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)
```

```
##   Transplant Home  Fitness        D_CI   D_CI_sel Env_sel1 Env_sel2     dM
## 1         T1   H1 0.790825 0.000000000 0.00000000       -1       -1 3.9996
## 2         T2   H1 0.817109 0.003351372 0.01194520       -1       -1 3.9996
## 3         T3   H1 0.711378 0.005807787 0.02071331       -1       -1 3.9996
## 4         T4   H1 0.521529 0.012116530 0.04544032       -1       -1 3.9996
## 5         T5   H1 0.321762 0.016701773 0.05602929       -1       -1 3.9996
## 6         T6   H1 0.321762 0.016701773 0.05602929       -1       -1 3.9996
```

```r
head(Popsenv)
```

```
##   Pop envPop1 envPop2
## 1  P1    -1.0      -1
## 2  P2    -0.5      -1
## 3  P3     0.0      -1
## 4  P4     0.5      -1
## 5  P5     1.0      -1
## 6  P6     1.0      -1
```

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)
```

```
## [1] 0.4488023
```

```r
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)
```

```
## [1] 0.6069238
```

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

```
## [1] 100   5
```

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

```
##   Pop envPop1 envPop2    fakeEnv1  fakeEnv2 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15
## 1  P1    -1.0      -1 -1.55766341 -1.218331 NA NA NA NA  NA  NA  NA  NA  NA  NA
## 2  P2    -0.5      -1 -0.30192757 -3.453716 NA NA NA NA  NA  NA  NA  NA  NA  NA
## 3  P3     0.0      -1  0.07655442 -1.083442 NA NA NA NA  NA  NA  NA  NA  NA  NA
## 4  P4     0.5      -1  1.74133551 -1.534257 NA NA NA NA  NA  NA  NA  NA  NA  NA
## 5  P5     1.0      -1  1.06068083 -1.822725 NA NA NA NA  NA  NA  NA  NA  NA  NA
## 6  P6     1.0      -1 -1.99403263 -0.799573 NA NA NA NA  NA  NA  NA  NA  NA  NA
```

```r
# 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)
```

```
## $egvalues
##  [1] 14.3810019 12.3649794  8.1594124  5.3925970  4.7695312  2.4840682
##  [7]  2.1885882  1.2331614  0.6088628  0.3122503
## 
## $Sigma
##              [,1]        [,2]       [,3]        [,4]       [,5]       [,6]
##  [1,]  1.46934833 -0.00838753  0.7582294 -0.45018087 -0.7707050 -1.0488042
##  [2,] -0.00838753  4.46040816  1.7888387  0.52981033  2.0275569 -0.9965486
##  [3,]  0.75822936  1.78883871  7.1348158 -2.09661109 -0.2773871 -0.2540386
##  [4,] -0.45018087  0.52981033 -2.0966111  5.68883846  0.4832758  0.4856977
##  [5,] -0.77070500  2.02755687 -0.2773871  0.48327582  5.0120917 -0.8678556
##  [6,] -1.04880420 -0.99654860 -0.2540386  0.48569769 -0.8678556  6.3739536
##  [7,]  0.80878943 -2.78730276  0.5645447 -2.90927683 -0.6271381 -1.7279472
##  [8,]  0.43752124  0.34746691 -1.3908978 -0.09202066 -1.0997294  0.5916816
##  [9,]  0.82919724  2.42988097  4.9089904 -0.94865381 -1.8774159 -0.5038826
## [10,]  0.00191363  0.58647739  0.6948003  0.92950386  0.6626657 -0.5107166
##             [,7]        [,8]       [,9]      [,10]
##  [1,]  0.8087894  0.43752124  0.8291972  0.00191363
##  [2,] -2.7873028  0.34746691  2.4298810  0.58647739
##  [3,]  0.5645447 -1.39089777  4.9089904  0.69480026
##  [4,] -2.9092768 -0.09202066 -0.9486538  0.92950386
##  [5,] -0.6271381 -1.09972936 -1.8774159  0.66266567
```

```
##  [6,] -1.7279472  0.59168161 -0.5038826 -0.51071656
##  [7,]  9.5153954 -1.83799111  0.2380995  0.51445578
##  [8,] -1.8379911  2.87214294 -0.5554513 -0.61912632
##  [9,]  0.2380995 -0.55545131  6.9670571 -0.28461817
## [10,]  0.5144558 -0.61912632 -0.2846182  2.40040127
```

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

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

```
##   Pop envPop1 envPop2    fakeEnv1  fakeEnv2          V6         V7         V8
## 1  P1    -1.0      -1 -1.55766341 -1.218331  2.48167095 -2.3266383 -0.9232911
## 2  P2    -0.5      -1 -0.30192757 -3.453716  1.54910583 -1.3225897  0.9342882
## 3  P3     0.0      -1  0.07655442 -1.083442  0.92022721  1.2784248 -0.6983980
## 4  P4     0.5      -1  1.74133551 -1.534257 -0.12918450 -0.6684629 -1.4975938
## 5  P5     1.0      -1  1.06068083 -1.822725 -1.33633610  4.0341913  5.4891920
## 6  P6     1.0      -1 -1.99403263 -0.799573  0.06213316  3.5631245  1.8600332
##           V9        V10        V11        V12        V13         V14
## 1 -3.0389792 -0.1175726 -2.6641191  4.5741079 -1.1509603 -0.40881482
## 2 -1.3969036 -1.8124941 -5.4520813  4.4217774  0.7996067  0.05580681
## 3 -0.2334544  0.7398533 -0.6867098  2.5490811  2.1866836  0.73516433
## 4 -2.2031747  1.9988298  1.4831886  1.2658055 -0.1680325  0.10422659
## 5 -2.0126869  3.7003157 -6.9057437 -0.3406874 -3.8752309  2.04014323
## 6  1.8319891  3.2122041 -1.8588675 -1.0658706 -2.4939325  0.85259549
##           V15
## 1 -2.91087895
## 2  2.26405947
## 3 -0.09874303
## 4 -2.88658319
## 5  2.34555500
## 6  1.64360227
```

```r
tail(Popsenv)
```

```
##      Pop envPop1 envPop2    fakeEnv1   fakeEnv2         V6         V7
## 95   P95     1.0      -1  0.76484523 -4.6654590  0.7039915  2.7391255
## 96   P96     1.0      -1  0.07741494 -1.8540389  0.7253867  4.6776082
## 97   P97     0.5      -1 -0.16833838 -1.7156925  1.8891493 -1.1020824
## 98   P98     0.0      -1  1.57707697 -0.4498347  1.1302106 -0.9258929
## 99   P99    -0.5      -1 -2.26708942 -2.5682792 -0.6513680  1.2355495
## 100 P100    -1.0      -1 -2.60504819 -1.0895667  0.4221163  0.5499470
##             V8       V9        V10        V11        V12       V13        V14
## 95  -1.4752431 1.561699  1.5223218  2.5080179 -3.2920535  1.829966 -0.9090794
## 96   2.3871361 2.033116  3.7923590 -3.1580239  0.5902527  1.180809  3.9866121
## 97   1.1066143 2.975969 -1.2312342 -2.5732375  5.5336787 -3.068432  2.4378462
## 98  -1.2135786 4.189807 -0.5734596 -4.8598584  3.4958394 -1.158432 -0.3223391
## 99   4.7008175 4.283219 -1.0114508 -2.9191373  2.2528688 -2.821596  4.8388257
## 100 -0.7851984 3.484999 -3.1363250  0.5727394 -4.9115813  2.972367  1.8102608
##               V15
## 95   1.1043191816
## 96  -0.9136777296
## 97   1.9174247328
## 98   0.0008324457
## 99   3.3267511319
```

```
## 100 -0.1886744320
```

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

Standardize environments

```r
head(Popsenv)
```

```
##   Pop envPop1 envPop2    fakeEnv1  fakeEnv2          V6         V7         V8
## 1  P1    -1.0      -1 -1.55766341 -1.218331  2.48167095 -2.3266383 -0.9232911
## 2  P2    -0.5      -1 -0.30192757 -3.453716  1.54910583 -1.3225897  0.9342882
## 3  P3     0.0      -1  0.07655442 -1.083442  0.92022721  1.2784248 -0.6983980
## 4  P4     0.5      -1  1.74133551 -1.534257 -0.12918450 -0.6684629 -1.4975938
## 5  P5     1.0      -1  1.06068083 -1.822725 -1.33633610  4.0341913  5.4891920
## 6  P6     1.0      -1 -1.99403263 -0.799573  0.06213316  3.5631245  1.8600332
##           V9        V10        V11        V12        V13         V14
## 1 -3.0389792 -0.1175726 -2.6641191  4.5741079 -1.1509603 -0.40881482
## 2 -1.3969036 -1.8124941 -5.4520813  4.4217774  0.7996067  0.05580681
## 3 -0.2334544  0.7398533 -0.6867098  2.5490811  2.1866836  0.73516433
## 4 -2.2031747  1.9988298  1.4831886  1.2658055 -0.1680325  0.10422659
## 5 -2.0126869  3.7003157 -6.9057437 -0.3406874 -3.8752309  2.04014323
## 6  1.8319891  3.2122041 -1.8588675 -1.0658706 -2.4939325  0.85259549
##           V15
## 1 -2.91087895
## 2  2.26405947
## 3 -0.09874303
## 4 -2.88658319
## 5  2.34555500
## 6  1.64360227
```

```r
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)
```

```
##   Pop    envPop1   envPop2    fakeEnv1   fakeEnv2          V6         V7
## 1  P1 -1.4071247 -1.407125 -0.93917620 -0.7124808  2.09026753 -1.1305261
## 2  P2 -0.7035624 -1.407125 -0.04072126 -2.0437866  1.30380489 -0.6618167
## 3  P3  0.0000000 -1.407125  0.23007536 -0.6321463  0.77345100  0.5523877
## 4  P4  0.7035624 -1.407125  1.42119435 -0.9006337 -0.11155218 -0.3564575
## 5  P5  1.4071247 -1.407125  0.93419896 -1.0724340 -1.12958257  1.8388333
## 6  P6  1.4071247 -1.407125 -1.25139002 -0.4630854  0.04979225  1.6189301
##           V8         V9         V10         V11        V12         V13
## 1 -0.1822761 -1.4732449 -0.06485067 -0.81207017  1.4145669 -0.64753382
## 2  0.4805639 -0.7410018 -0.83552324 -1.84712847  1.3668209  0.45957281
## 3 -0.1020275 -0.2221903  0.32501676 -0.07793756  0.7798497  1.24685267
## 4 -0.3872046 -1.1005385  0.89746727  0.72765847  0.3776244 -0.08964176
## 5  2.1058905 -1.0155952  1.67112467 -2.38681489 -0.1259090 -2.19378073
## 6  0.8108976  0.6988432  1.44918266 -0.51311259 -0.3532078 -1.40978064
```

```
##           V14        V15
## 1 -0.07983095 -2.1056939
## 2  0.11683709  1.4537763
## 3  0.40439992 -0.1714267
## 4  0.13733253 -2.0889826
## 5  0.95677979  1.5098312
## 6  0.45410693  1.0270081
```

```r
# Check for mistakes
round(colMeans(PopsenvStnd[all_env_cols]))
```

```
##  envPop1  envPop2 fakeEnv1 fakeEnv2       V6       V7       V8       V9
##        0        0        0        0        0        0        0        0
##      V10      V11      V12      V13      V14      V15
##        0        0        0        0        0        0
```

```r
round(apply(PopsenvStnd[all_env_cols], 2, sd))
```

```
##  envPop1  envPop2 fakeEnv1 fakeEnv2       V6       V7       V8       V9
##        1        1        1        1        1        1        1        1
##      V10      V11      V12      V13      V14      V15
##        1        1        1        1        1        1
```

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

```
##          envPop1 envPop2 fakeEnv1 fakeEnv2    V6    V7    V8    V9   V10   V11
## envPop1     1.00    0.00     0.45    -0.03 -0.08  0.15  0.19  0.01  0.14  0.06
## envPop2     0.00    1.00    -0.14     0.61 -0.02 -0.12 -0.26  0.08 -0.19  0.17
## fakeEnv1    0.45   -0.14     1.00     0.06 -0.16 -0.07 -0.01 -0.10  0.18  0.03
## fakeEnv2   -0.03    0.61     0.06     1.00 -0.11 -0.08 -0.25  0.02 -0.04  0.12
## V6         -0.08   -0.02    -0.16    -0.11  1.00  0.01  0.16 -0.13 -0.15 -0.50
## V7          0.15   -0.12    -0.07    -0.08  0.01  1.00  0.31  0.16  0.51 -0.27
## V8          0.19   -0.26    -0.01    -0.25  0.16  0.31  1.00 -0.24  0.05 -0.17
## V9          0.01    0.08    -0.10     0.02 -0.13  0.16 -0.24  1.00 -0.05  0.10
## V10         0.14   -0.19     0.18    -0.04 -0.15  0.51  0.05 -0.05  1.00 -0.22
## V11         0.06    0.17     0.03     0.12 -0.50 -0.27 -0.17  0.10 -0.22  1.00
## V12         0.06   -0.13     0.14    -0.17  0.30 -0.27  0.20 -0.42  0.00 -0.33
## V13        -0.18    0.24    -0.12     0.22  0.10  0.05 -0.44  0.12 -0.32  0.21
## V14         0.11   -0.14    -0.12    -0.24  0.21  0.43  0.71 -0.03 -0.18 -0.18
## V15         0.10   -0.01    -0.12    -0.17  0.02  0.35  0.29  0.27  0.04 -0.25
##            V12   V13   V14   V15
## envPop1   0.06 -0.18  0.11  0.10
## envPop2  -0.13  0.24 -0.14 -0.01
## fakeEnv1  0.14 -0.12 -0.12 -0.12
## fakeEnv2 -0.17  0.22 -0.24 -0.17
## V6        0.30  0.10  0.21  0.02
## V7       -0.27  0.05  0.43  0.35
## V8        0.20 -0.44  0.71  0.29
## V9       -0.42  0.12 -0.03  0.27
## V10       0.00 -0.32 -0.18  0.04
## V11      -0.33  0.21 -0.18 -0.25
## V12       1.00 -0.41  0.19  0.06
## V13      -0.41  1.00 -0.20 -0.18
## V14       0.19 -0.20  1.00  0.20
## V15       0.06 -0.18  0.20  1.00
```

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)
```

```
##   Transplant Home  Fitness        D_CI   D_CI_sel Env_sel1 Env_sel2     dM
## 1         T1   H1 0.790825 0.000000000 0.00000000       -1       -1 3.9996
## 2         T2   H1 0.817109 0.003351372 0.01194520       -1       -1 3.9996
## 3         T3   H1 0.711378 0.005807787 0.02071331       -1       -1 3.9996
## 4         T4   H1 0.521529 0.012116530 0.04544032       -1       -1 3.9996
## 5         T5   H1 0.321762 0.016701773 0.05602929       -1       -1 3.9996
## 6         T6   H1 0.321762 0.016701773 0.05602929       -1       -1 3.9996
```

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])
```

```
##     envPop1   envPop2   fakeEnv1   fakeEnv2       V6        V7         V8
## 1 -1.407125 -1.407125 -0.9391762 -0.7124808 2.090268 -1.130526 -0.1822761
##          V9         V10        V11      V12        V13         V14       V15
## 1 -1.473245 -0.06485067 -0.8120702 1.414567 -0.6475338 -0.07983095 -2.105694
```

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

```
##      envPop1  envPop2  fakeEnv1  fakeEnv2         V6        V7         V8
## 50 -1.407125 1.407125 0.1375876 0.7334861 0.01697772 -1.794801 -0.1612388
##           V9        V10        V11       V12        V13         V14       V15
## 50 0.2092681 -0.6145491 0.08241598 0.6354016 -0.3926396 -0.01581013 0.1863589
```

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

```
##          envPop1 envPop2 fakeEnv1 fakeEnv2    V6    V7    V8    V9   V10   V11
## envPop1     1.00    0.00     0.45    -0.03 -0.08  0.15  0.19  0.01  0.14  0.06
## envPop2     0.00    1.00    -0.14     0.61 -0.02 -0.12 -0.26  0.08 -0.19  0.17
## fakeEnv1    0.45   -0.14     1.00     0.06 -0.16 -0.07 -0.01 -0.10  0.18  0.03
## fakeEnv2   -0.03    0.61     0.06     1.00 -0.11 -0.08 -0.25  0.02 -0.04  0.12
## V6         -0.08   -0.02    -0.16    -0.11  1.00  0.01  0.16 -0.13 -0.15 -0.50
## V7          0.15   -0.12    -0.07    -0.08  0.01  1.00  0.31  0.16  0.51 -0.27
## V8          0.19   -0.26    -0.01    -0.25  0.16  0.31  1.00 -0.24  0.05 -0.17
## V9          0.01    0.08    -0.10     0.02 -0.13  0.16 -0.24  1.00 -0.05  0.10
## V10         0.14   -0.19     0.18    -0.04 -0.15  0.51  0.05 -0.05  1.00 -0.22
## V11         0.06    0.17     0.03     0.12 -0.50 -0.27 -0.17  0.10 -0.22  1.00
## V12         0.06   -0.13     0.14    -0.17  0.30 -0.27  0.20 -0.42  0.00 -0.33
## V13        -0.18    0.24    -0.12     0.22  0.10  0.05 -0.44  0.12 -0.32  0.21
## V14         0.11   -0.14    -0.12    -0.24  0.21  0.43  0.71 -0.03 -0.18 -0.18
```

```
## V15         0.10   -0.01    -0.12    -0.17  0.02  0.35  0.29  0.27  0.04 -0.25
##            V12   V13   V14   V15
## envPop1   0.06 -0.18  0.11  0.10
## envPop2  -0.13  0.24 -0.14 -0.01
## fakeEnv1  0.14 -0.12 -0.12 -0.12
## fakeEnv2 -0.17  0.22 -0.24 -0.17
## V6        0.30  0.10  0.21  0.02
## V7       -0.27  0.05  0.43  0.35
## V8        0.20 -0.44  0.71  0.29
## V9       -0.42  0.12 -0.03  0.27
## V10       0.00 -0.32 -0.18  0.04
## V11      -0.33  0.21 -0.18 -0.25
## V12       1.00 -0.41  0.19  0.06
## V13      -0.41  1.00 -0.20 -0.18
## V14       0.19 -0.20  1.00  0.20
## V15       0.06 -0.18  0.20  1.00
```

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

```
## [1] 30.95207
```

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

```
## [1] 0
```

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

```
##           1
## 50 5.076738
```

Calculate environment distances

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

head(PopsenvStnd)
```

```
##   Pop    envPop1   envPop2    fakeEnv1   fakeEnv2          V6         V7
## 1  P1 -1.4071247 -1.407125 -0.93917620 -0.7124808  2.09026753 -1.1305261
## 2  P2 -0.7035624 -1.407125 -0.04072126 -2.0437866  1.30380489 -0.6618167
## 3  P3  0.0000000 -1.407125  0.23007536 -0.6321463  0.77345100  0.5523877
## 4  P4  0.7035624 -1.407125  1.42119435 -0.9006337 -0.11155218 -0.3564575
## 5  P5  1.4071247 -1.407125  0.93419896 -1.0724340 -1.12958257  1.8388333
## 6  P6  1.4071247 -1.407125 -1.25139002 -0.4630854  0.04979225  1.6189301
##           V8         V9         V10         V11        V12         V13
## 1 -0.1822761 -1.4732449 -0.06485067 -0.81207017  1.4145669 -0.64753382
## 2  0.4805639 -0.7410018 -0.83552324 -1.84712847  1.3668209  0.45957281
## 3 -0.1020275 -0.2221903  0.32501676 -0.07793756  0.7798497  1.24685267
## 4 -0.3872046 -1.1005385  0.89746727  0.72765847  0.3776244 -0.08964176
## 5  2.1058905 -1.0155952  1.67112467 -2.38681489 -0.1259090 -2.19378073
```

```
## 6  0.8108976  0.6988432  1.44918266 -0.51311259 -0.3532078 -1.40978064
##           V14        V15
## 1 -0.07983095 -2.1056939
## 2  0.11683709  1.4537763
## 3  0.40439992 -0.1714267
## 4  0.13733253 -2.0889826
## 5  0.95677979  1.5098312
## 6  0.45410693  1.0270081
```

```
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)
```

```
##   Transplant Home  Fitness        D_CI   D_CI_sel Env_sel1 Env_sel2     dM
## 1         T1   H1 0.790825 0.000000000 0.00000000       -1       -1 3.9996
## 2         T2   H1 0.817109 0.003351372 0.01194520       -1       -1 3.9996
## 3         T3   H1 0.711378 0.005807787 0.02071331       -1       -1 3.9996
## 4         T4   H1 0.521529 0.012116530 0.04544032       -1       -1 3.9996
## 5         T5   H1 0.321762 0.016701773 0.05602929       -1       -1 3.9996
## 6         T6   H1 0.321762 0.016701773 0.05602929       -1       -1 3.9996
##   EdSelEnv MdSelEnv EdAllEnv MdAllEnv
## 1       NA       NA       NA       NA
## 2       NA       NA       NA       NA
## 3       NA       NA       NA       NA
## 4       NA       NA       NA       NA
## 5       NA       NA       NA       NA
## 6       NA       NA       NA       NA
```

```
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
```

```r
  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

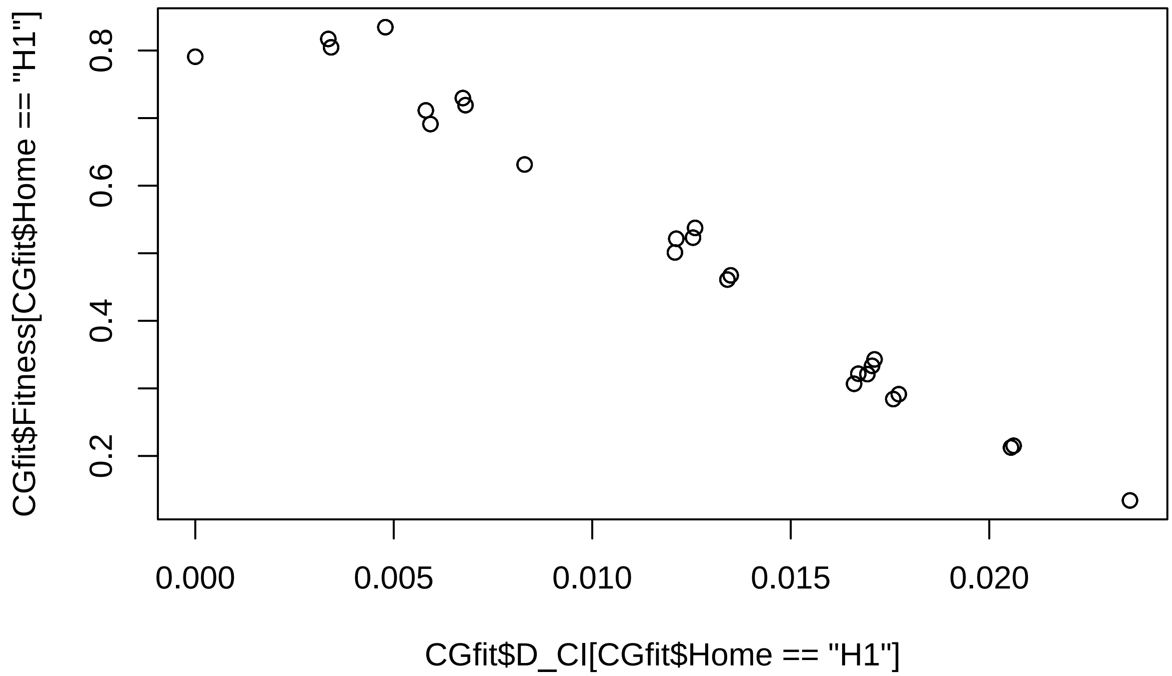
```r
head(CGfit)
```

```
##   Transplant Home  Fitness        D_CI   D_CI_sel Env_sel1 Env_sel2     dM
## 1         T1   H1 0.790825 0.000000000 0.00000000       -1       -1 3.9996
## 2         T2   H1 0.817109 0.003351372 0.01194520       -1       -1 3.9996
## 3         T3   H1 0.711378 0.005807787 0.02071331       -1       -1 3.9996
## 4         T4   H1 0.521529 0.012116530 0.04544032       -1       -1 3.9996
## 5         T5   H1 0.321762 0.016701773 0.05602929       -1       -1 3.9996
## 6         T6   H1 0.321762 0.016701773 0.05602929       -1       -1 3.9996
##    EdSelEnv MdSelEnv EdAllEnv MdAllEnv
## 1 0.0000000    0.000 0.000000  0.00000
## 2 0.7035624    0.495 4.526599 27.49723
## 3 1.4071247    1.980 4.260254 18.13383
## 4 2.1106871    4.455 4.518916 16.48463
## 5 2.8142495    7.920 7.784134 45.89380
## 6 2.8142495    7.920 6.457067 26.25406
```

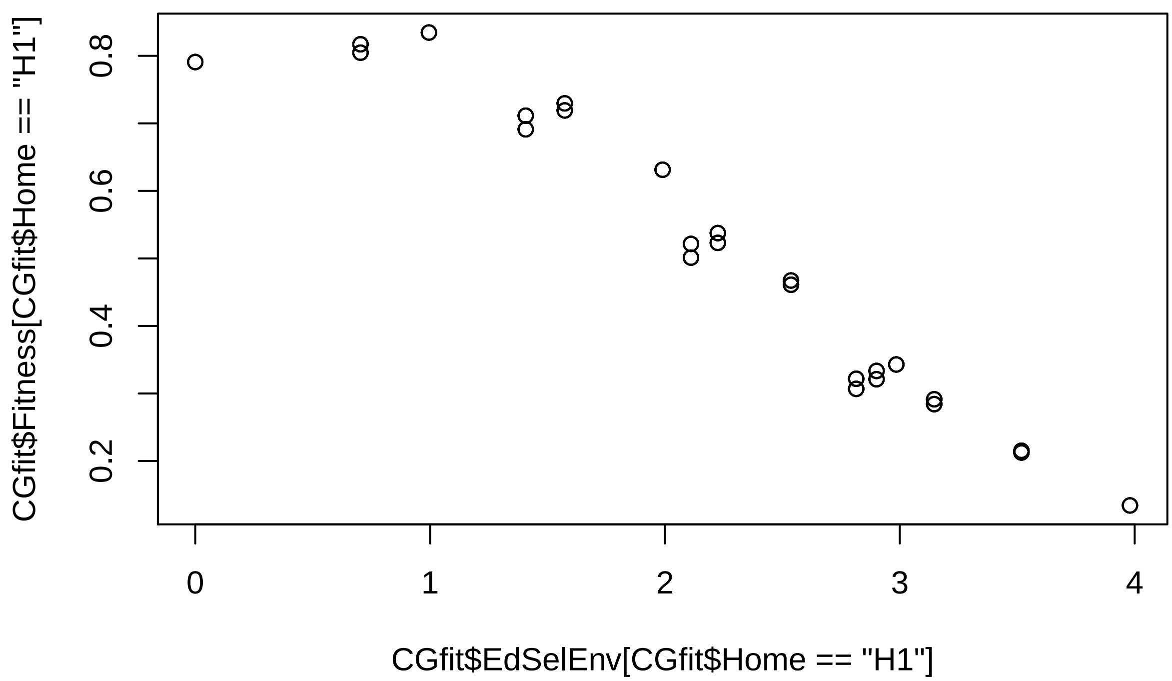
```r
#any missing data?
sum(!complete.cases(CGfit))
```

```
## [1] 0
```

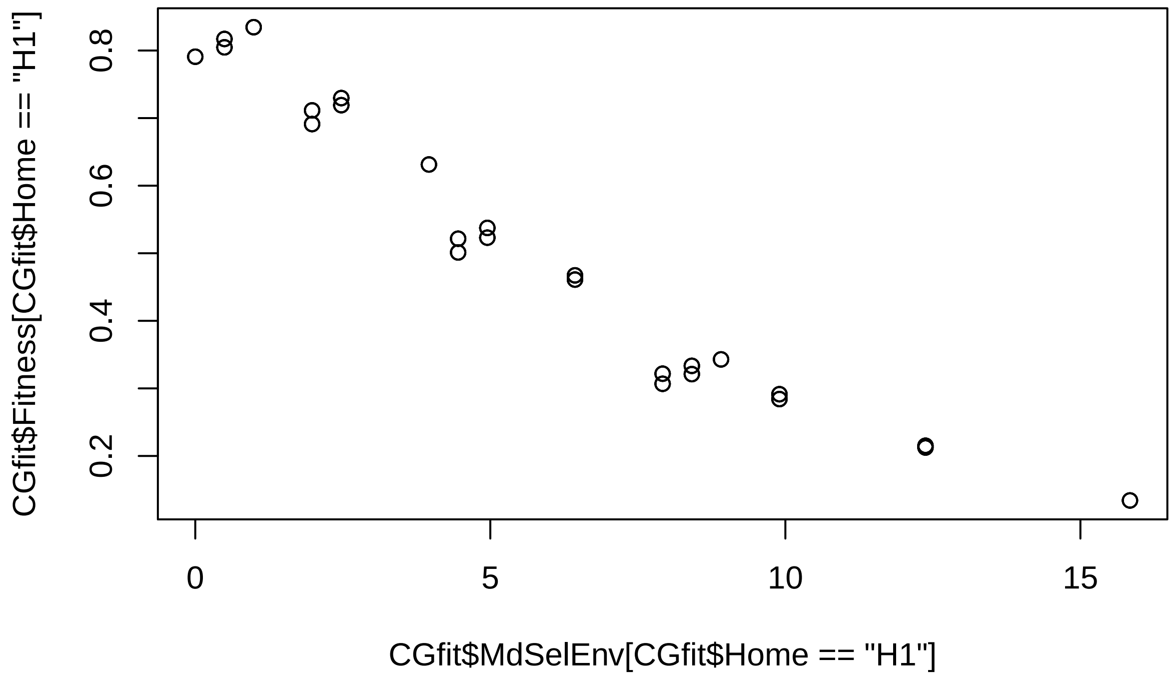
```r
# should be 0

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



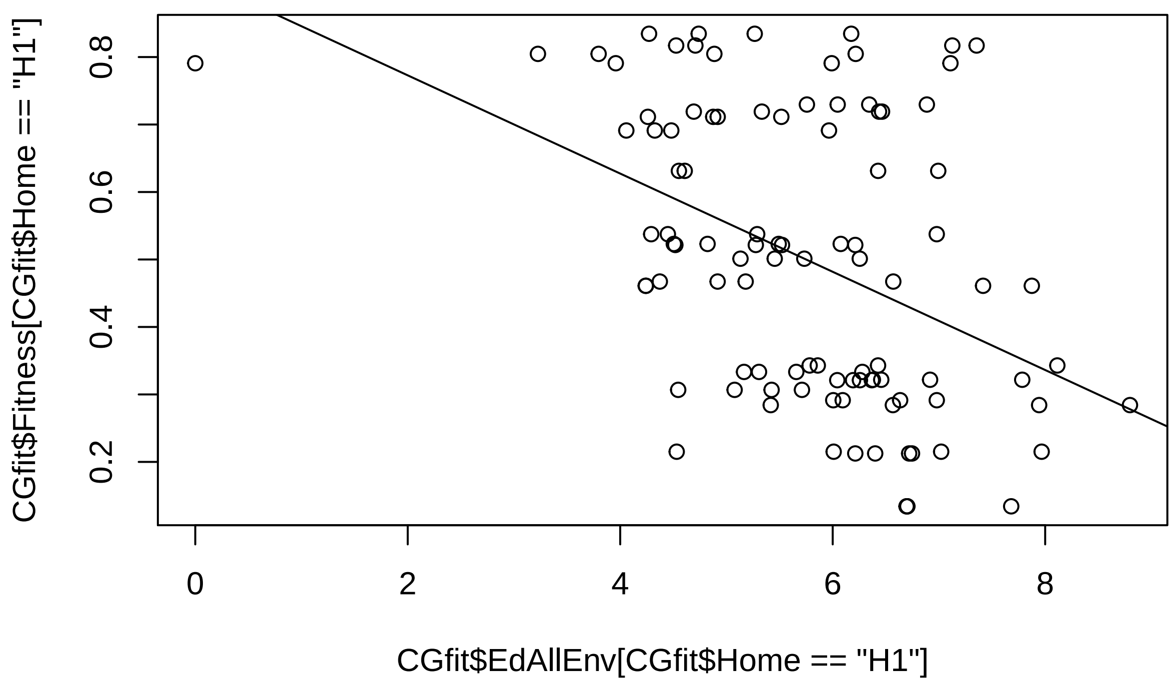

```
plot(CGfit$Fitness[CGfit$Home=="H1"]~
       CGfit$EdSelEnv[CGfit$Home=="H1"])
```




```
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"]))
```

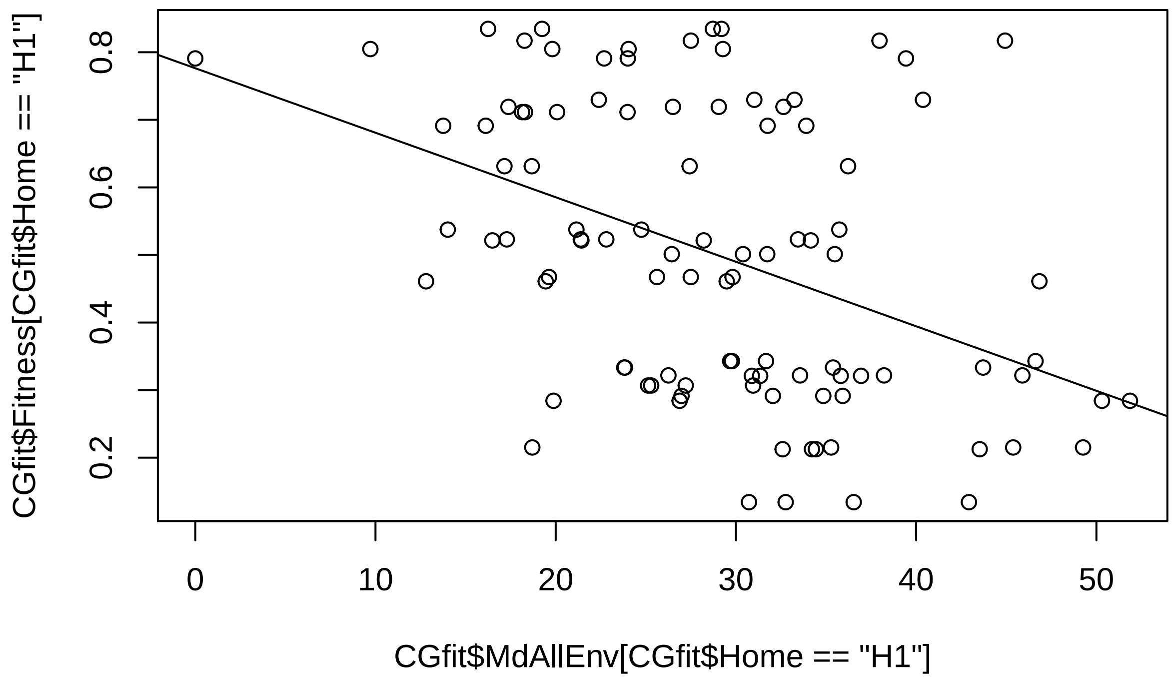



```
(cor(CGfit$Fitness[CGfit$Home=="H1"],
     CGfit$EdAllEnv[CGfit$Home=="H1"]))
```

```
## [1] -0.4265559
```

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

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




```r
cor(CGfit$Fitness[CGfit$Home=="H1"],
    CGfit$MdAllEnv[CGfit$Home=="H1"])
```

```
## [1] -0.4253841
```

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