.019097026
72   0.307366948
73   0.271095649
76   0.087467681
77   0.018910959
78   0.006724681
79   0.009789067
81  -0.065466545
```

      iii. The values look strange relative to the original values in the `phen` object. It turns out that values in each element are relative to the global intercept listed in the `mod3` output. So, for example, family 59 has a value of δ^{13}C that is 0.269382818 greater than the mean, whereas family 65 has a value of δ^{13}C 0.159631179 less than the mean. Look in the `mod3` output by printing it to screen and finding the global intercept (-30.59964).

      iv. Now, we need to add this number to the values in the $family output in `mod3_effs`: `mod3_fam_only <- mod3_eff$family + -30.59964`

v. We still are not done. Remember that families were nested within populations, so the total effect of a maternal tree was partitioned into an effect of population and trees within populations. Therefore, we should add the population effect to the numbers from iv. To get this we need to replicate the values in the $population part of the list for each tree in each population. You can just use the `pop_rep` function provided in `"supplemental_R_functions.R"`:

```
mod3_all_eff <- mod3_fam_only + pop_rep(pop.eff = 
mod3_eff$population, n.fam = 
nrow(mod3_eff$family), fam.eff = mod3_eff$family)
```

vi. The values held in the object `mod3_all_eff` are now the genetic effects of each maternal tree. In other words, this is the phenotypic trait value for the maternal tree for δ^{13}C. Note that we did not measure the maternal tree, but inferred her phenotype from her offspring in a common environment.

Part 2: Use LMMs to estimate heritability of a trait, its differentiation among populations, and its correlation with environment

<u>Motivation:</u> Now that we have learned how to estimate genetic values for δ^{13}C, let's learn how to estimate what fraction of the total variation in trait values is due to genetic effects and how much of this genetic effect is due to families nested within populations and to populations. These analyses provide key information about whether or not local adaptation should even be considered. Remember that local adaptation is about genetically determined phenotypes that vary across environments in responses to differing selective pressures. This step allows us to assess how genetic variation for a phenotypic trait is distributed across the landscape.

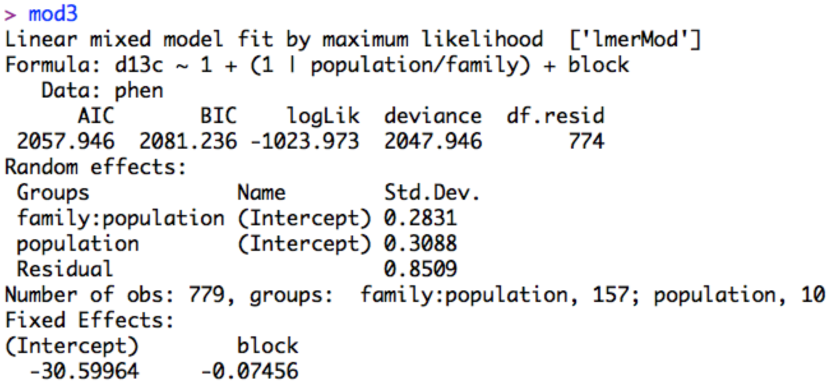
<u>Goals & Background:</u> The goal for this part of the laboratory is to estimate heritability, trait differentiation, and correlation with environment for trait values determined in Part 1. To do this, we will be using the output from the previous part of the laboratory and the environmental data contained in the file named `"WWP_environmental_data.txt"`. As with the phenotype file this is a tab-delimited text file.

<u>Mechanics:</u> The steps below give you the workflow to complete this part of the laboratory. If a function is unfamiliar to you, to can use either a single or double question mark in front of the function name to call up the help page (e.g. `?library()`).

1. Let's start with estimating the heritability of δ^{13}C. If you remember from your undergraduate evolution course, heritability refers generally to the proportion of phenotypic variance due to genetic variance. It comes in at least two different versions. The version we are interested in is narrow-sense heritability (h^2), which is defined as the ratio of additive genetic variance to total phenotypic variance:

$$h^2 = \frac{\sigma^2_{additive}}{\sigma^2_{total}}$$

a. We need to extract the variance components from `mod3` for all model terms. We do this visually by printing mod3 to screen or using a set of functions applied to `mod3`. For this lab, let's do it visually.



```
> mod3
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: d13c ~ 1 + (1 | population/family) + block
   Data: phen
      AIC       BIC    logLik  deviance  df.resid
 2057.946  2081.236 -1023.973  2047.946       774
Random effects:
 Groups            Name        Std.Dev.
 family:population (Intercept) 0.2831
 population        (Intercept) 0.3088
 Residual                      0.8509
Number of obs: 779, groups:  family:population, 157; population, 10
Fixed Effects:
(Intercept)        block
  -30.59964     -0.07456
```

b. Using the results from above, let's calculate h^2. If we assume that the seedlings from each maternal tree are half-siblings (i.e. same mom, but each with a different father) then $\sigma^2_A = 4\sigma^2_{family}$ (so variance due to family:population). If the seedlings were all full-siblings, then the 4 would be replaced with a 2. We also need to realize that we are using a hierarchical model, where some of the genetic effects are due to among populations, where h^2 is a measure within populations. That means we have to ignore the variance due to populations. Let's assume half-siblings. We can then do the following:

   i. `add_var <- 4*(0.2831^2)`

   ii. `total_wp_var <- (0.2831^2) + (0.8509^2)`

   iii. `h2 <- add_var/total_wp_var`

   iv. Inspect your value of h2. What does it mean? Why did we square the values in i. and ii. from above?

c. We have generated a point estimate for h^2. It represents the average h^2 across populations after removing the genetic effects due to population differences. Would it not be nice to also have a confidence interval? We can do that through an approach known as parametric bootstrapping. This approach simulates data using the fitted model a large number of times. Using the resulting distribution, you can create confidence intervals using the appropriate symmetric quantiles of the distribution. To see this, please do the following using the `mod_boot` function in `"supplemental_R_functions.R"`. It will takes a few moments to run step i.

   i. `h2_boot_out <- mod_boot(model = mod3, nboot = 1000)`

   ii. `ci_95 <- quantile(h2_boot_out, probs = c(0.025, 0.50, 0.975)) ### this is a 95% ci.`

   iii. `boxplot(h2_boot_out, range=5); abline(h = h2, col = "red") ## the red line is our original h2` `estimate for comparison to the bootstrap distribution.`

iv. Interpret your results from ii. and iii. Do you think that h^2 is statistically different than zero? Is this consistent with the AIC results from Part 1? Is it meaningful that the red line is very similar to the mean (or median) of the bootstrap distribution? How would I change ii. for a 99% confidence interval?

2. Great, we have shown that within population genetic variation is statistically greater than zero. What about among population genetic variation? Let's get to that right now. To measure among population genetic variation we will use a statistic known as Q_{ST}. It is similar in concept to F_{ST} from population genetics. To estimate Q_{ST}, we will use our LMM output again. If we assume that all seedlings are again half-siblings, then:

$$Q_{ST} = \frac{\sigma^2_{population}}{\sigma^2_{population} + 8\sigma^2_{family}}$$

   a. `num_qst <- 0.3088^2`
   b. `dem_qst <- (0.3088^2) + (8*(0.2831^2))`
   c. `qst <- num_qst/dem_qst`
   d. Inspect your value in qst object. What does it mean? Look at the quantities in the equation above, what is the denominator equal to? Is it the total phenotypic variance or the total genetic variance?

3. Now, we can again look at a confidence interval using parametric bootstrapping. Again, please use the function from those provided to you: `mod_boot_qst()` that is also located in `"supplemental_R_functions.R"`. As before, it will take a few moments for part a. to finish.
   a. `qst_boot_out <- mod_boot_qst(model = mod3, nboot = 1000)`
   b. `ci_95_qst <- quantile(qst_boot_out, probs = c(0.025, 0.50, 0.975)) ### this is a 95% ci.`
   c. `boxplot(qst_boot_out); abline(h = qst, col = "red")`
   d. Interpret your results from ii. and iii. Do you think that Q_{ST} is statistically different than zero? Is this consistent with the AIC results from Part 1? Is it meaningful that the red line is less similar to the mean (or median) of the bootstrap distribution as compared to h^2?

4. The last thing we want to do in this part of the lab is to test for correlations between genetic values of δ^{13}C and environmental data.
   a. First, we need to load the environmental and geographical data.
      i. `env <- read.delim("WWP_environmental_data.txt", sep = "\t", header = T)`
   b. First, let's standardize all the data. This means let's subtract the mean and divide by the standard deviation for each geographical and environmental variable. Luckily, R has a function called `scale()` that will do this for us.
      i. `env2 <- data.frame(matrix(nrow=nrow(env), ncol=ncol(env))); colnames(env2) <- colnames(env)`
      ii. `env2[,c(1:2)] <- env[,c(1:2)]`

iii. `for(i in 3:ncol(env2)) {env2[,i] <- scale(env[,i], center = T, scale = T)}`

c. Second, let's use multiple regression to test the effect of these variables on δ^{13}C. Luckily, our genetic values in `mod3_all_eff` are in the same order as the environmental data.

i. First, join the data: `phen_env <- cbind(mod3_all_eff[,1], env2[,c(3:11)]); colnames(phen_env) <- c("d13c", colnames(env2)[3:11])`

ii. `mod1_env <- lm(d13c ~ longitude + latitude + elev + max_rad + tmax_july + tmin_jan + ann_ppt + gdd_aug + AWS050, data = phen_env)`

1. This model tells us the effect of all variables on δ^{13}C. Use `summary(mod1_env)` to get relevant statistics.

iii. Now, let's get the effect of climate conditioned on longitude and latitude.

1. `res_geog <- residuals(lm(d13c ~ longitude + latitude, data = phen_env))`
2. `phen_env_mod2 <- cbind(res_geog, env2[,5:11]); colnames(phen_env_mod2) <- c("d13c", colnames(env2)[5:11])`
3. `mod2_env <- lm(d13c ~ elev + max_rad + tmax_july + tmin_jan + ann_ppt + gdd_aug + AWS050, data = phen_env, data = phen_env_mod2)`

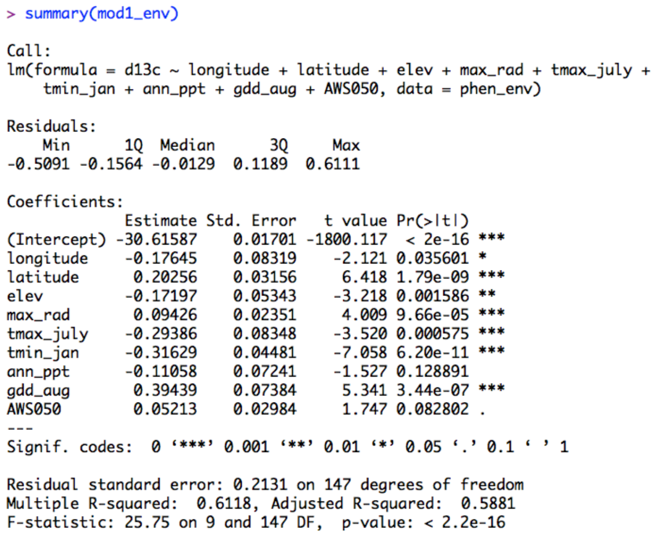
a. This model gives the effect of climate independent of geography. Use `summary(mod2_env)` to get relevant statistics.

iv. Now, let's get the effect of geography conditioned on climate.

1. `res_clim <- residuals(lm(d13c ~ elev + max_rad + tmax_july + tmin_jan + ann_ppt + gdd_aug + AWS050, data = phen_env))`
2. `phen_env_mod3 <- cbind(res_clim, env2[,c(3:4)]); colnames(phen_env_mod3) <- c("d13c", "longitude", "latitude")`
3. `mod3_env <- lm(d13c ~ longitude + latitude, data = phen_env_mod3)`

a. This model gives the effect of geography independent of climate. Use `summary(mod3_env)` to get relevant statistics.

v. We can now assess the impact of climate on the genetic values of δ^{13}C. Let's start with model 1. If you type `summary(mod1_env)`, the following should print to your screen:



```
> summary(mod1_env)

Call:
lm(formula = d13c ~ longitude + latitude + elev + max_rad + tmax_july + 
    tmin_jan + ann_ppt + gdd_aug + AWS050, data = phen_env)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.5091 -0.1564 -0.0129  0.1189  0.6111 

Coefficients:
             Estimate Std. Error   t value Pr(>|t|)    
(Intercept) -30.61587    0.01701 -1800.117  < 2e-16 ***
longitude    -0.17645    0.08319    -2.121 0.035601 *  
latitude      0.20256    0.03156     6.418 1.79e-09 ***
elev         -0.17197    0.05343    -3.218 0.001586 ** 
max_rad       0.09426    0.02351     4.009 9.66e-05 ***
tmax_july    -0.29386    0.08348    -3.520 0.000575 ***
tmin_jan     -0.31629    0.04481    -7.058 6.20e-11 ***
ann_ppt      -0.11058    0.07241    -1.527 0.128891    
gdd_aug       0.39439    0.07384     5.341 3.44e-07 ***
AWS050        0.05213    0.02984     1.747 0.082802 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2131 on 147 degrees of freedom
Multiple R-squared:  0.6118,	Adjusted R-squared:  0.5881 
F-statistic: 25.75 on 9 and 147 DF,  p-value: < 2.2e-16
```

1. Is this multiple regression model statistically significant? If so, why? Which variables provide the largest effects (use the column labeled “Estimate”, which is the partial regression coefficient)? Do the same sort of inspection for `mod2_env` and `mod3_env`. What can you conclude?

Part 3: Test whether or not phenotypic trait differentiation is statistically different than genetic differentiation at random molecular markers.

Motivation: Now that we have shown that genetic variation for δ^{13}C within populations is significantly greater than zero (i.e. $h^2 > 0$), that differentiation for δ^{13}C is statistically greater than zero (i.e. $Q_{ST} > 0$), and that climate, and to a lesser degree geography, is correlated with δ^{13}C values, we can formally test whether or not differentiation for δ^{13}C is unexplainable due to neutral processes such as genetic drift and gene flow. The general idea is use a set of genetic markers we think primarily reflects neutral processes to estimate what Q_{ST} should be without any form of natural selection operating in our system. To do that, we will use 164 SNPs sampled from gene regions that have no apparent functional connection to δ^{13}C. This will allow us to conclude that the differentiation we see is not just different from zero (done before), but different than expectations from a neutral model.

Goals & Background: The goal for this part of the laboratory is to test the hypothesis that Q_{ST} is greater than F_{ST}. We will do that using SNP data that are stored in the file named `"WWP_SNP_genotypes.txt"`. As with the previous files, this is a tab-delimited text file.

Mechanics: The steps below give you the workflow to complete this part of the laboratory. If a function is unfamiliar to you, to can use either a single or double question mark in front of the function name to call up the help page (e.g. `?library()`).

1. Let's load the required libraries we need using the `library()` function. If you do not have these libraries, you can use the `install.packages()` function to do so.
    a. `library(hierfstat)`
    b. `library(QstFstComp)`
2. First, let's examine differentiation for the SNP data using F_{ST}. There are a multitude of ways to do this, but we will use the algorithms and approaches in the `hierfstat` library.
    a. `snp <- read.delim("WWP_SNP_genotypes.txt", sep = "\t", header = T)`
    b. Now, we need to convert the format of the SNP genotypes to FSTAT format for use in `hierfstat.` Please use the `hierfstat_convert()` function from those provided to you in the file named `"supplemental_R_functions.R"`: `snp_reformat <- hierfstat_convert(snp = snp, ids = c(1,2))`
    c. Let's check for weird artifacts. Sometimes, for technical molecular biology reasons, a SNP in a data file turns out to be monomorphic (i.e. it has no variation). We can check this using:
       `mono <- numeric(ncol(snp_reformat)); for (i in 1:ncol(snp_reformat)) {mono[i] <- length(table(snp_reformat[,i]))}; snp_reformat2 <- snp_reformat[,-which(mono == 1)]`
    d. Now, we need to add names to the SNPs that are renaming and create the population identifiers for `hierfstat`. We can do this using the following:
        i. `colnames(snp_reformat2) <- colnames(snp)[3:ncol(snp)][-which(mono == 1)]`
        ii. `pop_ids <- data.frame(matrix(snp[,2], nrow=nrow(snp), ncol = 1)); colnames(pop_ids) <- c("population")`
    e. We can now estimate F_{ST} using the `varcomp.glob` function from `hierfstat: fst <- varcomp.glob(levels = pop_ids, loci = snp_reformat2, diploid = T)`
        i. Note you can also explore bootstrapping across loci to get a confidence interval using the `boot.vc()` function.
    f. Now, let's look at the output. The object `fst` has three elements. The first element is matrix of variance components for each SNP (`$loc`). The columns of this matrix are levels you used from the highest to the lowest (left to right). For us, that means column 1 is the variance component for population, column 2 is the variance component for individual, and column 3 is the variance component for the error (or residual). The second element is the sum of the columns (`$overall`). The last element is a matrix of *F*-statistics (`$F`). These work by using as subscripts the column title relative to the row title, so the first value on the first line is the *F*-statistic for population relative to total (i.e. F_{ST}). It is calculated as:

$$F_{pop,tot} = \frac{\sigma^2_{pop}}{\sigma^2_{pop}+\sigma^2_{ind}+\sigma^2_{error}},$$

where the variances are the variance components from `$overall.`

g. Let's inspect the results by SNP. We can calculate F_{ST} for each SNP (or any locus for that matter) using the equation above and the variance components in `$loc`. I have provided a function to do this for you in the `"supplemental_R_functions.R"` file: `fst_persnp()`

   i. `fst_snp <- fst_persnp(vc = fst$loc, names = colnames(snp_reformat2))`

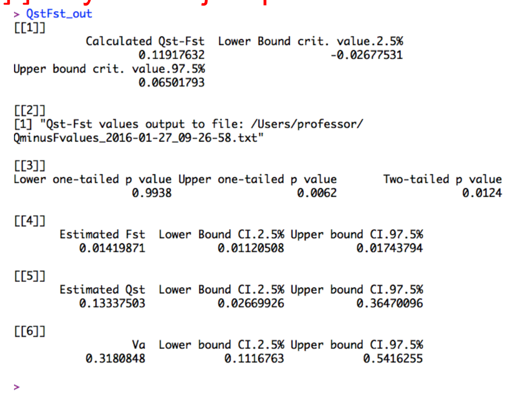
   ii. Inspect the variation across loci relative to the global (multilocus value). Please realize that negative values should be considered as 0. These values are artifacts of estimating the variance components with finite sample sizes. For fun, use the `het_snp()` function in the `"supplemental_R_functions.R"` file to get heterozygosity and plot F_{ST} against heterozygosity: `het_out <- het_snp(snp=snp_reformat2, finite.cor= T, names = colnames(snp_reformat2))`. Look for any trends. To make a plot like those from `Arlequin`, you can divide the heterozygosity values by (1 – F_{ST}) prior to plotting them.

3. Now that we have inspected overall genetic differentiation among populations, let's use the `QstFstComp` library to formally test whether or not Q_{ST} > F_{ST} for δ^{13}C.

   a. `phen_mod <- phen[,-c(2,4)]; snp_reformat3 <- cbind(snp[,2], snp_reformat2); colnames(snp_reformat3)[1] <- c("population")`

   b. `QstFst_out <- QstFstComp(fst.dat = snp_reformat3, qst.dat = phen_mod, numpops = 10, nsim = 10000, breeding.design = "half.sib.dam", dam.offspring.relatedness = 0.25, output = "concise")`

   c. Inspect the output located in `QstFst_out`. Look at the first and third elements of this list. You can call them using `QstFst_out[[1]]` and `QstFst_out[[3]]` or you can just print the entire object to your screen:



```
> QstFst_out
[[1]]
           Calculated Qst-Fst  Lower Bound crit. value.2.5%
                   0.11917632                   -0.02677531
Upper bound crit. value.97.5%
                   0.06501793

[[2]]
[1] "Qst-Fst values output to file: /Users/professor/
QminusFvalues_2016-01-27_09-26-58.txt"

[[3]]
Lower one-tailed p value Upper one-tailed p value       Two-tailed p value
                  0.9938                   0.0062                   0.0124

[[4]]
       Estimated Fst  Lower Bound CI.2.5% Upper bound CI.97.5%
          0.01419871           0.01120508           0.01743794

[[5]]
       Estimated Qst  Lower Bound CI.2.5% Upper bound CI.97.5%
          0.13337503           0.02669926           0.36470096

[[6]]
                  Va  Lower bound CI.2.5% Upper bound CI.97.5%
           0.3180848            0.1116763            0.5416255

>
```

Is Q_{ST} > F_{ST}? What does this mean biologically? Why is the estimated Q_{ST} a little higher here as opposed to the point estimate from before?

Part 4: Perform and assess output from basic association analyses linking genetic variation with environmental variation.

Motivation: So far we have shown the following: δ^{13}C is genetically determined, genetically structured among populations, correlated to a variety of environmental variables, and that the structuring of genetic diversity for δ^{13}C is statistically greater than that for random SNP markers. That sounds awesome, but we can go a bit further. Although we assumed that the SNP markers had no effect on δ^{13}C and were neutral, we should probably check to see if that is true. To do so, we will use an environmental association approach that also corrects for background levels of population structure. Alternatively, if we had other markers that were good candidates to affect δ^{13}C we could see if they were consistent with acting as the genetic architecture of δ^{13}C, which appears to contribute to local adaptation of *P. monticola* in the Lake Tahoe Basin.

Goals & Background: The goal of this part of the laboratory is to explore environmental associations of the SNP genotypes with the environmental data. We will do this with the relatively new R package named `LEA`, which allows use of latent factor mixed models (LFMM) to carry out the analysis (see more at: https://www.bioconductor.org/packages/release/bioc/html/LEA.html).

Mechanics: The steps below give you the workflow to complete this part of the laboratory. If a function is unfamiliar to you, to can use either a single or double question mark in front of the function name to call up the help page (e.g. `?library()`)

1. Let's load the required libraries we need using the `library()` function. If you do not have these library, you can use the following to get it:

   ```
   source("http://bioconductor.org/biocLite.R")
   biocLite("LEA")
   ```

   a. `library(LEA)`

2. Now, we have to reformat our data yet again. Do you not love software designers and their sadomasochistic need to make us do this? Okay, rant over. Let's get to work.

   a. We need to put the genotype data in a format labeled as `lfmm` for `LEA`. This format is a matrix, with sampled trees as rows and SNPs as columns. In each cell of this matrix is a 0, 1, or 2, which represents the count of a reference allele for that sampled tree for that SNP. Our reference allele will be the minor allele (i.e. the one with the lowest sample frequency across all sampled trees). I have provided a function to do this in the `"supplemental_R_functions.R" file: geno_reformat()`. This function operates on the `snp_reformat2` object from our previous work: `geno_snp <- geno_reformat(snp = snp_reformat2); write.table(t(geno_snp), file = "geno_format.lfmm", sep = "\t", row.names = F, col.names = F)`

b. Now, we need to do the same for the environmental data. Let's use the centered and scaled data from before, which are located in the `env2` object: `env3 <- env2[,-c(1:4)]; write.table(env3, file = "lfmm_env.env", sep = "\t", row.names = F, col.names = F)`

3. We should now be ready to use the `lfmm` function in `LEA`. Let's use the following command to run a simple analysis (note that the paths here assume that the needed files are in your working directory):
   a. `lfmm_out <- lfmm(input.file = "geno_format.lfmm", environment.file = "lfmm_env.env", K = 5, project = "new", missing.data = T, all = T, iterations = 10000, burnin = 5000)`
   b. You can now spend some time varying parameters. I would start with *K* and vary it from 2 to 10 to see how the results change. To get an idea of which *K* might be best, you can do the following. Use the `pca_out <- pca(input.file="geno_format.lfmm", scale = T, center = T)` and `tracy.widom(<output from pca>)` functions from LEA to determine the number of PCs that best explain your data. If you do this, look in the output of `tracy.widom()` for the column labeled pvalue. Select *K* as the number of PCs with this value below a threshold such as $P = 0.05$. Some of the main results you should inspect are the *p*-values for association of a SNP with an environmental gradient. These can be extracted for each environmental variable using the `mlog10p.values()` function: e.g. `mlog10p.values(lfmm_out, K = 5, d = 1)` will extract the $-\log_{10}$ *p*-values for the first environmental variable (higher values means lower *p*-values from testing the null hypothesis of no association between genotypic variation and environmental variation). Which environmental variables have the highest $-\log_{10}$ *p*-values? If you correct for multiple tests per environmental variable using a Bonferroni correction, the $-\log_{10}$ *p*-value threshold is: `-log10(0.05/160) = 3.50515`. Which environmental variables have SNPs with $-\log_{10}$ *p*-values greater than this threshold? How do these relate to the partial regression coefficients from the multiple regression model linking genetic values of $\delta^{13}C$? Are the SNPs with high –log10 *p*-values those with F_{ST} larger than the multilocus values?