Untitled

KE Lotterhos

9/18/2020

Calculating environment distances

1) Create fake environments

- Correlated with Env 1 (~0.5)
- Correlated with Env 2 (~0.5)
- Random environments with mean 0 and sd 1

2) Make sure each environment is standardized by subracting the mean and dividing by the sd. (This should also be true for what is being put into GF)

3) Calculate the following between all populations (although technically, only needed for core and edge populations):

- Euclidean distance for selective environments
- Mahalanobis distance for selective environments
- Euclidean distance for ALL environments
- Mahalanobis distance for ALL environments

```r
CGfit <- read.csv("Common_Garden_fit.csv")
Popsenv <- read.csv("Pops_env.csv")

head(CGfit)
head(Popsenv)
```

Create fake environments

Here I am going to create 2 fake environments, each correlated about 0.5 with the selective environment.

In addition, I am going to create 10 more fake environments with a multivariate normal distribution. The covariance matrix for the mvnorm was generated with a positive definite matrix/covariance matrix using the `genPositiveDefMat` function from the `clusterGeneration` package v1.3.4, using the `unifcorrmat` option. This generates the covariance matrix by sampling the correlation among variables from a uniform distribution. https://www.rdocumentation.org/packages/clusterGeneration/versions/1.3.4/topics/genPositiveDefMat

```r
fakeEnv1 <- Popsenv$envPop1 + rnorm(nrow(Popsenv),0,1.3)
  # this standard deviation generally produces a correlation between 0.3 and 0.6
cor(Popsenv$envPop1, fakeEnv1)

fakeEnv2 <- Popsenv$envPop2 + rnorm(nrow(Popsenv),0,1.3)
  # this standard deviation generally produces a correlation between 0.3 and 0.6
cor(Popsenv$envPop2, fakeEnv2)

Popsenv$fakeEnv1 <- fakeEnv1
Popsenv$fakeEnv2 <- fakeEnv2
dim(Popsenv)
```

```
nfake <- 10
Popsenv[,6:(5+nfake)] <- NA
head(Popsenv)

# All I'm doing here is creating environments with a covariance structure
#cov1 <- genPositiveDefMat(nfake, covMethod="eigen", rangeVar=c(1,10))
cov1 <- genPositiveDefMat(nfake,covMethod="unifcorrmat" )
head(cov1)

a<- mvrnorm(nrow(Popsenv),mu=rep(0, nfake), Sigma=cov1$Sigma)

Popsenv[,6:(5+nfake)] <- a
head(Popsenv)
tail(Popsenv)

sel_env_cols <- 2:3
all_env_cols <- 2:ncol(Popsenv)
```

Standardize environments

```
head(Popsenv)
means <- colMeans(Popsenv[all_env_cols])
  # beware of hard coding columns here
sds <- apply(Popsenv[all_env_cols], 2, sd)

PopsenvStnd <- Popsenv
for (i in all_env_cols){
  PopsenvStnd[,i] <- (Popsenv[,i] - means[i-1])/sds[i-1]
}
head(PopsenvStnd)

# Check for mistakes
round(colMeans(PopsenvStnd[all_env_cols]))
round(apply(PopsenvStnd[all_env_cols], 2, sd))

round(cov(PopsenvStnd[,all_env_cols]),2)
```

Understand CG fit

In this dataframe, it appears `Home` is the site of the common garden. `Transplant` is the location that the genotype came from `Fitness` is the average fitness of the individuals from the source location

`D_CI` is GF_offset_genome?

`D_CI_sel` is GF_offset for the causal loci?

```
head(CGfit)
```

Understanding Mahalanobis

?mahalanobis We are interested in calculating the Mahalanobis distance between pop1 and pop2, while controlling for the covariance among the environmental variables in the population

Let's look at an example where we take the Md between population 1 and population 50, for all the environments

```
(envpop1 <- PopsenvStnd[1,all_env_cols])
(envpop2 <- PopsenvStnd[50,all_env_cols])

# We calculate the covariance based on the entire landscape:
cov_allEnv <- cov(PopsenvStnd[,all_env_cols])
round(cov_allEnv,2)

mahalanobis(as.numeric(envpop1),
            as.numeric(envpop2),
            cov_allEnv)

# sanity check
mahalanobis(as.numeric(envpop1),
            as.numeric(envpop1),
            cov_allEnv)

#compare to eucl.
dist(rbind(envpop1, envpop2))
```

Calculate environment distances

```
cov_allEnv <- cov(PopsenvStnd[,all_env_cols])
cov_selEnv <- cov(PopsenvStnd[,sel_env_cols])

head(PopsenvStnd)

CGfit$EdSelEnv <- NA
  # Euclidean distance for selective environments

CGfit$MdSelEnv <- NA
  # Mahalanobis distance for selective environments

CGfit$EdAllEnv <- NA
  # Euclidean distance for ALL environments

CGfit$MdAllEnv <- NA
  # Mahalanobis distance for ALL environments

head(CGfit)

for (i in 1:nrow(CGfit)){
  # get the row in PopsenvStnd for the transplant genotype
  row1 = which(PopsenvStnd==gsub("T","P",as.character(CGfit$Transplant[i])))
    # get the row in PopsenvStnd for the common garden location
  row2 = which(PopsenvStnd==gsub("H","P",as.character(CGfit$Home[i])))

  # Look up the envi
  (envpop1_all <- PopsenvStnd[row1,all_env_cols])
  (envpop2_all <- PopsenvStnd[row2,all_env_cols])
  # Look up the envi
  (envpop1_sel <- PopsenvStnd[row1,sel_env_cols])
  (envpop2_sel <- PopsenvStnd[row2,sel_env_cols])
  # BEWARE HARD CODING


  ### Calculate the environmental distance between the two rows

  CGfit$EdSelEnv[i] <- dist(rbind(envpop1_sel,
                                  envpop2_sel))
  # Euclidean distance for selective environments

  CGfit$MdSelEnv[i] <- mahalanobis(as.numeric(envpop1_sel),
            as.numeric(envpop2_sel),
            cov_selEnv)
  # Mahalanobis distance for selective environments

  CGfit$EdAllEnv[i] <- dist(rbind(envpop1_all,
                                  envpop2_all))
  # Euclidean distance for ALL environments

  CGfit$MdAllEnv[i] <- mahalanobis(as.numeric(envpop1_all),
            as.numeric(envpop2_all),
            cov_allEnv)
  # Mahalanobis distance for ALL environments
}
```

Calculate environment distances

```
head(CGfit)

#any missing data?
sum(!complete.cases(CGfit))
# should be 0

plot(CGfit$Fitness[CGfit$Home=="H1"]~
       CGfit$D_CI[CGfit$Home=="H1"])

plot(CGfit$Fitness[CGfit$Home=="H1"]~
       CGfit$EdSelEnv[CGfit$Home=="H1"])

plot(CGfit$Fitness[CGfit$Home=="H1"]~
       CGfit$MdSelEnv[CGfit$Home=="H1"])

plot(CGfit$Fitness[CGfit$Home=="H1"]~
       CGfit$EdAllEnv[CGfit$Home=="H1"])
abline(lm(CGfit$Fitness[CGfit$Home=="H1"]~
            CGfit$EdAllEnv[CGfit$Home=="H1"]))
(cor(CGfit$Fitness[CGfit$Home=="H1"],
     CGfit$EdAllEnv[CGfit$Home=="H1"]))

plot(CGfit$Fitness[CGfit$Home=="H1"]~
       CGfit$MdAllEnv[CGfit$Home=="H1"])
abline(lm(CGfit$Fitness[CGfit$Home=="H1"]~
            CGfit$MdAllEnv[CGfit$Home=="H1"]))
cor(CGfit$Fitness[CGfit$Home=="H1"],
    CGfit$MdAllEnv[CGfit$Home=="H1"])
```

Results for Euclidean Dist and Mahalanobis are similar because we standardize the environments to have an SD=1 prior to analysis

Some notes:

When I first started, I had 2 fake environments (each correlated with one of the selective environments) and 1 random fake environment. This decreased the correlation between EdAllEnv and Fitness, but only slightly (cor ~ -0.8)

Then, I increased it to 10 random fake environments (with no correlation structure), which decreased it more (cor ~ -0.5)

Then, I added covariance structure to the environments, which decreased it more (cor ~ -0.35)

I think we should use the type of environmental data that I generated here, because it retains some realism that is present in empirical data