R Notebook

Here we provide an example for the covariance analysis. We use the dataset on amphibian (frog) development that was provided by Albecker and McCoy 2019 and is an example presented in the paper.

First steps are to load the data and the functions. These are available on the Github (https://github.com/RCN-ECS/CnGV/tree/master/src/)

```
# Packages
if (!require('lme4')) install.packages('lme4'); library('lme4')
```

```
## Loading required package: lme4
```

```
## Loading required package: Matrix
```

```
if (!require('emmeans')) install.packages('emmeans'); library('emmeans')
```

```
## Loading required package: emmeans
```

```
if (!require('tidyverse')) install.packages('tidyverse'); library('tidyverse')
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x tidyr::unpack() masks Matrix::unpack()
```

```
emm_options(msg.interaction = FALSE)

# Functions (may need to change your working directory)
source("~/Documents/GitHub/CnGV/CnGV/src/CovarianceDataFunctions.R")

# Dataset
frog <- read.csv("~/Documents/GitHub/CnGV/CnGV/data/frog.csv")
```

Background Information on study data

This dataset was produced in a study investigating differences in the length of the larval duration in response to salt-exposure for frogs (tadpoles) that were collected from either inland (freshwater, salt- naive) or coastal (brackish, putatively salt-adapted) populations.

We assume that each population is a separate genotype. The three inland populations ("G_1", "G_2", "G_3") are native to freshwater which is represented as "E_1". The four coastal populations ("G_4", "G_5", "G_6","G_7") are native to "saltwater" which is represented as "E_2".

The phenotype was the number of days it took tadpoles to reach metamorphosis (defined as the day of forelimb emergence). We include just 2 treatments, Freshwater ("E_1") and saltwater (6 parts per thousand; "E_2") to ensure that each genotype could be directly paired with its native environment.

In this dataset, we use "gen_factor" to refer to genotype, "exp_env_factor" to refer to experimental treatment, and "nat_env_factor" to refer to native environment. As specified in the paper, the native enviornment of each genotype needs to match on eof the experimental environments (i.e., the native environment must be one of the experimental treatments) and be named, "nat_env_factor".

The next step is to ensure the data are formatted for the analysis. We use an ANOVA to extract estimated marginal means for each genotype and environmental mean (yi and yj in CovGE equation), which requires that fixed effects (genotype and environment) are categorical.

```
frog$gen_factor = factor(frog$gen_factor)
frog$exp_env_factor = factor(frog$exp_env_factor)
frog$nat_env_factor = factor(frog$nat_env_factor)
```

Standardize data:

We standardize the phenotypic data by subtracting the overall mean phenotype from each phenotypic datapoint and dividing by the standard deviation of group means (group = genotype and experimental environment pair).

```
# Raw data:
frog$group = paste(frog$gen_factor,frog$exp_env_factor,sep = "-")
frog$phen_corrected = (frog$phen_data - mean(frog$phen_data, na.rm = TRUE))/sd(tapply(frog$phen_data, f
```

Balanced data?

Because unbalanced numbers of genotypes affects the analysis, we need to check to see if the data is balanced. We can do this by counting the number of genotypes that are native to the different environments.

Note that true reciprocal transplant designs should always be balanced, but common garden are more likely to be imbalanced. As seen below, because the I exclude one genotype from the inland group (E_1) due to zero survival in one of the treatments, this means I have 4 genotypes from Coastal (E_2) locations, and 3 genotypes from inland locations, which means this design is unbalanced.

Because unbalanced designs affect the overall mean (y_bar in the CovGE equation) and environmental mean (Yj_bar in the CovGE equation), we need to conduct a second ANOVA with a grouping variable (grouped according to native environment - see details in Supplemental Materials)

```
with(frog, tapply(gen_factor, nat_env_factor, function(X) length(unique(X))))
```

```
## E_1 E_2
##   3   4
```

```
# Assign groups according to native environment using reference dataframe (groupDF)
groupDF = data.frame("nat_env_factor" = c("E_1", "E_2", "E_3", "E_4"),
                     "group" = c("A","B","C", "D"))
frog$group = groupDF$group[match(frog$nat_env_factor, groupDF$nat_env_factor)] # its fine that we're re
```

Calculate CovGE:

To start out, we ran a basic categorical linear model to generate yi parameters via estimated marginal means. Estimated marginal means are more robust given unbalanced study designs.

Because the design is unbalanced, we will run a second anova to generate yj and y_bar parameters that correct for the bias

After running the model, we extract estimated marginal means using function emmeans().

```
# Anova for yi
aov.test <- lm(phen_corrected ~ exp_env_factor * gen_factor, data = frog)

# Anova for yj and ybar
aov.test2 <- lm(phen_corrected ~ exp_env_factor * group, data = frog)

# Estimated Marginal Means
emm_df1 = as.data.frame(emmeans(aov.test, ~ exp_env_factor*gen_factor))
emm_df2 = as.data.frame(emmeans(aov.test2, ~ exp_env_factor*group))
```

We next we need to calculate yi (genotypic means) and yj (experimental environment means) and y_bar (overall means).

We do this using tapply() to calculate the mean phenotype for each genotype ACROSS environments (G_matrix), and then the mean phenotype for each environment ACROSS genotypes (E_matrix).

Because this is a common garden design, there should be more genotypic means than environmental means.

```
# Use emms from aov.test1
G_matrix <- data.frame("G_means" = tapply(emm_df1$emmean, emm_df1$gen_factor, mean, na.rm=TRUE),
                       "gen_factor" = unique(emm_df1$gen_factor))

# Use emms from aov.test2
E_matrix <- data.frame("E_means" = tapply(emm_df2$emmean, emm_df2$exp_env_factor, mean, na.rm=TRUE),
                       "exp_env_factor" = unique(emm_df2$exp_env_factor))
```

To match each genotypic mean with the correct environmental mean, we have to ensure genotypes are correctly matched to their native environment. This is what the "I" term in the CovGE equation refers to.

Because there are more genotypes than environments, environments will be used more than once.

```
# First create Native Environment reference dataframe
native_df = data.frame("gen_factor" = unique(frog$gen_factor))
native_df$nat_env_factor = frog$nat_env_factor[match(native_df$gen_factor,frog$gen_factor)]

# Next reorder Gmatrix and Ematrix to reflect the above native environment
Cov_matrix = G_matrix
Cov_matrix$exp_env_factor = native_df$nat_env_factor[match(G_matrix$gen_factor,native_df$gen_factor)] #
Cov_matrix$E_means = E_matrix$E_means[match(Cov_matrix$exp_env_factor,E_matrix$exp_env_factor)]
```

Because G_1, G_2, G_3, and G_4 are all native to the same environment (in this case, a saltwater environment or E_2), they share the same E_mean. Similarly, because G_5, G_6, and G_7 are all native to the same environment (freshwater environment), they share the same E_mean.

Now we have the estimated marginal means and they are formatted properly, we can calculate CovGE.

```
N = length(Cov_matrix$gen_factor) # Length of number of genotypes
overallmean = mean(c(Cov_matrix$G_means,Cov_matrix$E_means),na.rm=TRUE) # ybar (overall mean phenotype
numerator = sum((Cov_matrix$G_means - overallmean)*(Cov_matrix$E_means - overallmean)) # Follows Numera

standardize_max = max(var(Cov_matrix$E_means),var(Cov_matrix$G_means)) # standardize CovGE by max varia
CovGE = (1/(N-1))*(numerator/standardize_max)
CovGE
```

```
## [1] -0.4208096
```

CovGE is -0.42, countergradient variation!

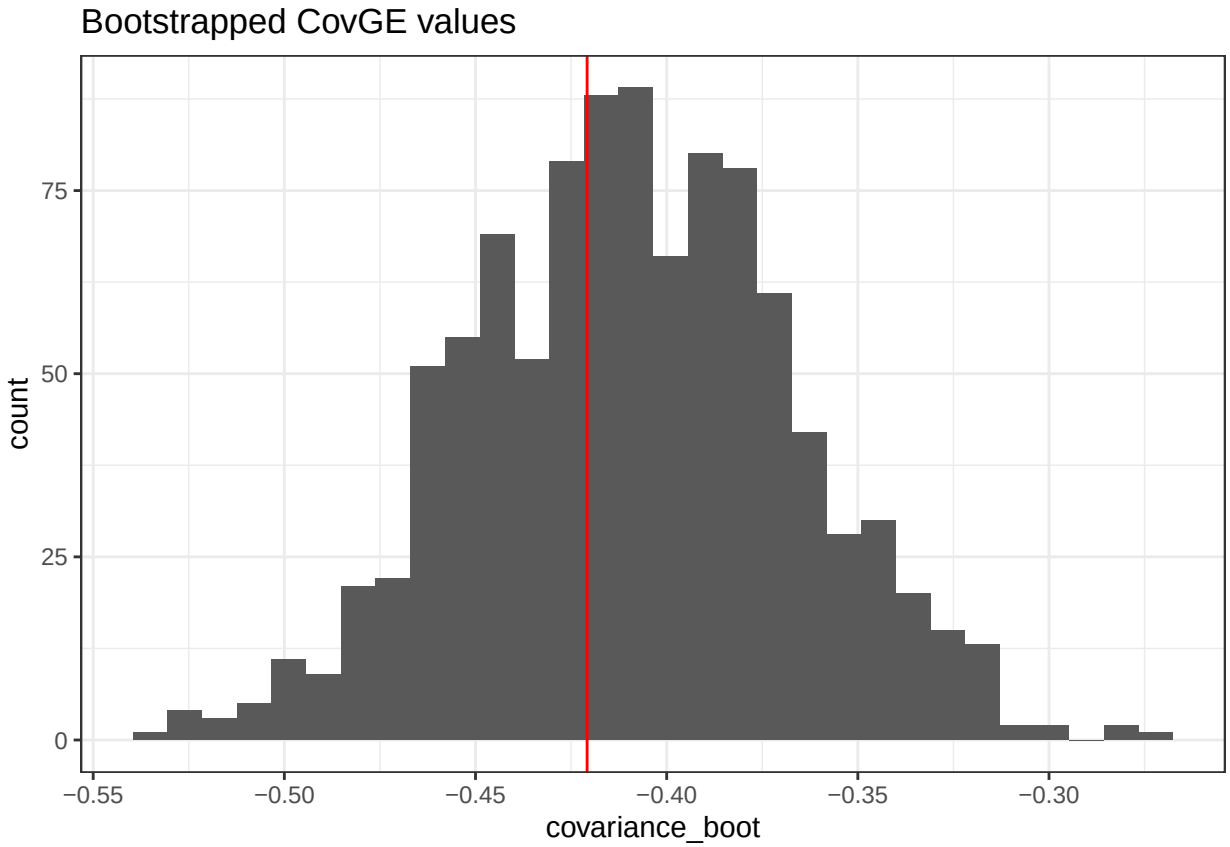
Bootstrapped Confidence Intervals

To estimate confidence intervals, we use bootstrapping, in which we shuffle phenotype within each genotype/environment and recalculate covGE after each reshuffle. This generates a distribution of CovGE estimates that form 95% confidence intervals.

For this step, I am going to use imported functions that do the same thing as shown above. "bootstrap_raw" function shuffles the raw datadata, "mod.GxE" function compiles the cov_matrix dataframe "cov.function" function calculates CovGE

```
n_boot <- 999
balanced = FALSE # Desnotes unequal numbers of genotypes for functions

boot_dat_raw = boot_df_raw = data.frame()

for(i in 1:n_boot){

  # Shuffle Data
  shuffle_dat <- bootstrap_raw(frog)

  # Anova model fit & GxE estimates
  m2 <- mod.Cov(shuffle_dat, balanced) # Insert shuffled raw phenotype dataframe

  # Pull info from mod.GxE output
  cov_matrix_boot <- m2[[1]]

  # Covariance Estimates
  cov_corrected_boot = round(cov.function(cov_matrix_boot,balanced),3)

  # Bootstrap dataframe
  boot_dat_raw <- data.frame("covariance_boot" = cov_corrected_boot)
  boot_df_raw <- rbind(boot_df_raw,boot_dat_raw)
}

# Check: Histograms of distribution around CovGE - Should be around verticle line which is CovGE estima
ggplot(boot_df_raw, aes(x = covariance_boot), alpha = 0.5) + geom_histogram() + geom_vline(aes(xinterce
  ggtitle("Bootstrapped CovGE values") + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Covariance Confidence Intervals
cov_CI = quantile(boot_df_raw$covariance_boot, probs=c(0.025, 0.975), type=1)
cov_CI
```

```
##   2.5%  97.5%
## -0.492 -0.325
```

The 95% confidence intervals are -0.48 to -0.34.

Hypothesis Testing using Permutation

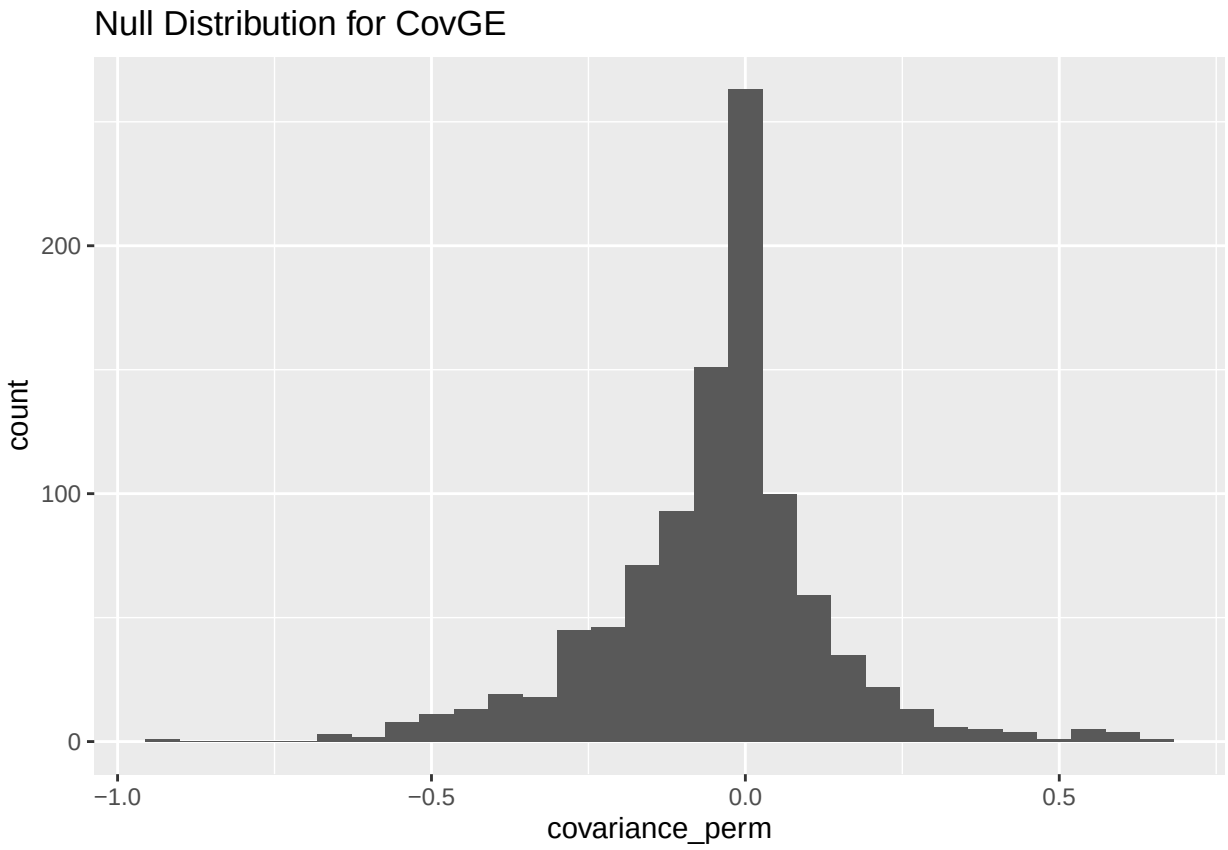
For hypothesis testing, we used permutation. Permutation also resamples phenotypic data but does not resample or maintain the genotypic/environmental levels. As a result, it creates a distribution around the null expectation that CovGE = 0. If the CovGE estimate (-0.42) is outside of the tails of this null distribution, it is considered statistically significant.

Again, I will used some canned functions that simply speed up the above CovGE calculation. "permutation_raw" permutes data. "mod.GxE" again takes those permuted data and generates Cov_matrix dataframes "cov.function" calculates CovGE of permuted data

```
# Output dataframe
perm_df_raw = perm_dat_raw = data.frame()

for(i in 1:n_boot){

  # Resample Data
  perm_dat <- permutation_raw(frog)

  # Anova model fit & GxE estimates
```

```
  m3 <- mod.Cov(perm_dat, balanced) # Insert permuted data

  # GxE Estimates
  cov_matrix_perm <- m3[[1]]

  # Covariance Estimates
  cov_corrected_perm = round(cov.function(cov_matrix_perm,balanced),3)

  # Permutation dataframe
  perm_dat_raw <- data.frame("covariance_perm" = cov_corrected_perm)
  perm_df_raw <- rbind(perm_df_raw,perm_dat_raw)
}

# Check: Permutation histogram - should be around zero
ggplot(perm_df_raw, aes(x = covariance_perm), alpha = 0.5)+ geom_histogram()  + ggtitle("Null Distributi
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Now to calculate the p-value from this null distribution:

```
# Covariance P-values (proportion of values that fall outside the estimated covGE value)
cov_pvalue <- sum(abs(perm_df_raw$covariance_perm) >= abs(CovGE))/(n_boot+1) # Two-tailed
cov_pvalue
```

```
## [1] 0.049
```

Congratulations! You have estimated CovGE, 95% confidence intervals, and the P-value for age to metamorphosis phenotype between coastal and inland frogs. From these results, we can infer that coastal frogs do

exhibit significant counter gradient variation in time of larval development, suggesting that they have evolved to develop faster in saline environments which typically promotes slower development.